
**Information technology — Dynamic
adaptive streaming over HTTP
(DASH) —**

**Part 1:
Media presentation description and
segment formats**

*Technologies de l'information — Diffusion en flux adaptatif
dynamique sur HTTP (DASH) —*

*Partie 1: Description de la présentation et formats de remise des
médias*



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword.....	vii
Introduction.....	viii
1 Scope.....	1
2 Normative references.....	1
3 Terms, definitions, symbols and abbreviated terms	2
3.1 Terms and definitions	2
3.2 Symbols and abbreviated terms	7
3.3 Conventions	9
4 Overview	10
4.1 System description	10
4.2 DASH Client model.....	11
4.3 DASH data model overview	12
4.4 Protocols	15
4.5 Media stream and Representation properties	16
4.5.1 Switching and Random Access Support.....	16
4.5.2 Media stream access points	16
4.5.3 Non-overlapping Segments and Subsegments	17
4.5.4 Conforming Segment track.....	18
4.6 Brands.....	18
4.7 Schemes	19
5 Media Presentation	22
5.1 General.....	22
5.2 Media Presentation Description.....	22
5.2.1 General	22
5.2.2 Schema.....	23
5.2.3 Elements and Attributes added in revisions and amendments.....	24
5.3 Hierarchical data model	26
5.3.1 General	26
5.3.2 Period.....	33
5.3.3 Adaptation Sets.....	39
5.3.4 Media content component.....	50
5.3.5 Representation	51
5.3.6 Sub-Representation	62
5.3.7 Common attributes and elements	63
5.3.8 Subsets.....	70
5.3.9 Segments and Segment information	71
5.3.10 Label and Group Label	93
5.3.11 Preselection.....	95
5.3.12 Initialization Set, Group and Presentation	99
5.3.13 Resynchronization	103
5.4 Media Presentation Description updates.....	106
5.4.1 General	106
5.4.2 MPD Reset.....	108
5.5 MPD assembly	108
5.5.1 General	108
5.5.2 Syntax and semantics	108
5.5.3 Processing	109
5.6 Base URL Processing.....	110

5.6.1	Overview	110
5.6.2	Semantics	111
5.6.3	XML syntax.....	112
5.6.4	Reference resolution.....	112
5.6.5	Alternative base URLs.....	113
5.7	Program information.....	113
5.7.1	Overview	113
5.7.2	Semantics	113
5.7.3	XML syntax.....	114
5.8	Descriptors	114
5.8.1	General.....	114
5.8.2	Semantics of generic descriptor	115
5.8.3	XML syntax of generic descriptor.....	115
5.8.4	Specific descriptors	116
5.8.5	Specific scheme definitions	122
5.9	DASH metrics descriptor	133
5.9.1	Overview	133
5.9.2	Semantics	133
5.9.3	XML syntax.....	134
5.9.4	Metric reporting	135
5.10	Events	135
5.10.1	Overview	135
5.10.2	MPD Events.....	135
5.10.3	Inband Event Signalling.....	139
5.10.4	DASH-specific events.....	142
5.11	MPD Chaining	146
5.11.1	General.....	146
5.11.2	Regular Chaining	146
5.11.3	Fallback Chaining.....	147
5.12	Producer Reference Time	148
5.12.1	General.....	148
5.12.2	Semantics	148
5.12.3	XML Syntax	150
5.13	Leap seconds.....	150
5.13.1	Overview	150
5.13.2	Semantics	151
5.13.3	XML-Syntax.....	151
5.13.4	Leap second information updates.....	152
5.14	Content Popularity Rate	152
5.14.1	General.....	152
5.14.2	Semantics	153
5.14.3	XML syntax.....	154
5.15	MPD patch framework.....	155
5.15.1	Overview	155
5.15.2	MPD Patch Location.....	155
5.15.3	MPD patch document.....	156
5.15.4	Processing Model	158
5.15.5	Recommended Client Operation.....	159
6	Segment formats.....	159
6.1	General	159
6.2	Segment types	160
6.2.1	General.....	160
6.2.2	Initialization Segment	160
6.2.3	Media Segment.....	160

6.2.4	Index Segment	162
6.2.5	Bitstream Switching Segment	162
6.2.6	Missing Content Segment.....	162
6.3	Segment formats for ISO base media file format.....	162
6.3.1	General	162
6.3.2	Preliminaries: Refinements of generic concepts	163
6.3.2.5	Resynchronization Point	163
6.3.3	Initialization Segment format	164
6.3.4	Media Segment types.....	164
6.3.5	Self-Initializing Media Segment formats	166
6.4	Segment formats for MPEG-2 transport streams	166
6.4.1	General	166
6.4.2	Preliminaries: Refinements of generic concepts	167
6.4.3	Initialization Segment types and formats.....	168
6.4.4	Media Segment types and formats	169
6.4.5	Bitstream Switching Segment	169
6.4.6	Index Segment	170
6.4.7	Boxes used with MPEG-2 TS Index Segments	172
7	Combined semantics of MPD and Segment formats	172
7.1	Overview	172
7.2	General.....	173
7.2.1	Media Presentation timeline	173
7.2.2	Segment Index	174
7.2.3	Segment alignment	174
7.2.4	Subsegment alignment	174
7.3	Media Presentation based on the ISO base media file format	174
7.3.1	General	174
7.3.2	Media presentation timeline	175
7.3.3	Authoring Rules for specific MPD attributes	175
7.3.4	Sub-Representations	176
7.3.5	Segment Timeline without Segment Index.....	176
7.4	Media Presentation based on MPEG-2 TS	176
7.4.1	General	176
7.4.2	Media presentation timeline	177
7.4.3	Authoring rules for specific MPD attributes.....	177
7.4.4	Sub-Representations	178
8	Profiles.....	178
8.1	Definition	178
8.2	Full profile	180
8.2.1	General	180
8.2.2	Media Presentation Description constraints.....	180
8.2.3	Segment format constraints	180
8.3	ISO Base media file format On Demand profile.....	180
8.3.1	General	180
8.3.2	Media Presentation Description constraints.....	181
8.3.3	Segment format constraints	182
8.4	ISO Base media file format live profile.....	182
8.4.1	General	182
8.4.2	Media Presentation Description constraints.....	182
8.4.3	Segment format constraints	183
8.5	ISO Base media file format main profile.....	183
8.5.1	General	183
8.5.2	Media Presentation Description constraints.....	184

8.5.3	Segment format constraints	184
8.6	MPEG-2 TS main profile	185
8.6.1	General	185
8.6.2	Media Presentation Description constraints	185
8.6.3	Segment format constraints	185
8.6.4	Comments and recommendations	185
8.7	MPEG-2 TS simple profile	186
8.7.1	General	186
8.7.2	Media Presentation Description constraints	186
8.7.3	Segment format constraints	186
8.7.4	Recommendations	187
8.8	ISO Base media file format extended live profile.....	187
8.8.1	General	187
8.8.2	Media Presentation Description constraints	187
8.8.3	Segment format constraints	188
8.8.4	Inband Events	188
8.9	ISO Base media file format extended On Demand profile	188
8.9.1	General	188
8.9.2	Media Presentation Description constraints	189
8.9.3	Segment format constraints	190
8.10	ISO Base media file format common profile.....	190
8.10.1	General	190
8.10.2	Media Presentation Description constraints	190
8.10.3	Segment format constraints	191
8.11	ISO Base media file format broadcast TV profile.....	191
8.11.1	General	191
8.11.2	Media Presentation Description constraints	191
8.11.3	Segment format constraints	193
8.11.4	MPD Updates and Inband Event Streams	193
8.12	DASH profile for CMAF content	194
8.12.1	General	194
8.12.2	CMAF content model for DASH profile definition.....	194
8.12.3	CMAF to DASH Mapping Principles	197
8.12.4	DASH profiles for CMAF content - Common constraints	200
8.12.5	DASH CMAF Core Profile	208
8.12.6	DASH CMAF Extended Profile	208
8.12.7	Conformance Checking Considerations	208
Annex A (informative)	Example DASH Client behaviour	210
Annex B (normative)	MPD schema	234
Annex C (normative)	MIME type registration for MPD and other resources.....	235
Annex D (normative)	DASH Metrics	242
Annex E (normative)	Byte range requests with regular HTTP GET methods	251
Annex F (informative)	Guidelines for extending DASH with other delivery formats.....	254
Annex G (informative)	MPD Examples and MPD Usage.....	256
Annex H (normative)	Spatial Relationship Description.....	284
Annex I (normative)	Flexible Insertion of URL Parameters	293
Annex J (informative)	Open GOP resolution change	307
Annex K (normative)	DASH Service Description	308
Bibliography	319

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This fifth edition cancels and replaces the fourth edition (ISO/IEC 23009-1:2020), which has been technically revised.

The main changes are as follows:

- DASH profile for using Common Media Application Format (CMAF) are added;
- The concept Resynchronization is added in order to identify stream access points in Segments;
- MPD patching is updated to support explicit MPD updates of smaller size, not only as inband messages;
- A client processing model for Event Streams and Timed Metadata tracks is introduced;
- Extensions are added to content protection for efficient signalling and to support robustness levels.
- A descriptor is added in order to describe the minimum required device output protection security;
- More flexible bandwidth signalling is provided to signal variable bitrate encoding.

A list of all parts in the ISO/IEC 23009 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and <https://www.iec.ch/national-committees>.

Introduction

Dynamic adaptive streaming over HTTP (DASH) is intended to support a media-streaming model for delivery of media content in which control lies primarily with the client. Clients may request data using the HTTP protocol from standard web servers that have no DASH-specific capabilities. Consequently, this document focuses not on client or server procedures but on the data formats used to provide a DASH Media Presentation.

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or <https://patents.iec.ch>.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Dynamic adaptive streaming over HTTP (DASH) —

Part 1: Media presentation description and segment formats

1 Scope

This document primarily specifies formats for the Media Presentation Description and Segments for dynamic adaptive streaming delivery of MPEG media over HTTP. It is applicable to streaming services over the Internet.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 13818-1, *Information technology — Generic coding of moving pictures and associated audio information — Part 1: Systems*

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 23000-19, *Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media*

ISO/IEC 23091-2, *Information technology — Coding-independent code points — Part 2: Video*

ISO/IEC 23091-3, *Information technology — Coding-independent code points — Part 3: Audio*

IETF RFC 2397, *The “data” URL scheme*

IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*

IETF RFC 4337, *MIME Type Registration for MPEG-4*

IETF RFC 4648, *The Base16, Base32, and Base64 Data Encodings*

IETF RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*

IETF RFC 5261, *An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors*

IETF RFC 5646, *Tags for Identifying Languages*

IETF RFC 6381, *The ‘Codecs’ and ‘Profiles’ Parameters for “Bucket” Media Types*

IETF RFC 6838, *Media Type Specifications and Registration Procedures*

IETF RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

IETF RFC 7233, *Hypertext Transfer Protocol (HTTP/1.1): Range Requests*

IETF RFC 8141, *URN Syntax*

IETF RFC 8673, *HTTP Random Access and Live Content*

HTML 4.01 Specification, W3C Recommendation, 24 December 1999

W3C Canonical XML Version 1.1, W3C Recommendation, 2 May 2008

W3C Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation, 26 November 2008

W3C XLINK, XML Linking Language (XLink) Version 1.1, W3C Recommendation 06, May 2010

W3C Media Fragments URI 1.0 (basic), W3C Recommendation, 25 September 2012

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23000-19 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1.1

access unit

unit of a *media stream* (3.1.29) with an assigned Media Presentation time

3.1.2

accessibility

degree to which a media content or certain *media content components* (3.1.22) are available to as many people as possible

3.1.3

Adaptation Set

set of interchangeable encoded versions of one or several *media content components* (3.1.22)

3.1.4

asset

content including media and metadata together with the rights to use the content by the content provider

3.1.5**associated Representation**

Representation (3.1.38) which provides supplemental or descriptive information for at least one other *Representation*

3.1.6**available Segment**

Segment (3.1.40) that is accessible at its assigned *HTTP-URL* (3.1.17) and a possibly assigned byte range that is the request with an HTTP GET results in a reply of the *Segment* and 2xx status code

3.1.7**Bitstream Switching Segment**

Segment (3.1.40) that if present contains essential data to switch to the *Representation* (3.1.38) it is assigned to

3.1.8

complementary *Representation*

Representation (3.1.38) which complements at least one *dependent Representation* (3.1.12)

3.1.9**continuous media**

media with an inherent notion of time

EXAMPLES Speech, audio, video, timed text or timed metadata.

3.1.10**DASH metric**

metric computed by the DASH Client and uniquely identified by a key

3.1.11**data URL**

URL with a fixed scheme “data”

3.1.12**dependent Representation**

Representation (3.1.38) for which *Segments* (3.1.40) from its *complementary Representations* (3.1.8) are necessary for presentation and/or decoding of the contained *media content components* (3.1.22)

3.1.13**earliest presentation time**

smallest *presentation time* (3.1.36) of any *access unit* (3.1.1) of a *Media Segment* (3.1.28) Or *Subsegment* (3.1.50) for a *media stream* (3.1.29)

3.1.14**event**

aperiodic sparse media-time related auxiliary information to the DASH Client or to an application

3.1.15**event stream**

sequence of related *events* (3.1.14)

3.1.16**group**

collection of *Adaptation Sets* (3.1.3) that are not expected to be presented simultaneously

3.1.17

HTTP-URL

URL with a fixed scheme of “http” or “https”

3.1.18

Index Segment

Segment (3.1.40) that primarily contains indexing information for *Media Segments* (3.1.28)

3.1.19

Initialization Segment

Segment (3.1.40) containing metadata that is necessary to present the *media streams* (3.1.29) encapsulated in *Media Segments* (3.1.28)

3.1.20

Main Adaptation Set

Adaptation Set (3.1.3) In A *Preselection* (3.1.35) that contains the *Initialization Segment* (3.1.19) for the complete experience

3.1.21

media content

single *media content period* (3.1.24) or contiguous sequence of media content periods

3.1.22

media content component

single continuous component of the *media content* (3.1.21) with an assigned *media content component type* (3.1.23)

3.1.23

media content component type

single type of *media content* (3.1.21)

EXAMPLES Audio, video, or text.

3.1.24

media content period

set of *media content components* (3.1.22) that have a common timeline as well as relationships on how they can be presented

3.1.25

Media Presentation

collection of data that establishes a bounded or unbounded presentation of *media content* (3.1.21)

3.1.26

Media Presentation Description

MPD

formalized description for a *Media Presentation* (3.1.25) for the purpose of providing a streaming service

3.1.27

Media Presentation timeline

concatenation of the timeline of all *Periods* (3.1.34) which itself is common to all *Representations* (3.1.38) in the Period

3.1.28**Media Segment**

Segment (3.1.40) that complies with media format in use and enables playback when combined with zero or more preceding Segments and an *Initialization Segment* (3.1.19) (if any)

3.1.29**media stream**

encoded version of a *media content component* (3.1.22)

3.1.30**Media Subsegment**

Subsegment (3.1.50) that only contains media data but no *Segment Index* (3.1.44)

3.1.31**message**

part of an *event* (3.1.14) containing information that is exclusively handled by the event handler

3.1.32**MPD start time**

approximate presentation start time of a *Media Segment* (3.1.28) signalled in *MPD* (3.1.26)

3.1.33**MPD duration**

approximate presentation duration of a *Media Segment* (3.1.28) signalled in *MPD* (3.1.26)

3.1.34**Period**

interval of the *Media Presentation* (3.1.24), where a contiguous sequence of all Periods constitutes the Media Presentation

3.1.35**Preselection**

set of *media content components* (3.1.22) that are intended to be consumed jointly

3.1.36**presentation time**

time associated to an *access unit* (3.1.1) that maps it to the *Media Presentation timeline* (3.1.27)

3.1.37**remote element entity**

entity that contains one or more elements and is referenced in the *MPD* (3.1.26) with an *HTTP-URL* (3.1.17) contained in an @xlink:href attribute, referred to as "remote resource" by XLink

3.1.38**Representation**

collection and encapsulation of one or more *media streams* (3.1.29) in a delivery format and associated with descriptive metadata

3.1.39**Resynchronization Point**

point within a Segment from which it is possible to start processing the Representation

3.1.40

Segment

unit of data associated with an *HTTP-URL* (3.1.17) and optionally a byte range that are specified by an *MPD* (3.1.26), or with a *data URL* (3.1.11)

3.1.41

Segment availability start time

latest time instant in *wall-clock time* (3.1.53) at which a *Segment* (3.1.40) becomes an *available Segment* (3.1.6)

3.1.42

adjusted Segment availability start time

time instant in *wall-clock time* (3.1.53) at which a *Segment* (3.1.40) becomes an *available Segment* (3.1.6)

3.1.43

Segment availability end time

time instant in *wall-clock time* (3.1.53) at which a *Segment* (3.1.40) ceases to be an *available Segment* (3.1.6)

3.1.44

Segment Index

compact index of the time range to byte range mapping within a *Media Segment* (3.1.28) separately from the *MPD* (3.1.26)

3.1.45

Segment Sequence

sequence of *Segments* (3.1.40) that are sharing a common address prefix

3.1.46

Segment Track

concatenation of *Segments* (3.1.40) forming a track with potential conformance properties

3.1.47

spatial object

media content component (3.1.22) corresponding to a region in a coordinate system associated to this media content component

3.1.48

stream access point

SAP

position in a *Representation* (3.1.38) enabling playback of a *media stream* (3.1.29) to be started using only the information contained in *Representation* data starting from that position onwards preceded by initializing data in the *Initialization Segment* (3.1.19), if any

3.1.49

sub-asset

media content component (3.1.22) (or part thereof) identified as corresponding to a part of an *asset* (3.1.4)

3.1.50**Sub-Representation**

part of a *Representation* (3.1.38) described in the *MPD* (3.1.26) that is present in the entire *Period* (3.1.34)

3.1.51**Subsegment**

unit within *Media Segments* (3.1.28) that is indexed by a *Segment Index* (3.1.44)

3.1.52**valid Segment URL**

HTTP-URL (3.1.17) that is promised to reference a *Segment* (3.1.40) during its Segment availability period

3.1.53**wall-clock time**

time as stated by UTC

3.2 Symbols and abbreviated terms

4CC	four character code
ABNF	augmented Backus-Naur form
ACK	acknowledgement message in TCP
AEMS	application event or metadata streams
API	application programming interface
AVC	advanced video coding
CAT	conditional access table
CDN	content delivery network
CMAF	common media application format
DASH	dynamic adaptive streaming over HTTP
DM	DASH metrics
DOM	document object model
DRM	digital rights management
ECM	entitlement control message
EIDR	entertainment identifier registry
EME	encrypted media extensions
EMIO	event/metadata internal object
EMM	entitlement management message
EPT	earliest presentation time
GAA	generic authentication architecture
GBA	generic bootstrapping architecture
GDR	gradual decoding refresh
GMT	Greenwich Mean Time

GOP	group of pictures
GPS	global positioning system
HDCP	high-bandwidth digital content protection
HDMI	high definition multimedia interface
HEVC	high efficiency video coding
HTML	hypertext markup language
HTTP	hypertext transfer protocol
HTTPS	secure hypertext transfer protocol
IAB	interactive advertising bureau
IDR	instantaneous decoding refresh
IP	internet protocol
ISO BMFF	ISO base media file format
LAT	Latest Arrival Time
MBT	minimum buffer time
MIME	multipurpose internet mail extensions
MPD	media presentation description
MPP	MPD patch
MVC	multi-view video coding
NGA	next generation audio
NTP	network time protocol
OATC	open authentication technical committee
OAuth	open standard for authorization
OMAP	online multimedia authorization protocol
PAT	program association table
PCR	program clock reference
PES	packetized elementary stream
PID	packet identifier
PMT	program map table
PSI	program specific information
PTS	presentation time stamp
RAP	random access point
SAML	security assertion markup language
SAP	stream access point
SEI	supplementary enhancement information
SNTP	simple NTP
SPS	sequence parameter set

SRD	spatial relationship description
SVC	scalable video coding
TCP	transmission control protocol
TLS	transport layer security
TS	transport stream
URI	uniform resource identifier
URL	uniform resource locator
URN	uniform resource name
UTC	coordinated universal time
UUID	universally unique identifier
VAST	video ad serving template
VOD	video-on-demand
XML	extensible mark-up language

3.3 Conventions

The following naming conventions apply in this document.

- Elements in an XML document are identified by an upper-case first letter and in bold face as **Element**. To express that an element **Element1** is contained in another element **Element2**, the following format is used: **Element2.Element1**. If an element's name consists of two or more combined words, camel-casing is typically used, e.g. **ImportantElement**. Elements may be present either exactly once, or the minimum and maximum occurrence is defined by <minOccurs> ... <maxOccurs>.
- Attributes in an XML document are identified by a lower-case first letter as well as they are preceded by an '@'-sign, e.g. @attribute. To point to a specific attribute @attribute contained in an element **Element**, one may write **Element@attribute**. If an attribute's name consists of two or more combined words, camel-casing is typically used after the first word, e.g. @veryImportantAttribute. Attributes may have assigned a status in the XML as mandatory (M), optional (O), optional with default value (OD) and conditionally mandatory (CM).
- Namespace qualification of elements and attributes is used as per XML standards, in the form of **namespace:Element** or @namespace:attribute. The fully qualified namespace is provided in the schema fragment associated with the declaration. External specifications extending the namespace of DASH are expected to document the element name in the semantic table with an extension namespace prefix.
- Variables defined in the context of this document are specifically highlighted with *italics*, e.g. *InternalVariable*.
- Structures that are defined as part of the hierarchical data model are identified by an upper-case first letter, e.g. Period, Adaptation Set, Representation, Segment, etc.
- The term "this clause" refers to the entire clause included within the same first heading number. The term "this subclause" refers to all text contained in the subclause with the lowest hierarchy heading.

- For improved interoperability, this document uses ABNF notation according to IETF RFC 5234 to specify certain underdefined values of XML attributes and elements.

4 Overview

4.1 System description

Dynamic adaptive streaming over HTTP (DASH) specifies XML and binary formats that enable delivery of continuous media content from standard HTTP servers to HTTP clients and enable caching of content by standard HTTP caches.

This document primarily defines two formats:

- The Media Presentation Description (MPD) describes a *Media Presentation*, i.e. a bounded or unbounded presentation of continuous media content. In particular, it defines formats to announce resource identifiers for *Segments* and to provide the context for these identified resources within a Media Presentation. These resource identifiers are HTTP-URLs possibly combined with a byte range, or with a data URL.
- The Segment formats specify the formats of the entity body of the HTTP response to an HTTP GET request or a partial HTTP GET with the indicated byte range using HTTP/1.1 as defined in IETF RFC 7233 to a resource identified in the MPD. Segments typically contain efficiently coded media data and metadata conforming to or at least closely aligned with common media formats.

The MPD provides sufficient information for a client to provide a streaming service to the user by accessing the Segments through the protocol specified in the scheme of the defined resources. In the context of this document, the assumed protocol is HTTP/1.1. Such a client is referred to as a DASH Client in the remainder of this document. However, this document does not provide a normative specification for such a client.

Figure 1 shows a possible deployment architecture in which the formats defined in this document may be used. Boxes with solid lines indicate devices that are mentioned in this document as they host or process the formats defined in this document whereas dashed boxes are conceptual or transparent. This document deals with the definition of formats that are accessible on the interface to the DASH Client, indicated by the solid lines. Any other formats or interfaces are outside the scope of this document. In the considered deployment scenario, it is assumed that the DASH Client has access to an MPD. The MPD provides sufficient information for the DASH Client to provide a streaming service to the user by requesting Segments from an HTTP server and demultiplexing, decoding and rendering the included media streams.

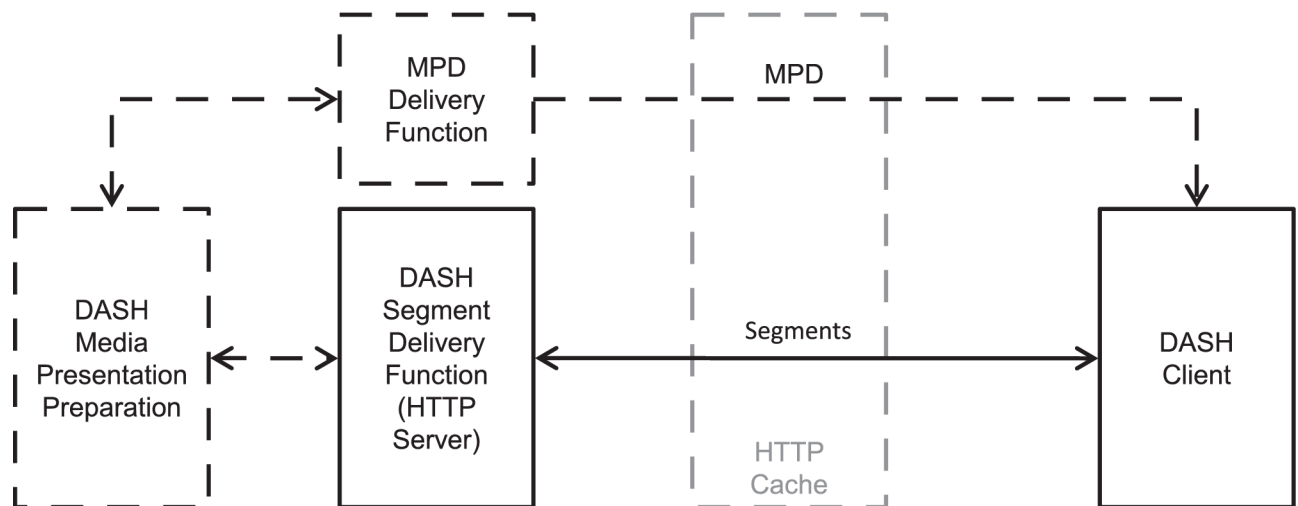


Figure 1 — Example system for DASH formats

Although the formats are initially designed to be used in the above deployment scenario, their application is obviously not restricted to this scenario. The particular aspect on "HTTP" in DASH is the usage of HTTP-URLs in the MPD for the purpose to refer to Segments. The usage of HTTP-URLs enables unique location information and it provides well-defined methods to access the resources, in particular HTTP GET and HTTP partial GET.

4.2 DASH Client model

The design of the formats defined in this document is based on the informative client model as shown in Figure 2. The figure illustrates the logical components of a conceptual DASH Client model and the relation to other components in a media streaming application. In this figure, the DASH access engine receives the Media Presentation Description (MPD), constructs and issues requests and receives Segments or parts of Segments. The DASH Client may use metadata provided in the MPD for the selection of media components by communication with the media streaming application. Such metadata may for example include codec capability information, language codes, accessibility information and other information for the selection of media components. In the context of this document, the output of the DASH access engine consists of media in MPEG container formats (ISO/IEC 14496-12 ISO base media file format or ISO/IEC 13818-1 MPEG-2 Transport Stream), or parts thereof, together with timing information that maps the internal timing of the continuous media to the timeline of the Media Presentation. In Annex F, guidance on enabling the use of this document with other container formats is provided. In addition, the DASH access engine may also receive and extract Events that are related to the media time. The events may be processed in the DASH Client or may be forwarded to an event processing application in the execution environment of the DASH Client.

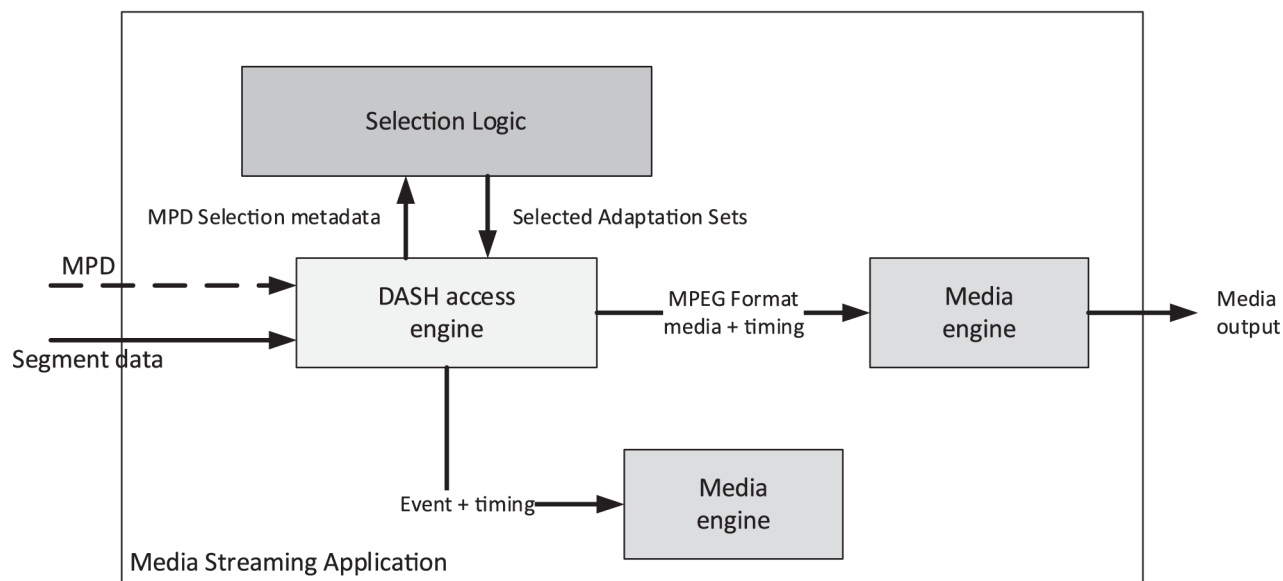


Figure 2 — DASH Client model

Furthermore, Annex K defines the Service Description to support and guide DASH Client operations.

4.3 DASH data model overview

DASH is intended to support a media-streaming model for delivery of continuous media content in which control lies primarily with the client. Clients may request data using the HTTP protocol from standard web servers that have no DASH-specific capabilities. Consequently, this document focuses not on client or server procedures but on the data formats used to provide a DASH Media Presentation.

The collection of encoded and deliverable versions of continuous media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent or certain periods of a Media Presentation may belong to the same Asset, for example a Media Presentation is a collection of main program composed of multiple periods, each assigned to the same Asset, and interleaved with inserted advertisement periods. Each media content period is composed of one or multiple **media content components**, for example audio components in various languages, different video components providing different views of the same program, subtitles in different languages, etc. Each media content component has an assigned **media content component type**, for example audio or video. The same asset over multiple periods may be identified by a DASH descriptor enabling DASH Clients to maintain the continuity across periods' boundaries. Furthermore, sub-assets composing the same asset may also be identified using a similar method. For instance, if an asset is composed of multiple video components, sub-assets enable selecting the previously selected video component after an ad insertion.

Each media content component may have several encoded versions, referred to as **media streams**. Each media stream inherits the properties of the media content, the media content period, the media content component from which it was encoded and, in addition, it gets assigned the properties of the encoding process such as sub-sampling, codec parameters, encoding bitrate, etc. This descriptive metadata is relevant for static and dynamic selection of media content components and media streams.

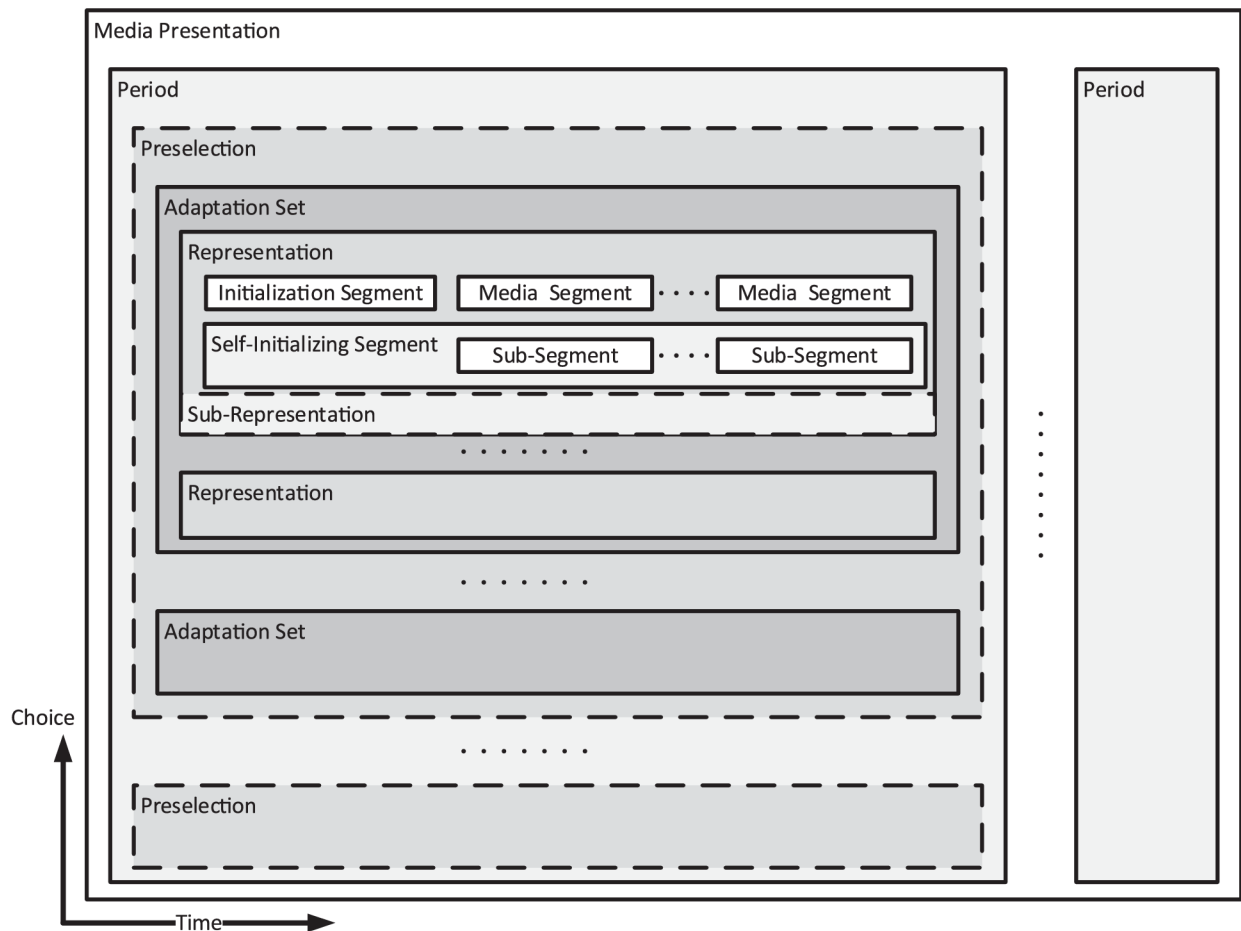


Figure 3 — DASH High-Level Data Model

DASH is based on a hierarchical data model aligned with the presentation in Figure 3. In the horizontal domain, this shows the sequence in time of the Media Presentation, and in the vertical domain it shows the choices offered in a Media Presentation, to be selected by the DASH Client in a static and dynamic manner. A DASH **Media Presentation** is described by a Media Presentation Description document. This describes the sequence of **Periods** (see subclause 5.3.2) in time that make up the Media Presentation. A Period typically represents a media content period during which a consistent set of encoded versions of the media content is available i.e. the set of available bitrates, languages, captions, subtitles etc. does not change during a Period.

Within a Period, material is arranged into **Adaptation Sets** (see subclause 5.3.3). An Adaptation Set represents a set of interchangeable encoded versions of one or several media content components (see subclause 5.3.4). For example, there may be one Adaptation Set for the main video component and a separate one for the main audio component. If there is other material available, for example captions or audio descriptions, then these may each have a separate Adaptation Set. Material may also be provided in multiplexed form, in which case interchangeable versions of the *multiplex* may be described as a single Adaptation Set, for example an Adaptation Set containing both the main audio and main video for a Period. Each of the multiplexed components may be described individually by a media content component description. In the third edition, the concept of **Preselections** (see subclause 5.3.11) was added in order to enable the combination of different Adaptation Sets into a single decoding instance and user experience.

An Adaptation Set contains a set of **Representations** (see subclause 5.3.5). A Representation describes a *deliverable encoded version* of one or several media content components. A Representation includes one or more media streams (one for each media content component in the multiplex). Any single Representation within an Adaptation Set is sufficient to render the contained media content components. By collecting different Representations in *one* Adaptation Set, the Media Presentation author expresses that the Representations represent perceptually equivalent content. Typically, this means that clients may switch dynamically from Representation to Representation within an Adaptation Set in order to adapt to network conditions or other factors. Switching refers to the presentation of decoded data up to a certain time *t*, and presentation of decoded data of another Representation from time *t* onwards. If Representations are included in one Adaptation Set, and the client switches properly, the Media Presentation is expected to be perceived seamless across the switch. DASH Clients may ignore Representations containing unsupported codecs or rendering technologies, or that are otherwise unsuitable.

Within a Representation, the content may be divided in time into **Segments** (see subclause 5.3.9 and Clause 5.15) for proper accessibility and delivery. In order to access a Segment, a URL is provided for each Segment. Consequently, a Segment is the largest unit of data that can be retrieved with a single HTTP request. For segmented Representations, two types of Segments are differentiated: **Initialization Segments** contain static metadata for the Representation, **Media Segments** contain media samples and advance the timeline. Representations may also be organized by a single self-initializing Segment which contains both initialization information as well as media data.

NOTE The MPD can also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client can in principle construct a single request for multiple Segments, but this is not the typical case.

DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components. This timeline is referred to as Media Presentation timeline. The Media Segments themselves contain accurate Media Presentation timing information enabling synchronization of components and seamless switching.

A second timeline is used to signal to clients the availability time of Segments at the specified HTTP-URLs. These times are referred to as **Segment availability times** and are provided in wall-clock time. Clients typically compare the wall-clock time to Segment availability times before accessing the Segments at the specified HTTP-URLs in order to avoid erroneous HTTP request responses. For static Media Presentations, the availability times of all Segments are identical. For dynamic Media Presentations, the availability times of segments depend on the position of the Segment in the Media Presentation timeline, i.e. the Segments get available over time. Whereas static Media Presentations are suitable to offer On-Demand content, dynamic Media Presentations are mostly suitable to offer live services.

Segments are assigned a duration, which is the duration of the media contained in the Segment when presented at normal speed. Typically, all Segments in a Representation have the same or a roughly similar duration. However, Segment duration may differ from Representation to Representation. A DASH presentation can be constructed with relative short segments (for example a few seconds), or longer Segments including a single Segment for the whole Representation.

Short Segments are usually required in the case of live content, where there are restrictions on end-to-end latency. The duration of a Segment is typically a lower bound on the end-to-end latency. DASH does not support the possibility for Segments to be extended over time: a Segment is treated as an object as a complete and discrete unit that is made available in its entirety. However, this does not prevent from applying advanced HTTP transfer modes such as chunked transfer to optimize deployments and reduce end-to-end latency.

In particular, self-initializing Segments may be further subdivided into **Subsegments** each of which contains a number of complete access units. There may also be media-format-specific restrictions on Subsegment boundaries; for example, in the ISO base media file format, a Subsegment contains a number of complete movie fragments. If a Segment is divided into Subsegments, they are described by a compact **Segment index**, which provides the presentation time range in the Representation and corresponding byte range in the Segment occupied by each Subsegment. Clients may download this index in advance and then issue requests for individual Subsegments.

Clients may switch from Representation to Representation within an Adaptation Set at any point in the media. However, switching at arbitrary positions can be complex because of coding dependencies within Representations and other factors, potentially requiring parallel download and decoding in the DASH Client. It is also desirable to avoid downloading 'overlapping' data i.e. media for the same time period from multiple Representations. Usually, switching is simplest at a stream access point (SAP) in the new stream. In order to formalize requirements related to switching, DASH defines a codec-independent concept of stream access points and identifies various types of stream access points.

Segmentation and Subsegmentation may be performed in ways that make switching simpler. For example, in the very simplest cases, each Segment or Subsegment begins with a SAP and the boundaries of Segments or Subsegments are aligned across the Representations of one Adaptation Set. In this case, switching Representation involves playing to the end of a (Sub)Segment of one Representation and then playing from the beginning of the next (Sub)Segment of the new Representation. The Media Presentation Description and Segment Index provide various indications, which describe properties of the Representations that may make switching simpler. Profiles of this document can then require these indicators to be set in certain ways, making implementation of clients for those profiles simpler at the cost of requiring the media data to obey the indicated constraints.

For On-Demand services, the Media Presentation Description is typically a static document describing the various aspects of the Media Presentation. All Segments of the Media Presentation are available on the server once any Segment is available. For live services, however, Segments become available with time as the content is produced and therefore, dynamic Media Presentations are suitable. The Media Presentation Description may be updated regularly to reflect changes in the presentation over time, for example Segment URLs for new segments may be added to the MPD and those for old, no longer available Segments may be removed. However, if Segment URLs are described using a template, this updating may not be necessary except for some redundancy/failover cases.

Events may be provided in the MPD or within a Representation in order to signal aperiodic information to the DASH Client or to an application. Events are timed, i.e. each event starts at a specific media presentation time and typically has a duration. Events include DASH-specific signalling or application-specific events. Examples for events are indication of MPD updates on the server, possibly providing the detailed update as part of the messages. The event mechanisms may also be used to deliver media time related application events, for example information about ad insertion opportunities, etc.

4.4 Protocols

This document may be deployed in a system according Figure 1 for which

- the DASH Client includes a *client* as specified in IETF RFC 7230^[22], IETF RFC 7231, IETF RFC 7232^[23], IETF RFC 7233, IETF RFC 7234^[24] and IETF RFC 7235^[25], and
- the HTTP Server hosting the DASH Segments complies with a *server* as specified in IETF RFC 7230, IETF RFC 7231, IETF RFC 7232, IETF RFC 7233, IETF RFC 7234 and IETF RFC 7235.

DASH Clients typically use the HTTP GET method or the HTTP partial GET method, as specified in IETF RFC 7231:2014, subclause 4.3, to access Segments or parts thereof.

The use of HTTP as a transport protocol inherently provides many advanced features such as caching (IETF RFC 7233), redirection (IETF RFC 7231) or authentication (IETF RFC 7235). As another example,

transport security in HTTP-based delivery may be achieved by using HTTP over TLS as specified in IETF RFC 2818. Yet another example is the use of HTTP state management mechanisms (also known as Cookies) as defined in IETF RFC 6265^[21].

However, the formats defined in this document may also be used with other protocols. In particular, the objects may be delivered with any object delivery protocol that provides a binding between an HTTP-URL and the delivered object.

4.5 Media stream and Representation properties

4.5.1 Switching and Random Access Support

The formats defined in this document are designed to provide a good user experience even in cases where the access bandwidth between the DASH segment delivery function or the cache varies. A key functionality is the ability of the DASH Client to seamlessly switch across different Representations of the same media component without severely impacting the user experience.

Assume two Representations A and B. A switch from Representation A to Representation B at media time t is considered seamless, if the result of the presentation after this switch is applied is the same as if Representation A was decoded from the beginning and presented up to time t and Representation B is decoded from the beginning and presented from time t onwards.

Media Presentations may provide different Representations in one Adaptation Set representing the same media component. If such Representations are properly time-aligned (as expected by the Media Presentation), then DASH Clients may apply seamless switching across different Representations provided in one Adaptation Set at any time t to obtain a perceptually continuous experience.

However, in practical implementations, the operation of seamless switching can be complex, as switching at time t can require parallel download and decoding of two Representations. Therefore, providing suitable switching opportunities in regular time intervals simplifies client implementations. This document provides means for providing suitable switching opportunities and in addition provides abilities to signal the position and media time of the switching opportunities.

For this purpose, this subclause defines three relevant concepts to support seamless switching:

- Media stream access points in subclause 4.5.2 to signal positions where to easily switch to a Representation, and in addition where to suitable access a Representation at start-up or seek.
- Non-overlapping Segments and Subsegments in subclause 4.5.3 to signal that, at the signalled stream access points, no overlap decoding of Representations is necessary in order to provide a continuous switch.
- Segment concatenation in subclause 4.5.4 to signal that the concatenation of two Representations at a switch point results in a conforming bitstream.

These three properties are neither sufficient nor necessary for seamless switching, but certain implementation or profiles may use these properties in order to simplify practical implementations.

4.5.2 Media stream access points

To be able to access a Representation, each of the media streams that are contained in the Representation requires media stream access points (SAPs). SAPs in the context of this document refer to the SAP definition in ISO/IEC 14496-12. ISO/IEC 14496-12 SAP types defines different types of SAPs that provide a relationship between the position where a stream can be accessed, relative to the start of a Segment or Subsegment, its presentation time and the presentation times and position of other access unit in the stream. The same SAP type definitions shall apply for this document.

A SAP is a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initializing data in the Initialization Segment, if any).

For each SAP, the properties, I_{SAP} , T_{SAP} , I_{SAU} , T_{DEC} , T_{EPT} , and T_{PTF} , are identified and defined in ISO/IEC 14496-12 SAP properties.

In particular, T_{SAP} is defined to be the earliest presentation time of any access unit of the media stream such that all access units of the media stream with presentation time greater than or equal to T_{SAP} can be correctly decoded using data in the Representation starting at byte position I_{SAP} and no data before I_{SAP} .

NOTE The type of SAP is dependent only on which access units are correctly decodable and their arrangement in presentation order. The types informally correspond with some common terms:

- Type 1 corresponds to what is known in some coding schemes as a “Closed GoP random access point” (in which all access units, in decoding order, starting from I_{SAP} can be correctly decoded, resulting in a continuous time sequence of correctly decoded access units with no gaps) and in addition the access unit in decoding order is also the first access unit in presentation order.
- Type 2 corresponds to what is known in some coding schemes as a “Closed GoP random access point”, for which the first access unit in decoding order in the media stream starting from I_{SAU} is not the first access unit in presentation order.
- Type 3 corresponds to what is known in some coding schemes as an “Open GoP random access point”, in which there are some access units in decoding order following I_{SAU} that cannot be correctly decoded and have presentation times less than T_{SAP} .
- Type 4 corresponds to what is known in some coding schemes as a “gradual decoder refresh (GDR) random access point”, in which there are some access units in decoding order starting from and following I_{SAU} that cannot be correctly decoded and have presentation times less than T_{SAP} .
- Type 5 corresponds to the case for which there is at least one access unit in decoding order starting from I_{SAP} that cannot be correctly decoded and has presentation time greater than T_{DEC} and where T_{DEC} is the earliest presentation time of any access unit starting from I_{SAU} .
- Type 6 corresponds to the case for which there is at least one access unit in decoding order starting from I_{SAP} that cannot be correctly decoded and has presentation time greater than T_{DEC} and where T_{DEC} is not the earliest presentation time of any access unit starting from I_{SAU} .

SAPs are mostly relevant for two purposes in this document:

- 1) For randomly accessing a Media Presentation, for example at the startup of the Media Presentation, after a seeking operation or after an error event especially in live cases.
- 2) To permit switching between two Representations whereby for seamless switching each media stream i in the switch-from Representation is presented up to $T_{SAP}(i)$ and each media stream i in the switch-to Representation is presented from the media stream access point starting from $T_{SAP}(i)$.

There are obvious benefits for the client to be able to identify SAPs and one or several of their properties, in particular I_{SAP} and T_{SAP} for each media stream without requiring to access data at positions following I_{SAP} . DASH provides functionalities to explicitly signal such information by using signals in the MPD or the Segment Index or combinations of the two.

4.5.3 Non-overlapping Segments and Subsegments

Segments and Subsegments represent units for which the client has an exact map on how to access and download the unit using HTTP GET or HTTP partial GET methods.

Segments (respectively Subsegments) are typically generated by segmenting encoded media streams into appropriate units. If the generation of Segments (respectively Subsegments) adheres to certain rules,

then the sequential decoding and presentation of Media Segments (respectively Subsegments) results in a correct presentation of all contained media streams. To define such rules the notion of “non-overlapping” segments (respectively Subsegments) is defined as follows.

Let

- $T_E(S,i)$ be the earliest presentation time of any access unit in stream i of a Segment or Subsegment S , and
- $T_L(S,i)$ be the latest presentation time of any access unit in stream i of a Segment or Subsegment S .

Then two segments (respectively Subsegments), A and B , which may or may not be of different Representations, are *non-overlapping* if $T_L(A,i) < T_E(B,i)$ for all media streams i in A and B or if $T_L(B,i) < T_E(A,i)$ for all streams i in A and B where i refers to the same media component.

The property of “non-overlapping” segments (respectively Subsegments) is used to define the terms Segment alignment and Subsegment alignment.

4.5.4 Conforming Segment track

A sequence of Segments (respectively Subsegments) is a “conforming Segment (respectively Subsegment) track” if the concatenation of all Segments (respectively Subsegments) in the sequence of Segments (respectively Subsegments) results in a bitstream that conforms to the media formats in use (including container and codecs).

NOTE This implies that a player conforming to the media format can play the resulting bitstream.

4.6 Brands

The ISO base media file format, ISO/IEC 14496-12, defines the concept of brands; brand values identify specifications or conformance points. This document specifies several brands, as listed in Table 1.

Table 1 — Brands defined in this document

Brand identifier	Clause in this document	Informative description
emsg	5.10.3.3	Event message box.
miss	6.2.6	Missing content segment
msdh	6.3.4.2	Media Segment conforming to the general format type for ISO base media file format.
msix	6.3.4.3	Media Segment conforming to the Indexed Media Segment format type for ISO base media file format.
dsms	6.3.5.1	Media Segment conforming to the DASH Self-Initializing Media Segment format type for ISO base media file format.
dash	6.3.5.2	ISO base media file format file specifically designed for DASH including movie fragments and Segment Index.
sims	6.4.4.4	Media Segment conforming to the Sub-Indexed Media Segment format type for ISO base media file format.
sisx	6.4.6.2	Single Index Segment used to index MPEG-2 TS based Media Segments.
risx	6.4.6.3	Representation Index Segment used to index MPEG-2 TS based Media Segments.

ssss	6.4.6.4	Subsegment Index Segment used to index MPEG-2 TS based Media Segments.
lmsg	7.3.1	Last Media Segment indicator for ISO base media file format.
dums	8.11.3	Segment sequence indicator in ISO base media file format broadcast TV profile.

4.7 Schemes

This document specifies several schemes as listed in Table 2.

Table 2 — Schemes defined in this document

Scheme identifier	Clause in this document	Informative description
urn:mpeg:dash:schema:mpd:2011	Annex B	The namespace of the XML schema for the MPD.
urn:mpeg:dash:period-continuity:2015	5.3.2.4	Period continuity signalling.
urn:mpeg:dash:period-connectivity:2015	5.3.2.4	Period connectivity signalling.
urn:mpeg:dash:adaptation-set-switching:2016	5.3.3.5	Scheme Identifier for switching across Adaptation Sets.
urn:mpeg:dash:preselection:2016	5.3.11.2	Preselection descriptor.
urn:mpeg:dash:reset:2016	5.4.2	MPD reset indicator.
urn:mpeg:dash:resolve-to-zero:2013	5.5.3	xlink resolution to zero element.
urn:mpeg:dash:mp4protection:2011	5.8.5.2	Protection schemes identified by the Scheme Type within the Scheme Type Box of the Protection Scheme Information Box of ISO/IEC 14496-12.
urn:mpeg:dash:13818:1:CA_descriptor:2011	5.8.5.2	Conditional Access System used for ISO/IEC 13818-1 (MPEG-2 Transport Stream).
urn:mpeg:dash:14496:10:frame_packing_arrangement_type:2011	5.8.5.3	Frame-packing arrangement.
urn:mpeg:dash:13818:1:stereo_video_format_type:2011	5.8.5.3	Frame-packing arrangement.
urn:mpeg:dash:23003:3:audio_channel_configuration:2011	5.8.5.4	Channel configuration. Legacy format for backward-compatibility, it is recommended to use the signalling as defined in ISO/IEC 23091-3 instead.
urn:mpeg:dash:outputChannelPositionList:2012	5.8.5.4	A list of output channel position to signal individual speaker positions as defined in ISO/IEC 23001-8. Legacy format for backward-compatibility, it is recommended to use

Scheme identifier	Clause in this document	Informative description
		the signalling as defined in ISO/IEC 23091-3 instead.
urn:mpeg:dash:role:2011	5.8.5.5	DASH role scheme.
urn:mpeg:dash:stereoid:2011	5.8.5.6	Scheme for multiple views media content description.
urn:mpeg:dash:utc:ntp:2014	5.8.5.7	UTC Timing scheme for NTP servers.
urn:mpeg:dash:utc:sntp:2014	5.8.5.7	UTC Timing scheme for SNTP servers.
urn:mpeg:dash:utc:http-head:2014	5.8.5.7	UTC Timing scheme for HTTP date headers.
urn:mpeg:dash:utc:http-xsdate:2014	5.8.5.7	UTC Timing scheme for HTTP server with xsdate format.
urn:mpeg:dash:utc:http-iso:2014	5.8.5.7	UTC Timing scheme for HTTP server with ISO timing format.
urn:mpeg:dash:utc:http-ntp:2014	5.8.5.7	UTC Timing scheme for HTTP server with NTP format.
urn:mpeg:dash:utc:direct:2014	5.8.5.7	UTC Timing scheme for direct inclusion of time.
urn:mpeg:dash:audio-receiver-mix:2014	5.8.5.8	Scheme identifier for receiver mix.
urn:mpeg:dash:mpd-as-linking:2015	5.8.5.9	MPD Adaptation Set Linking scheme.
urn:mpeg:dash:sai:2015	5.8.5.10	Sub-Asset Scheme Identifier.
urn:mpeg:dash:client-authentication:2015	5.8.5.11	Client Authentication scheme.
urn:mpeg:dash:content-authorization:2015	5.8.5.11	Content Access Authorization scheme.
urn:mpeg:dash:audio-interactivity:2016	5.8.5.12	Scheme to indicate content interactivity.
urn:mpeg:dash:qr-equivalence:2019	5.8.5.13	Quality Equivalence Descriptor
urn:mpeg:dash:output-protection:hdc:2020	5.8.5.14.2	Scheme to indicate the HDCP output protection
urn:mpeg:dash:event:2012	5.10.4	DASH event signalling scheme.
urn:mpeg:dash:event:callback:2015	5.10.4.5	DASH call back event.
urn:mpeg:dash:event:ttfn:2016	5.10.4.6	Presentation Termination Event.
urn:mpeg:dash:mpd-chaining:2016	5.11.1	MPD chaining descriptor.
urn:mpeg:dash:fallback:2016	5.11.2	MPD fallback descriptor.
urn:mpeg:dash:profile:full:2011	8.2	Identifier for Full profile.
urn:mpeg:dash:profile:isoff-on-demand:2011	8.3	Identifier for ISO Base media file format On Demand profile.

Scheme identifier	Clause in this document	Informative description
urn:mpeg:dash:profile:isoff-live:2011	8.4	Identifier for ISO Base media file format live profile.
urn:mpeg:dash:profile:isoff-main:2011	8.5	Identifier for ISO Base media file format main profile.
urn:mpeg:dash:profile:mp2t-main:2011	8.6	Identifier for MPEG-2 TS main profile.
urn:mpeg:dash:profile:mp2t-simple:2011	8.7	Identifier for MPEG-2 TS simple profile.
urn:mpeg:dash:profile:isoff-ext-live:2014	8.8	Identifier for ISO Base media file format extended live profile.
urn:mpeg:dash:profile:isoff-ext-on-demand:2014	8.9	Identifier for ISO Base media file format extended On Demand profile.
urn:mpeg:dash:profile:isoff-common:2014	8.10	Identifier for ISO Base media file format common profile.
urn:mpeg:dash:profile:isoff-broadcast:2015	8.11	Identifier for ISO Base media file format broadcast TV profile
urn:mpeg:dash:profile:cmaf:2019	8.12	Identifier for DASH core profile for CMAF content
urn:mpeg:dash:profile:cmaf-extended:2019	8.12	Identifier for DASH extended profile for CMAF content
urn:mpeg:dash:event:catchall:2020	A.13	URI for subscribing to all application event and timed metadata received by DASH Client
urn:mpeg:dash:srd:2014	H.1	Scheme identifier for Spatial Relationship Description. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in Annex H.
urn:mpeg:dash:srd:dynamic:2016	H.1	Scheme to signal dynamic SRD. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in Annex H.
urn:mpeg:dash:urlparam:2014	I.2	Scheme identifier for indicating usage of the flexible insertion of URL query parameters. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in Annex I.
urn:mpeg:dash:urlparam:2016	I.3	Scheme identifier for indicating usage of the extended parameterization scheme. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in Annex I.
urn:mpeg:dash:resolutionSwitching:2016	Annex J	Descriptor for Open GOP resolution change

5 Media Presentation

5.1 General

A Media Presentation is a collection of data that is accessible to a DASH Client to provide a streaming service to the user.

A Media Presentation is described by an MPD including possible updates of the MPD. The MPD is defined in subclause 5.2 and the MPD update mechanism is defined in subclause 5.4. Assembly of a fragmented MPD is defined in subclause 5.5. The data model that constitutes a Media Presentation is defined in subclause 5.2.3.5. In subclause 5.6, the formats and processing of URLs in the MPD is introduced. Program information is defined in subclause 5.7. Descriptors associated to Representations or collections thereof are provided in subclause 5.8. DASH metric collection description is specified in subclause 5.9. Subclause 5.10 defines Events and Event Streams. Subclause 5.11 defines MPD chaining. Subclause 5.12 defines the Producer Reference Time. Subclause 5.13 defines how to deal with leap seconds. Subclause 5.14 defines content popularity rates.

5.2 Media Presentation Description

5.2.1 General

The Media Presentation Description (MPD) is a document that contains metadata required by a DASH Client to construct appropriate HTTP-URLs to access Segments and to provide the streaming service to the user.

NOTE 1 Actual playback of the media streams included in the Representations is not controlled by the MPD information. Playback is controlled by the media engine operating on the media streams contained in the Representations in the usual way.

The format of URLs in the MPD and the process to generate HTTP GET and partial GET requests from URLs provided in the MPD is defined in subclause 5.6.

The MPD is an XML document that shall be formatted according to the XML schema provided in Annex B. Some context on the schema is provided in subclause 5.2.2.

The extension of the DASH XML schema (as provided in Annex B), in particular the addition of XML attributes or elements in the DASH namespace, is reserved to ISO/IEC. Elements and attributes that have been added to the namespace compared to earlier revisions of this document are documented in subclause 5.2.3.

The MPD shall be authored such that, after XML attributes or elements in the DASH namespace but not in the XML schema documented in Annex B are removed, the result is a valid XML document formatted according to that schema and that conforms to this document.

In addition, the MPD shall be authored such that, after XML attributes or elements in the other namespaces than the DASH namespace are removed, the result is a valid XML document formatted according to that schema and that conforms to this document.

NOTE 2 Based on the last two paragraphs, if DASH Clients remove all XML attributes and elements from the MPD in the DASH namespace and in other namespaces that are not in the XML schema documented in Annex B, the MPD results in a valid XML document which complies with this document. The DASH Client can use such a resulting MPD for presentation of a conforming Media Presentation.

In addition, rules for authoring of MPDs conforming to a specific profile are provided in 8. Certain profiles as defined in 8 may permit ignoring certain elements and attributes. However, this has no effect on the general MPD conformance rules defined in this subclause.

Following XML rules, the MPD document shall contain exactly one **MPD** element as specified in subclause 5.2.3.5.

The MIME type of the MPD document is defined in Annex C.

The encoding of the MPD shall be UTF-8 as defined in IETF RFC 3629. All data provided in extension namespaces shall be UTF-8 as defined in IETF RFC 3629. If binary data needs to be added, it shall be included in Base64 as described in IETF RFC 4648 within a UTF-8 encoded element with a proper name space or identifier, such that an XML parser knows how to process or ignore it.

The delivery of the MPD is outside the scope of this document. However, if the MPD is delivered over HTTP, then the MPD document may be transfer encoded for transport, as described in IETF RFC 7230.

NOTE 3 As an example, the GZip algorithm as defined in IETF RFC 1952^[3] can be used for Transfer-encoding.

NOTE 4 MPD encryption is not a normative part of this document. However, if operating in an insecure environment and required by the content/service provider, elements and attributes of MPD can be encrypted to protect their confidentiality by using the syntax and processing rules specified in the “XML Encryption Syntax and Processing” by W3C^[8].

NOTE 5 MPD integrity protection is not a normative part of this document. However, if operating in an insecure environment and required by the content/service provider, the digital signing and verification procedures specified in the “XML Signature Syntax and Processing” by W3C^[9] can be used to protect data origin authenticity and integrity of the MPD.

Selected MPD examples are provided in Annex G.

5.2.2 Schema

The initial part of the XML schema of the MPD is provided below, including namespace and other definitions. Specific types, elements and attributes are introduced in the remainder of this subclause. The complete normative MPD schema is provided in Annex B. In case of inconsistencies, the schema in Annex B takes precedence both over the XML syntax snippets provided in this clause and all prose text in this document.

Implementors are supported by files available at: <https://standards.iso.org/iso-iec/23009/-1/ed-5/en>. These files include the schema as well as all examples provided in Annex G.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="urn:mpeg:dash:schema:mpd:2011" targetNamespace="urn:mpeg:dash:schema:mpd:2011"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import
    schemaLocation="http://www.w3.org/XML/2008/06/xlink.xsd" namespace="http://www.w3.org/1999/xlink"
    >
    <xs:annotation>
      <xs:documentation xml:lang="en">
        xlink import. URL is per https://www.w3.org/TR/xlink/ Users are encouraged to store
        a local copy
      </xs:documentation>
    </xs:annotation>
  </xs:import>
  <xs:annotation>
    <xs:appinfo>Media Presentation Description</xs:appinfo>
    <xs:documentation xml:lang="en">
      This Schema defines the Media Presentation Description for MPEG-DASH.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="MPD" type="MPDtype">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        MPD: main element
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="MPDtype">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        MPD Type
      </xs:documentation>
    </xs:annotation>
    ...
  </xs:complexType>
  </xs:schema>
```

```
</xs:complexType>
</xs:schema>
```

5.2.3 Elements and Attributes added in revisions and amendments

5.2.3.1 Overview

In amendments and revisions of this document, the schema defined in Annex B may have been extended. In order to track this, subclause 5.2.3 tracks the addition of elements and attributes.

By following the rules in subclause 5.2.1, a single MPD can be authored for clients that implement different versions of this document.

5.2.3.2 Elements and Attributes added in the second edition of this document (ISO/IEC 23009-1:2014)

ISO/IEC 23009-1:2014 added the following elements and attributes to the schema defined in Annex B compared to the 2012 edition (ISO/IEC 23009-1:2012/Cor 1:2013):

- **MPD@publishTime**
- **MPD.EssentialProperty**
- **MPD.SupplementalProperty**
- **Period.AssetIdentifier**
- **Period.EventStream**
- **Period.SupplementalProperty**
- **RepresentationBase.InbandEventStream**
- **SegmentBase@availabilityTimeOffset**
- **SegmentBase@availabilityTimeComplete**
- **BaseURL@availabilityTimeOffset**
- **BaseURL@availabilityTimeComplete**
- **Subset@id**
- **SegmentTimeline.S@n**

5.2.3.3 Elements and Attributes added in the third edition of this document (ISO/IEC 23009-1:2019)

ISO/IEC 23009-1:2019 added the following elements and attributes to the schema defined in Annex B compared to the 2014 revision (ISO/IEC 23009-1:2014) of this document:

- **MPD.UTCTiming**
- **Period.GroupLabel**
- **Period.Preselection**
- **Period.EmptyAdaptationSet**

- **RepresentationBase.Switching**
- **RepresentationBase.RandomAccess**
- **RepresentationBase.GroupLabel**
- **RepresentationBase.Label**
- **RepresentationBase@selectionPriority**
- **Representation@tag**
- **Representation@associationId**
- **Representation@associationType**
- **SegmentBase@presentationDuration**
- **SegmentBase@timeShiftBufferDepth**
- **SegmentTimeline.S@n**
- **SegmentTimeline.S@k**

5.2.3.4 Elements and Attributes added in the fourth edition (ISO/IEC 23009-1)

This revision adds the following elements and attributes to the schema defined in Annex B compared to the 2019 revision (ISO/IEC 23009-1:2019) of this document:

- **MPD.ServiceDescription**
- **MPD.InitializationSet**
- **MPD.InitializationGroup**
- **MPD.InitializationPresentation**
- **MPD.LeapSecondInformation**
- **Period.ServiceDescription**
- **EventStream@presentationTimeOffset**
- **Event@contentEncoding**
- **AdaptationSet@initializationSetRef**
- **RepresentationBase.ProducerReferenceTime**
- **RepresentationBase.ContentPopularityRate**
- **Preselection@order**

- **SegmentBase.FailoverContent**
- **SegmentBase@eptDelta**
- **MultiSegmentBase@endNumber**
- **BaseURL@timeShiftBufferDepth**

5.2.3.5 Elements and Attributes added in the fifth edition (ISO/IEC 23009-1:2021)

This revision adds the following elements and attributes to the schema defined in Annex B compared to the 2020 revision (ISO/IEC 23009-1:2020) of this document:

- **MPD.ContentProtection**
- **MPD.PatchLocation**
- **Period.ContentProtection**
- **AdaptationSet@initializationPrincipal**
- **RepresentationBase.Resync**
- **RepresentationBase@containerProfiles**
- **Representation.ExtendedBandwidth**
- **ContentProtection@ref**
- **ContentProtection@refId**
- **ContentProtection@robustness**
- **RepresentationBase.OutputProtection**

5.3 Hierarchical data model

5.3.1 General

5.3.1.1 Overview

A Media Presentation as described in the **MPD** consists of a sequence of one or more Periods as described in subclause 5.3.2.

- Each Period contains one or more Adaptation Sets as described in subclause 5.3.3. In case an Adaptation Set contains multiple media content components, then each media content component is described individually as defined in subclause 5.3.4.
- Each Adaptation Set contains one or more Representations as described in subclause 5.3.5.
- Adaptation Sets, Representations and Sub-Representations share common attributes and elements that are described in subclause 5.3.7.
- Preselections may be used to select experiences as defined in subclause 5.3.11. Each Preselection references to one or more media content components within one or multiple Adaptation Sets.

- Each Period may contain one or more Subsets that restrict combination of Adaptation Sets for presentation. Subsets are described in subclause 5.3.8.
- Each Representation consists of one or more Segments described in subclause 5.15. Segment Information is introduced in subclause 5.3.9. Segments contain media data and/or metadata to access, decode and present the included media content. Representations may also include Sub-Representations as defined in subclause 5.3.5.6 to describe and extract partial information from a Representation.
- Each Segment consists of one or more Subsegments. Subsegments are described in subclause 6.2.3.2.
- Labels for annotation of different hierarchies are defined in subclause 5.3.10.

The summary of the semantics of the attributes and elements within an **MPD** element are provided in Table 3, subclause 5.3.1.2. The XML syntax of the **MPD** element is provided in subclause 5.3.1.3.

5.3.1.2 Semantics

Table 3 — Semantics of MPD element

Element or Attribute Name	Use	Description
MPD		The root element that carries the Media Presentation Description for a Media Presentation.
@id	0	specifies an identifier for the Media Presentation. It is recommended to use an identifier that is unique within the scope in which the Media Presentation is published. If not specified, no MPD-internal identifier is provided. However, for example the URL to the MPD may be used as an identifier for the Media Presentation.
@profiles	M	specifies a list of Media Presentation profiles as described in 8. The contents of this attribute shall conform to either the <code>pro-simple</code> or <code>pro-fancy</code> productions of IETF RFC 6381:2011, Section 4.5, without the enclosing <code>DQUOTE</code> characters, i.e. including only the <code>unencodedv</code> or <code>encodedv</code> elements respectively. As profile identifier a restricted URI format as defined in 8.1 shall be used.

Element or Attribute Name	Use	Description
@type	OD default: static	<p>specifies the type of the Media Presentation. For static Media Presentations (@type="static"), all Segments are available between the @availabilityStartTime and the @availabilityEndTime. For dynamic Media Presentations (@type="dynamic"), Segments typically have different availability times. For details, refer to subclause 5.3.9.5.3.</p> <p>In addition, the Media Presentation Description may be updated in dynamic Media Presentations, i.e. the @minimumUpdatePeriod may be present.</p> <p>NOTE Static Media Presentations are typically used for On-Demand services, whereas dynamic Media Presentations are used for live services.</p>
@availabilityStartTime	CM shall be present for @type='dynamic'	<p>For @type='dynamic', this attribute shall be present. In this case, it specifies the anchor for the computation of the earliest availability time (in UTC) for any Segment in the Media Presentation.</p> <p>For @type="static" if present, it specifies the Segment availability start time for all Segments referred to in this MPD. If not present, all Segments described in the MPD shall become available at the time the MPD becomes available.</p>
@publishTime	OD shall be present for @type='dynamic'	specifies the wall-clock time when the MPD was generated and published at the origin server. MPDs with a later value of @publishTime shall be an update as defined in subclause 5.4 to MPDs with earlier @publishTime.
@availabilityEndTime	0	specifies the latest Segment availability end time for any Segment in the Media Presentation. When not present, the value is unknown.
@mediaPresentationDuration	0	<p>specifies the duration of the entire Media Presentation. If the attribute is not present, the duration of the Media Presentation is unknown.</p> <p>This attribute shall be present when neither the attribute MPD@minimumUpdatePeriod nor the Period@duration of the last Period are present.</p>

Element or Attribute Name	Use	Description
@minimumUpdatePeriod	0	<p>If this attribute is present, it specifies the smallest period between potential changes to the MPD. This can be useful to control the frequency at which a client checks for updates.</p> <p>From a client perspective, after a client fetches an MPD, it specifies the minimum period during which the MPD remains valid. Validity is defined in subclause 5.4.</p> <p>If this attribute is not present, it indicates that the MPD does not change.</p> <p>If MPD@type is not 'dynamic', @minimumUpdatePeriod shall not be present.</p> <p>Details on the use of the value of this attribute are specified in subclause 5.4.</p>
@minBufferTime	M	specifies a common duration used in the definition of the Representation data rate (see @bandwidth attribute in subclauses 5.3.5.2 and 5.3.5.4).
@timeShiftBufferDepth	0	<p>specifies the duration of the smallest time shifting buffer for any Representation in the MPD that is guaranteed to be available for a Media Presentation with type 'dynamic'.</p> <p>When not present, the value is infinite. This value of the attribute is undefined if the type attribute is equal to 'static'.</p>
@suggestedPresentationDelay	0	<p>When @type is 'dynamic', it specifies a fixed delay offset in time from the presentation time of each access unit that is suggested to be used for presentation of each access unit. For more details, refer to subclause 7.2.1. When not specified, then no value is provided and the client is expected to choose a suitable value.</p> <p>When @type is 'static' the value of the attribute is undefined and may be ignored.</p>
@maxSegmentDuration	0	specifies the maximum duration of any Segment in any Representation in the Media Presentation, i.e. documented in this MPD and any future update of the MPD. If not present, then the maximum Segment duration shall be the maximum duration of any Segment documented in this MPD.
@maxSubsegmentDuration	0	specifies the maximum duration of any Media Subsegment in any Representation in the Media Presentation. If not present, the same value as for the maximum Segment duration is implied.

Element or Attribute Name	Use	Description
ProgramInformation	0...N	specifies descriptive information about the program. For more details, refer to the description in subclause 5.7.
BaseURL	0...N	specifies a Base URL that can be used for reference resolution and alternative URL selection. For more details, refer to the description in subclause 5.6.
Location	0...N	specifies a location at which the MPD is available. A reference processing model is provided in Annex A.11.
PatchLocation	0 ... N	specifies a location at which the MPD patch document is available. Details on the MPD patch document, this element, and expected processing models are available in subclause 5.15. If this element is present, the MPD@id attribute and the MPD@publishTime shall be present. When @type is 'static' or the @minimumUpdatePeriod attribute is not present, then value of the element is undefined and may be ignored. If this element is not present, no MPD patch document is available. If multiple elements are present, any PatchLocation element may be used.
ServiceDescription	0 ... N	specifies the service description detailing how the service provider expects the service is to be consumed. It shall provide a valid service description as described by Annex K.3 and Annex K.4. For details refer to Annex K.
InitializationSet	0 ... N	specifies a suitable initialization for a specific media type for the presentation. If present, at least one Period of the Media Presentation shall include at least one Adaptation Set that can be played when initialized by this Initialization Set. For details see subclause 5.3.12.
InitializationGroup	0 ... N	Specifies a white space separated list of ids of Initialization Sets of the same content type. This indicates that any Period in the Media Presentation has at least one Adaptation Set that conforms to one of the Initialization Sets referenced in this element. For details see subclause 5.3.12.

Element or Attribute Name	Use	Description
@id	M	specifies an unsigned integer identifier for this Initialization Group. The attribute shall be unique among all Initialization Set and Initialization Group ids in the MPD.
@profiles	0	<p>specifies the profiles which the listed Initialization Groups conform to. The value shall be a subset of the value in the MPD@profiles attribute.</p> <p>If not present the value is inferred to be the same as the MPD@profiles attribute.</p> <p>The same syntax is used as defined for the MPD@profiles attribute.</p>
@contentType	0	specifies the content type of Initialization Sets listed in the element.
InitializationPresentation	0 ... N	<p>specifies a white space separated list of ids of Initialization Sets and Initialization Groups to indicate a combination which creates a complete presentation. A client supporting all listed Initialization Sets and Initialization Groups of an Initialization Presentation is expected to be able to play the entire Media Presentation as intended by the service provider.</p> <p>For details see subclause 5.3.12.</p>
@id	M	specifies a unique unsigned integer identifier for this Initialization Presentation. The attribute shall be unique among all Initialization Presentation ids in the MPD.
@profiles	0	<p>specifies the profiles which the listed Initialization Groups conform to. The value shall be a subset of the value in the MPD@profiles attribute.</p> <p>If not present the value is inferred to be the same as the MPD@profiles attribute.</p> <p>The same syntax is used as defined for the MPD@profiles attribute.</p>
@contentType	0	specifies the content type of initialization Groups listed in the element.
ContentProtection	0 ... N	<p>specifies information about content protection and encryption schemes used in this Media Presentation. If present on this level, it shall include the @refId attribute.</p> <p>For details, see subclauses 5.8.1 and 5.8.4.1.</p>
Period	1...N	specifies the information of a Period. For more details, refer to the description in subclause 5.3.2.

Element or Attribute Name	Use	Description
Metrics	0 ... N	specifies the DASH Metrics. For more details, see subclause 5.9.
EssentialProperty	0 ... N	specifies information about the containing element that is considered essential by the Media Presentation author for processing the containing element. For details, see subclause 5.8.4.8.
SupplementalProperty	0 ... N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause 5.8.4.9.
UTCTiming	0 ... N	specifies information on ways to obtain a synchronization to wall-clock time as used in this Media Presentation. The order of the elements expresses a preference of choice by the Media Presentation author. For more details, refer to subclause 5.8.4.11.
LeapSecondInformation	0 ... 1	specifies leap second information affecting MPD timing calculations. For details refer to clause 5.13.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non- bold and preceded with an @ .		

5.3.1.3 XML syntax

```

<xs:complexType name="MPDtype">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      MPD Type
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ProgramInformation" type="ProgramInformationType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="BaseURI" type="BaseURIType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Location" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PatchLocation" type="PatchLocationType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ServiceDescription" type="ServiceDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="InitializationSet" type="InitializationSetType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="InitializationGroup" type="UIntVWithIDType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="InitializationPresentation" type="UIntVWithIDType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ContentProtection" type="ContentProtectionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Period" type="PeriodType" maxOccurs="unbounded"/>
    <xs:element name="Metrics" type="MetricsType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="EssentialProperty" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="SupplementalProperty" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="UTCTiming" type="DescriptorType" minOccurs="0"

```



```

maxOccurs="unbounded"/>
    <xs:element name="LeapSecondInformation" type="LeapSecondInformationType"
minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:string"/>
<xs:attribute name="profiles" type="ListOfProfilesType" use="required"/>
<xs:attribute name="type" type="PresentationType" default="static"/>
<xs:attribute name="availabilityStartTime" type="xs:dateTime"/>
<xs:attribute name="availabilityEndTime" type="xs:dateTime"/>
<xs:attribute name="publishTime" type="xs:dateTime"/>
<xs:attribute name="mediaPresentationDuration" type="xs:duration"/>
<xs:attribute name="minimumUpdatePeriod" type="xs:duration"/>
<xs:attribute name="minBufferTime" type="xs:duration" use="required"/>
<xs:attribute name="timeShiftBufferDepth" type="xs:duration"/>
<xs:attribute name="suggestedPresentationDelay" type="xs:duration"/>
<xs:attribute name="maxSegmentDuration" type="xs:duration"/>
<xs:attribute name="maxSubsegmentDuration" type="xs:duration"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="PresentationType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Presentation Type enumeration
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="static"/>
        <xs:enumeration value="dynamic"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="UIntVWithIDType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            UInt Vector With ID
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="UIntVectorType">
            <xs:attribute name="id" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="profiles" type="ListOfProfilesType"/>
            <xs:attribute name="contentType" type="RFC6838ContentTypeType"/>
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="UIntVectorType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Whitespace-separated list of unsigned integers
        </xs:documentation>
    </xs:annotation>
    <xs:list itemType="xs:unsignedInt"/>
</xs:simpleType>
<xs:simpleType name="ListOfProfilesType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            List of Profiles
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="(&URN;|&URL;)(,&comma_sep;(&URN;|&URL;))*"/>
    </xs:restriction>
</xs:simpleType>

```

5.3.2 Period

5.3.2.1 Overview

A Media Presentation consists of one or more Periods. A Period is defined by a **Period** element in the **MPD** element.

Three types of Periods are defined:

- Regular Period,
- Early Available Period,
- Early Terminated Period.

The type of the Period, as well as the *PeriodStart* time of a regular Period or early terminated Period is determined as follows:

- If the attribute `@start` is present in the **Period**, then the Period is a regular Period or an early terminated Period and the *PeriodStart* is equal to the value of this attribute.
- If the `@start` attribute is absent, but the previous **Period** element contains a `@duration` attribute then this new Period is also a regular Period or an early terminated Period. The start time of the new Period *PeriodStart* is the sum of the start time of the previous Period *PeriodStart* and the value of the attribute `@duration` of the previous Period.
- If (i) `@start` attribute is absent, and (ii) the **Period** element is the first in the MPD, and (iii) the **MPD@type** is 'static', then the *PeriodStart* time shall be set to zero.
- If (i) `@start` attribute is absent, and (ii) the previous **Period** element does not contain a `@duration` attribute or the **Period** element is the first in the MPD, and (iii) the **MPD@type** is 'dynamic', then this Period is an Early Available Period (see below for details).
- If (i) `@duration` attribute is present, and (ii) the next **Period** element contains a `@start` attribute or the `@minimumUpdatePeriod` is present, then this Period is an Early Terminated Period (see below for details).

For any regular Period and early terminated Period, the following holds: *PeriodStart* reflects the actual time that should elapse after playing the media of all prior Periods in this Media Presentation relative to the *PeriodStart* time of the first Period in the Media Presentation. The Period extends until the *PeriodStart* of the next Period, or until the end of the Media Presentation in the case of the last Period. For regular Periods, the difference between the *PeriodStart* time of a Period and either the *PeriodStart* time of the following Period, if this is not the last Period, or the value of the **MPD@mediaPresentationDuration** if this is the last one, is the presentation duration in Media Presentation time of the media content represented by the Representations in this Period. For Early Terminated Periods, the value of the **Period@duration** is the presentation duration in Media Presentation time of the media content represented by the Representations in this Period.

Early Available Periods may be used to advertise initialization of other non-media data before the media data itself is available. **Period** elements documenting early available Periods shall not occur before any **Period** element documenting a regular Period. For Early Available Periods, any resources that are announced in such a **Period** element shall be available. The data contained in such a **Period** element does not represent a Period in the Media Presentation. Only when the *PeriodStart* time becomes known through an update of the MPD, such a **Period** element represents a regular Period. However, an update of the MPD may even remove a **Period** element representing an Early Available Period in later updates of the MPD as long as no *PeriodStart* time is associated with the Period. To avoid dereferencing of a remote element entity containing a **Period** element solely to determine the Period timeline, e.g. in case of seeking, **Period@start** or previous Period's **Period@duration** should be present in the MPD.

The semantics of the attributes and elements within a **Period** element are provided in Table 4 of subclause 5.3.2.2. The XML syntax of the **Period** element is provided in subclause 5.3.2.3.

5.3.2.2 Semantics

Table 4 — Semantics of **Period** element

Element or Attribute Name	Use	Description
Period		specifies the information of a Period.
@xlink:href	0	specifies a reference to a remote element entity that is either empty or contains one or multiple top-level elements of type Period .
@xlink:actuate	OD default: onRequest	specifies the processing instructions, which can be either "onLoad" or "onRequest". This attribute shall not be present if the @xlink:href attribute is not present.
@id	0	specifies an identifier for this Period. The identifier shall be unique within the scope of the Media Presentation. If the MPD@type is "dynamic", then this attribute shall be present and shall not change in case the MPD is updated. If not present, no identifier for the Period is provided. In case of a remote period element, every Period@id within the remote entity has to be unique within the media presentation. If the generator of the remote entity is unaware of the Period@id values used within the media presentation, it can use unique identifiers (such as UUIDs) as values for Period@id . If the xlink resolver maintains the same generated id for every client and for each time that the remote period is requested, the remote element is cacheable.
@start	0	if present, specifies the <i>PeriodStart</i> time of the Period. The <i>PeriodStart</i> time is used as an anchor to determine the MPD start time of each Media Segment as well as to determine the presentation time of each access unit in the Media Presentation timeline. If not present, refer to the details in subclause 5.3.2.1. The value of PeriodStart, together with the value of the MPD@availabilityStartTime enables to derive the Segment availability times for dynamic media presentations. For details, refer to subclause 5.3.9.
@duration	0	if present, specifies the duration of the Period to determine the <i>PeriodStart</i> time of the next Period. If not present, refer to the details in subclause 5.3.2.1.
@bitstreamSwitching	OD Default: false	When set to 'true', this is equivalent as if the AdaptationSet@bitstreamSwitching for each Adaptation Set contained in this Period is set to 'true'. In this case, the AdaptationSet@bitstreamSwitching attribute shall not be set to 'false' for any Adaptation Set in this Period.

Element or Attribute Name	Use	Description
BaseURL	0...N	specifies a base URL that can be used for reference resolution and alternative URL selection. For more details, refer to the description in subclause 5.6.
SegmentBase	0...1	specifies default Segment Base information. Information in this element is overridden by information in AdaptationSet.SegmentBase and Representation.SegmentBase , if present. For more details, see subclause 5.3.9.
SegmentList	0...1	specifies default Segment List information. Information in this element is overridden by information in AdaptationSet.SegmentList and Representation.SegmentList , if present. For more details, see subclause 5.3.9.
SegmentTemplate	0...1	specifies default Segment Template information. Information in this element is overridden by information in AdaptationSet.SegmentTemplate and Representation.SegmentTemplate , if present. For more details, see subclause 5.3.9.
AssetIdentifier	0...1	specifies that this Period belongs to a certain asset. For more details, see subclause 5.8.5.7.
EventStream	0...N	specifies an event stream. For more details, see subclause 5.10.2.
ServiceDescription	0 ... N	specifies the service description that is intended by the service provider on how the service is expected to be consumed in this Period. For details refer to Table 3 in subclause 5.3.1.3. If any element is present in the Period, then this information overrides all Service Description information provided on MPD level with the same scope.
ContentProtection	0 ... N	specifies information about content protection and encryption schemes used in this Media Presentation. If present on this level, it shall include the @refId attribute. For details, see subclauses 5.8.1 and 5.8.4.1.
AdaptationSet	0...N	specifies an Adaptation Set. At least one Adaptation Set shall be present in each Period unless the value of the @duration attribute of the Period is set to zero. For more details, see 5.3.3.
Subset	0...N	specifies a Subset. For more details, see subclause 5.3.8.

Element or Attribute Name	Use	Description
SupplementalProperty	0...N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause 5.8.4.9.
EmptyAdaptationSet	0...N	specifies an Adaptation Set that does not contain any Representation element. The empty Adaptation Set is of the same type as a regular Adaptation Set but shall neither contain an xlink nor contain any Representation element. This element shall only be present, if an Essential Descriptor is present with @schemeIDURI set to "urn:mpeg:dash:mpd-as-linking:2015". For more details, see subclause 5.8.5.8.
GroupLabel	0 ... N	A summary label for a group of Labels. For more details, refer to subclause 5.3.10.
Preselection	0 ... N	specifies a preselection, i.e. a combination of Adaptation Sets that form a specific experience and can be selected for joint decoding and rendering. For more details, refer to subclause 5.3.11.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using xlink:href. If linking is used, then all attributes are "optional" and <minOccurs=0>. Elements are bold ; attributes are non-bold and preceded with an @.		

5.3.2.3 XML syntax

```

<xs:complexType name="PeriodType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Period
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SegmentBase" type="SegmentBaseType" minOccurs="0"/>
    <xs:element name="SegmentList" type="SegmentListType" minOccurs="0"/>
    <xs:element name="SegmentTemplate" type="SegmentTemplateType" minOccurs="0"/>
    <xs:element name="AssetIdentifier" type="DescriptorType" minOccurs="0"/>
    <xs:element name="EventStream" type="EventStreamType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ServiceDescription" type="ServiceDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ContentProtection" type="ContentProtectionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="AdaptationSet" type="AdaptationSetType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Subset" type="SubsetType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="SupplementalProperty" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="EmptyAdaptationSet" type="AdaptationSetType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="GroupLabel" type="LabelType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Preselection" type="PreselectionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"

```

```

maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute ref="xlink:href"/>
<xs:attribute ref="xlink:actuate" default="onRequest"/>
<xs:attribute ref="xlink:type" fixed="simple"/>
<xs:attribute ref="xlink:show" fixed="embed"/>
<xs:attribute name="id" type="xs:string"/>
<xs:attribute name="start" type="xs:duration"/>
<xs:attribute name="duration" type="xs:duration"/>
<xs:attribute name="bitstreamSwitching" type="xs:boolean" default="false"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.2.4 Content offering with multiple Periods

Content with multiple Periods may be created for different reasons, for example:

- to enable splicing of content, for example for ad insertion,
- to provide synchronization in segment numbering, e.g. compensate non-constant segment durations,
- to remove or add certain Representations in an Adaptation Set,
- to remove or add certain Adaptation Sets,
- to remove or add content offering on certain CDNs,
- to enable signalling of shorter segments, if produced by the encoder.

Periods provide opportunities for resync, for ad insertion, for adding and removing Representations. However, in certain circumstances, the content across Period boundaries is continuous and, in this case, continuous playout of the client is expected.

In certain circumstances, the Media Presentation is offered such that the next Period is a continuation of the content in the previous Period, possibly the immediately following Period or in a later Period (e.g. after an advertisement Period had been inserted).

The content provider may express that the media components contained in two Adaptation Sets in two different Periods are associated by assigning equivalent Asset Identifiers to both Periods and by identifying both Adaptation Sets with identical value for the attribute `@id`. Association expresses a logical continuation of the media component in the next Period and may for example be used by the client to continue playing an associated Adaptation Set in the new Period.

In addition, two Adaptation Sets in one MPD are period-continuous if all the following holds:

- The Adaptation Sets are associated.
- The sum of the value of the `@presentationTimeOffset` and the presentation duration of all Representations in one Adaptation Set are identical to the value of the `@presentationTimeOffset` of the associated Adaptation Set in the next Period.
- If Representations in both Adaptation Sets have the same value for `@id`, then they shall have functionally equivalent Initialization Segments, i.e. the Initialization Segment may be used to continue the play-out of the Representation. The concatenation of the Initialization Segment of the first Period, if present, and all consecutive Media Segments in the Representation in the first Period and subsequently the concatenation with all consecutive Media Segments in the Representation of the second Period shall represent a conforming Segment track as defined in subclause 4.5.4 conforming to the media type as specified in the `@mimeType` attribute for the Representation in the

first Period. Additionally, the @mimeType attribute for the Representation in the next Period shall be the same as one of the first Period.

Media Presentations should signal period-continuous Adaptation Sets by using a supplemental descriptor on Adaptation Set level with @schemeIdUri set to "urn:mpeg:dash:period-continuity:2015" with

- the @value of the descriptor matching the value of an @id of a Period that is contained in the MPD, and
- the value of the **AdaptationSet**@id being the same in both Periods.

MPD should signal period-continuous Adaptation Sets if the MPD contains Periods with identical Asset Identifiers.

There exist special cases for which the media in one Adaptation Set is a continuation of the previous one, but the timestamps are not continuous. Examples are timestamp wrap around, encoder reset, splicing, or other aspects. Two Adaptation Sets in one MPD are period-connected if all conditions from period-continuity from above hold, except that the timestamps across Period boundaries may be non-continuous but adjusted by the value of the @presentationTimeOffset at the Period boundary. However, for example the Initialization Segment is equivalent within the two Adaptation Sets. Media Presentations should signal period-connected Adaptation Sets by using a supplemental descriptor on Adaptation Set level with @schemeIdUri set to "urn:mpeg:dash:period-connectivity:2015".

Period continuity implies period connectivity.

For appropriate client behaviour, please refer to A.9.

5.3.2.5 Special Periods in Live Services

In order to address live services for which Periods are extended into the future and Periods are gradually removed due to the timeshift buffer depth, special types of Periods are defined as follows:

- *Partially Unavailable Period*: any Period in a dynamic Media Presentation for which the sum of the **MPD**@availabilityStartTime and the value of **Period**@start is smaller than the value of @publishTime minus the value of **MPD**@timeShiftBufferDepth. For this Period, some Segments may no longer be available as they have been fallen out of the time shift buffer window.
- *Live-Edge Period*: newest regular Period in a dynamic Media Presentation, for which the Period duration is unknown, and Segment availability start time of at least one Segment is greater than the value of **MPD**@publishTime. For such Periods, the exact duration of Segments that are not yet available, may be unknown.

Note that an MPD may contain both types of Periods, and further, a single Period may be both at the same time, a Partially Unavailable Period and a Live-Edge Period.

5.3.3 Adaptation Sets

5.3.3.1 Overview

Each Period consists of one or more Adaptation Sets. An Adaptation Set is described by an **AdaptationSet** element. **AdaptationSet** elements are contained in a **Period** element.

An Adaptation Set contains alternate Representations, i.e. only one Representation within an Adaptation Set is expected to be presented at a time. All Representations contained in one Adaptation Set represent the same media content components and therefore contain media streams that are considered to be perceptually equivalent. The Adaptation Set and the contained Representations shall be prepared and

contain sufficient information such that seamless switching (as defined in subclause 4.5.1) across different Representations in one Adaptation Set is enabled. If an Adaptation Set is expected to be consumed by DASH Clients with restrictions in terms of switching, then the Media Presentation author should provide sufficient means to enable seamless switching under these restrictions.

Representations are arranged into Adaptation Sets according to the media content component properties of the media content components present in the Representations, namely:

- the language as described by the `@lang` attribute,
- the media content component type described by the `@contentType` attribute,
- the picture aspect ratio as described by the `@par` attribute,
- the role property as described by the **Role** elements,
- the accessibility property as described by the **Accessibility** elements,
- the viewpoint property as described by the **Viewpoint** elements,
- the rating property as described by the **Rating** elements.

Representations shall appear in the same Adaptation Set if and only if they have identical values for all of these media content component properties for each media content component.

The values for the elements **Role**, **Accessibility**, **Viewpoint** and **Rating** are generally not provided within the scope of this document. However, a number of simple schemes are defined in subclause 5.8.5.

If there exist multiple media content components, then the properties of each media content component shall be described by a separate **ContentComponent** element as defined in subclause 5.3.4. The **ContentComponent** element shares common elements and attributes with the **AdaptationSet** element. Default values, or values applicable to all media content components, may be provided directly in the **AdaptationSet** element. Attributes present in the **AdaptationSet** shall not be repeated in the **ContentComponent** element.

The **AdaptationSet** element may contain default values for elements and attributes associated to the contained Representations. The list of possible present elements and attributes that are common to **AdaptationSet** and **Representation** (and also **SubRepresentation**) are collected in subclause 5.3.7. Any of the common attributes shall only be present either in the **AdaptationSet** element or in the **Representation** element, but not in both.

The **AdaptationSet** element also supports the description of ranges for the `@bandwidth`, `@width`, `@height` and `@frameRate` attributes associated to the contained Representations, which provide a summary of all values for all the Representations within this Adaptation Set. The Representations contained within an Adaptation Set shall not contain values outside the ranges documented for that Adaptation Set.

Adaptation Sets may be further arranged into groups using the `@group` attribute. The semantics of this grouping is that the media content within one Period is represented by:

- 1) either one Representation from group 0, if present,
- 2) or the combination of at most one Representation from each non-zero group.

If the **AdaptationSet**@group attribute is not present, then all Representations in this Adaptation Set are assigned to a non-zero group specific to this Adaptation Set.

The semantics of the attributes and elements within an **AdaptationSet** element are provided in Table 5 in subclause 5.3.3.2. The XML syntax of the **AdaptationSet** element is provided in subclause 5.3.3.3.

5.3.3.2 Semantics

Table 5 — Semantics of **AdaptationSet** element

Element or Attribute Name	Use	Description
AdaptationSet		Adaptation Set description.
@xlink:href	0	specifies a reference to a remote element entity that shall contain exactly one element of type AdaptationSet .
@xlink:actuate	OD default: 'onRequest'	specifies the processing instructions, which can be either "onLoad" or "onRequest".
@id	0	specifies a unique identifier for this Adaptation Set in the scope of the Period. The attribute shall be a unique unsigned integer value in the scope of the containing Period. The attribute shall not be present in a remote element entity. If not present, no identifier for the Adaptation Set is specified.
@group	0	specifies an identifier for the group that is unique in the scope of the containing Period. The value is an unsigned integer. For details, refer to subclause 5.3.3.1.
CommonAttributesElements	—	specifies the common attributes and elements (attributes and elements from base type RepresentationBaseType). For details, see subclause 5.3.7.
@lang	0	Declares the language code for this Adaptation Set. The syntax and semantics according to IETF RFC 5646 shall be used. If not present, the language code may be defined for each media component or it may be unknown. If the language is unknown, the 'und' code for undetermined primary language or the 'zxx' (Non-Linguistic, Not Applicable) code can be used.

Element or Attribute Name	Use	Description
@contentType	0	<p>specifies the media content component type for this Adaptation Set. A value of the top-level Content-type 'type' value as defined in 4 of IETF RFC 6838:2013 shall be taken.</p> <p>If not present, the media content component type may be defined for each media content component or it may be unknown.</p>
@par	0	<p>specifies the picture aspect ratio of the video media component type, in the form of a string consisting of two integers separated by ':', e.g. "16:9". When this attribute is present, and the attributes @width and @height for the set of Representations are also present, the picture aspect ratio as specified by this attribute shall be the same as indicated by the values of @width, @height, and @sar, i.e. it shall express the same ratio as $(@width * sar_x) : (@height * sar_y)$, with sar_x the first number in @sar and sar_y the second number.</p> <p>If not present, the picture aspect ratio may be defined for each media component or it may be unknown.</p>
@minBandwidth	0	<p>specifies the minimum @bandwidth value in all Representations in this Adaptation Set. This value has the same units as the @bandwidth attribute.</p> <p>If not present, the value is unknown.</p>
@maxBandwidth	0	<p>specifies the maximum @bandwidth value in all Representations in this Adaptation Set. This value has the same units as the @bandwidth attribute.</p> <p>If not present, the value is unknown.</p>
@minWidth	0	<p>specifies the minimum @width value in all Representations in this Adaptation Set. This value has the same units as the @width attribute.</p> <p>If not present, the value is unknown.</p>
@maxWidth	0	<p>specifies the maximum @width value in all Representations in this Adaptation Set. This value has the same units as the @width attribute.</p> <p>If not present, the value is unknown.</p>
@minHeight	0	<p>specifies the minimum @height value in all Representations in this Adaptation Set. This value has the same units as the @height attribute.</p> <p>If not present, the value is unknown.</p>
@maxHeight	0	<p>specifies the maximum @height value in all Representations in this Adaptation Set. This value has the same units as the @height attribute.</p> <p>If not present, the value is unknown.</p>

Element or Attribute Name	Use	Description
@minFrameRate	0	<p>specifies the minimum @framerate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute.</p> <p>If not present, the value is unknown.</p>
@maxFrameRate	0	<p>specifies the maximum @framerate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute.</p> <p>If not present, the value is unknown.</p>
@segmentAlignment	OD default: false	<p>When set to 'true', this specifies that for any two Representations, X and Y, within the same Adaptation Set, the m-th Segment of X and the n-th Segment of Y are non-overlapping (as defined in subclause 4.5.3) whenever m is not equal to n.</p> <p>For Adaptation Sets containing Representations with multiple media content components, this attribute value shall be either 'true' or 'false'.</p> <p>NOTE Previous editions permitted non-boolean values for this attribute. As this was not used in practice and caused confusion, it is deprecated in this edition.</p>
@bitstreamSwitching	0	<p>When this flag is set to 'true', the following applies:</p> <ul style="list-style-type: none"> — All Representations in the Adaptation Set shall have the same number M of Media Segments; — Let R_1, R_2, \dots, R_N be all the Representations within the Adaptation Set. — Let <ul style="list-style-type: none"> — $S_{i,j}$, for $j > 0$, be the j^{th} Media Segment in the i^{th} Representation (i.e. R_i) — if present, let $S_{i,0}$ be the Initialization Segment in the i^{th} Representation, and — if present, let B_i be the Bitstream Switching Segment in the i^{th} Representation. — The sequence of <ul style="list-style-type: none"> — any Initialization Segment, if present, in the Adaptation Set, with,

Element or Attribute Name	Use	Description
		<p>— if Bitstream Switching Segments are present,</p> $B_{i(1)}, S_{i(1),1}, B_{i(2)}, S_{i(2),2}, \dots, B_{i(k)}, S_{i(k),k}, \dots, B_{i(M)}, S_{i(M),M}$ <p>— else</p> $S_{i(1),1}, S_{i(2),2}, \dots, S_{i(k),k}, \dots, S_{i(M),M},$ <p>wherein any $i(k)$ for all k values in the range of 1 to M, respectively, is an integer value in the range of 1 to N,</p> <p>results in a "conforming Segment track" as defined in subclause 4.5.4 with the media format as specified in the @mimeType attribute.</p> <p>More detailed rules may be defined for specific media formats.</p> <p>For more details, refer to subclause 0.</p> <p>NOTE When this attribute is set to TRUE, then seamless switching across Representations can be achieved without re-initialization of the decoder. Content authors are encouraged to set this attribute to true only if the media content components across Representations do not need the media decoder to be re-initialized.</p>
@subsegmentAlignment	OD default: false	<p>If the @subsegmentAlignment for an Adaptation Set is set to 'true', all following conditions shall be satisfied:</p> <p>— Each Media Segment shall be indexed (i.e. either it contains a Segment index or there is an Index Segment providing an index for the Media Segment).</p> <p>— For any two Representations, X and Y, within the same Adaptation Set, the m-th Subsegment of X and the n-th Subsegment of Y are non-overlapping (as defined in subclause 4.5.3) whenever m is not equal to n.</p> <p>NOTE Previous editions permitted non-boolean values for this attribute. As this was not used in practice and caused confusion, it is deprecated in this edition.</p>

Element or Attribute Name	Use	Description
@subsegmentStartsWithSAP	OD default: 0	when greater than 0, specifies that each Subsegment with SAP_type greater than 0 starts with a SAP of type less than or equal to the value of @subsegmentStartsWithSAP. A Subsegment starts with SAP when the Subsegment contains a SAP, and for the first SAP, I _{SAU} is the index of the first access unit that follows I _{SAP} , and I _{SAP} is contained in the Subsegment. The semantics of @subsegmentStartsWithSAP equal to 0 are unspecified.
@initializationSetRef	0	specifies a white space separated list of Initialization Set identifiers. The Adaptation Set shall conform to all Initialization Sets that are referenced in this attribute.
@initializationPrincipal	0	specifies the URL of an Initialization Segment that is sufficient to initialize the Adaptation Set. If not present, an Initialization Segment from one of the Representations is sufficient.
Accessibility	0 ... N	specifies information about accessibility scheme For more details, refer to subclauses 5.8.1 and 5.8.4.3.
Role	0 ... N	specifies information on role annotation scheme For more details, refer to subclauses 5.8.1 and 5.8.4.2.
Rating	0 ... N	specifies information on rating scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.4.
Viewpoint	0 ... N	specifies information on viewpoint annotation scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.5.
ContentComponent	0...N	specifies the properties of one media content component contained in this Adaptation Set. For more details, refer to subclause 5.3.4.
BaseURL	0...N	specifies a base URL that can be used for reference resolution and alternative URL selection. For more details, refer to the description in subclause 5.6.
SegmentBase	0...1	specifies default Segment Base information. Information in this element is overridden by information in the Representation.SegmentBase , if present. For more details, see subclause 5.3.9.
SegmentList	0...1	specifies default Segment List information. Information in this element is overridden by information in the Representation.SegmentList , if present. For more details, see subclause 5.3.9.

Element or Attribute Name	Use	Description
SegmentTemplate	0..1	<p>specifies default Segment Template information.</p> <p>Information in this element is overridden by information in the Representation.SegmentTemplate, if present.</p> <p>For more details, see subclause 5.3.9.</p>
Representation	0 ... N	<p>specifies a Representation.</p> <p>At least one Representation element shall be present in each Adaptation Set. The actual element may however be part of a remote element entity if xlink is used on the containing AdaptationSet element.</p> <p>For more details, refer to subclause 5.3.5.</p>

Key

For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed

For elements: <minOccurs>...<maxOccurs> (N=unbounded)

The conditions only hold without using `xlink:href`. If linking is used, then all attributes are "optional" and <minOccurs=0>.

Elements are **bold**; attributes are non-bold and preceded with an @; list of elements and attributes is in *italics* **bold** referring to those taken from the Base type that has been extended by this type.

5.3.3.3 XML syntax

```

<xs:complexType name="AdaptationSetType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Adaptation Set
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="Accessibility" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Role" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Rating" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Viewpoint" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ContentComponent" type="ContentComponentType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="BaseURL" type="BaseURLType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="SegmentBase" type="SegmentBaseType" minOccurs="0"/>
        <xs:element name="SegmentList" type="SegmentListType" minOccurs="0"/>
        <xs:element name="SegmentTemplate" type="SegmentTemplateType"
minOccurs="0"/>
        <xs:element name="Representation" type="RepresentationType"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="xlink:href"/>
      <xs:attribute ref="xlink:actuate" default="onRequest"/>
      <xs:attribute ref="xlink:type" fixed="simple"/>
      <xs:attribute ref="xlink:show" fixed="embed"/>
      <xs:attribute name="id" type="xs:unsignedInt"/>
      <xs:attribute name="group" type="xs:unsignedInt"/>
      <xs:attribute name="lang" type="xs:language"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="contentType" type="RFC6838ContentTypeType"/>
        <xs:attribute name="par" type="RatioType"/>
        <xs:attribute name="minBandwidth" type="xs:unsignedInt"/>
        <xs:attribute name="maxBandwidth" type="xs:unsignedInt"/>
        <xs:attribute name="minWidth" type="xs:unsignedInt"/>
        <xs:attribute name="maxWidth" type="xs:unsignedInt"/>
        <xs:attribute name="minHeight" type="xs:unsignedInt"/>
        <xs:attribute name="maxHeight" type="xs:unsignedInt"/>
        <xs:attribute name="minFrameRate" type="FrameRateType"/>
        <xs:attribute name="maxFrameRate" type="FrameRateType"/>
        <xs:attribute name="segmentAlignment" type="xs:boolean" default="false"/>
        <xs:attribute name="subsegmentAlignment" type="xs:boolean" default="false"/>
        <xs:attribute name="subsegmentStartsWithSAP" type="SAPType" default="0"/>
        <xs:attribute name="bitstreamSwitching" type="xs:boolean"/>
        <xs:attribute name="initializationSetRef" type="UIntVectorType"/>
        <xs:attribute name="initializationPrincipal" type="xs:anyURI"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:simpleType name="RatioType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Ratio Type for sar and par
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]*:[0-9]*"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FrameRateType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Type for Frame Rate
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]+(/[1-9][0-9]*)?"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RFC6838ContentTypeType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Type for RFC6838 Content Type
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="text"/>
        <xs:enumeration value="image"/>
        <xs:enumeration value="audio"/>
        <xs:enumeration value="video"/>
        <xs:enumeration value="application"/>
        <xs:enumeration value="font"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SAPType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Stream Access Point type enumeration
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="6"/>
    </xs:restriction>
</xs:simpleType>

```

5.3.3.4 Switching within Adaptation Sets

Switching refers to the presentation of decoded data from one Representation up to a certain time t , and presentation of decoded data of another Representation from time t onwards; for details, refer to subclause 4.3.

The **Switching** element as defined in Table 6 provides instructions of switch points within an Adaptation Set and the permitted switching options as defined in Table 7. When this element is present, it signals opportunities for simple switching across Representations in one Adaptation Set. This element may be used instead of the attributes `@segmentAlignment` or `@bitstreamSwitching`.

Table 7 defines different switching strategies that provide instructions to the client on the procedures to switch appropriately within an Adaptation Set.

Table 6 — Switch point signalling

Element or Attribute Name	Use	Description
Switching		Switching logic description for the associated Representation
<code>@interval</code>	M	<p>specifies the interval between two switching points in the scale of the <code>@timescale</code> on Representation level. Any Segment for which the earliest presentation time minus the <code>@t</code> value of the S element describing the segment is an integer multiple of the product of <code>@timescale</code> and <code>@interval</code> is a switch-to opportunity, i.e. it enables to switch to this Representation with the switching strategy as defined by the <code>@type</code> value.</p> <p>The value should be chosen such that the resulting time matches MPD start time of segments, otherwise no switching will be described.</p>
<code>@type</code>	OD default: 'media'	specifies the switching strategy for the switch points identified in by the <code>@interval</code> attribute. Switching strategies are defined in Table 7.

Table 7 — Switching strategies

Type	Description
media	Media level switching: in this case, switching is possible at the switch point by decoding and presenting switch-from Representation up to switch point t , initializing the switch-to Representation with the associated Initialization Segment and continue decoding and presenting the switch-to Representation from time t onwards.
bitstream	<p>Bitstream switching: in this case, switching is possible at the switch point by decoding and presenting switch-from Representation up to switch point t, and continue decoding and presenting the switch-to Representation from time t onwards. More specifically, the concatenation of two Representations at the switch point results in a "conforming Segment track" as defined in 4.5.4 with the media format as specified in the <code>@mimeType</code> attribute.</p> <p>Initialization of the switch-to Representation is not necessary and is not recommended.</p> <p>In order to enable this feature, it is recommended to use the same Initialization Segment for all Representations in the Adaptation Set, i.e. the highest profile/level is signalled in the Initialization Segment.</p>

The XML schema snippet is as follows:

```
<xs:complexType name="SwitchingType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Switching
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="interval" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="type" type="SwitchingTypeType" default="media"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="SwitchingTypeType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Switching Type type enumeration
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="media"/>
    <xs:enumeration value="bitstream"/>
  </xs:restriction>
</xs:simpleType>
```

5.3.3.5 Switching across Adaptation Sets

Representations in two or more Adaptation Sets may provide the same content or the content may be offered such that seamless switching is desired (for example in case of multiple viewpoints). In addition, the content may be time-aligned and may be offered such that seamless switching across Representations in different Adaptation Sets is simplified. Typical examples are the offering of the same content with different codecs, for example H.264/AVC and H.265/HEVC and the content author wants to provide such information to the receiver in order to seamlessly switch Representations (as defined in subclause 4.5.1) across different Adaptation Sets.

A content author may signal such seamless switching property across Adaptation Sets by providing a Supplemental Descriptor along with an Adaptation Set with @schemeIdURI set to urn:mpeg:dash:adaptation-set-switching:2016 and the @value is a comma-separated list of Adaptation Set IDs that may be seamlessly switched to from this Adaptation Set.

If the content author signals the ability of Adaptation Set switching and as @segmentAlignment or @subsegmentAlignment are set to TRUE, the (Sub)Segment alignment element shall be valid for *all* Representations in *all* Adaptation Sets for which the @id value is included in the @value attribute of the Supplemental descriptor.

For Adaptation Sets containing Representations, when two Adaptation Sets signal Adaptation Set switching, then for any two Representations, X and Y, within the union of the two Adaptation Sets, the *m*-th Subsegment of X and the *n*-th Subsegment of Y shall be non-overlapping (as defined in subclause 4.5.3) whenever *m* is not equal to *n*.

If the content author signals the ability of Adaptation Set switching and **Switching** element is provided, the signalled switch points apply for *all* Representations in *all* Adaptation Sets for which the @id value is included in the @value attribute of the Supplemental descriptor.

As an example, a content author may signal that seamless switching across an H.264/AVC Adaptation Set with **AdaptationSet**@id="4" and an HEVC Adaptation Set with **AdaptationSet**@id="5" is possible by adding a Supplemental Descriptor to the H.264/AVC Adaptation Set with @schemeIdURI set to urn:mpeg:dash:adaptation-set-switching:2016 and the @value="5" and by adding a Supplemental Descriptor to the HEVC Adaptation Set with @schemeIdURI set to urn:mpeg:dash:adaptation-set-switching:2016 and the @value="4".

In addition, if the content author signals the ability of Adaptation Set switching for any Adaptation Sets then the parameters as defined for an Adaption Set shall also hold for all Adaptation Sets that are included

in the @value attribute. This constraint may result that the switching may only be signalled with one Adaptation Set, but not with both as for example one Adaptation Set signalling may include all spatial resolutions of another one, whereas it is not the case the other way around.

5.3.4 Media content component

5.3.4.1 Overview

Each Adaptation Set contains one or more media content components. The properties of each media content component are described by a **ContentComponent** element or may be described directly on the **AdaptationSet** element if only one media content component is present in the Adaptation Set. **ContentComponent** elements are contained in an **AdaptationSet** element.

The semantics of the attributes and elements within a **ContentComponent** element are provided in Table 8 in subclause 5.3.4.2. The XML syntax of the **ContentComponent** element is provided in subclause 5.3.4.3.

5.3.4.2 Semantics

Table 8 — Semantics of ContentComponent element

Element or Attribute Name	Use	Description
ContentComponent		Description of a content component.
@id	0	<p>specifies an identifier for this media component. The attribute shall be unique in the scope of the containing Adaptation Set.</p> <p>The value of this attribute should be the media component identifier in the media segment (i.e. the Track Id in ISO BMFF segments and PID in MPEG-2 TS segments) described by this ContentComponent element.</p> <p>Multiplexing of media components may also be provided on elementary stream level, for example for next generation audio codecs, different audio components may be mixed. In this case, the id may also point to specific components in a single elementary stream.</p>
@lang	0	Same semantics as in Table 5 for @lang attribute.
@contentType	0	Same semantics as in Table 5 for @contentType attribute.
@par	0	Same semantics as in Table 5 for @par attribute.
@tag	0	specifies the tag of the Content Component which may be used for selection purposes towards the decoder.
Accessibility	0 ... N	Same semantics as in Table 5 for Accessibility element.
Role	0 ... N	Same semantics as in Table 5 for Role element.
Rating	0 ... N	Same semantics as in Table 5 for Rating element.
Viewpoint	0 ... N	Same semantics as in Table 5 for Viewpoint element.

Element or Attribute Name	Use	Description
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @; list of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

5.3.4.3 XML syntax

```

<xs:complexType name="ContentComponentType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Content Component
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Accessibility" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Role" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Rating" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Viewpoint" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedInt"/>
  <xs:attribute name="lang" type="xs:language"/>
  <xs:attribute name="contentType" type="RFC6838ContentTypeType"/>
  <xs:attribute name="par" type="RatioType"/>
  <xs:attribute name="tag" type="TagType"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="TagType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Tag
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>

```

5.3.5 Representation

5.3.5.1 Overview

Representations are described by the **Representation** element. **Representation** elements are contained in an **AdaptationSet** element.

A Representation is one of the alternative choices of the complete set or subset of media content components comprising the media content during the defined Period.

A Representation starts at the start of the Period *PeriodStart* and continues to the end of the Period, i.e. the start of the next Period or the end of the Media Presentation.

Each Representation includes one or more media streams, where each media stream is an encoded version of one media content component.

A Representation consists of one or more Segments.

Each Representation either shall contain an Initialization Segment or each Media Segment in the Representation shall be self-initializing, i.e. the Media Segment itself conforms to the media type as specified in the @mimeType attribute for this Representation.

When a Representation is not a dependent Representation, i.e. the `@dependencyId` attribute is absent, then concatenation of the Initialization Segment, if present, and all consecutive Media Segments in one Representation shall represent a conforming Segment track as defined in subclause 4.5.4 conforming to the media type as specified in the `@mimeType` attribute for this Representation.

Dependent Representations are described by a **Representation** element that contains a `@dependencyId` attribute. Dependent Representations are regular Representations except that they depend on a set of complementary Representations for decoding and/or presentation. The `@dependencyId` contains the values of the `@id` attribute of all the complementary Representations, i.e. Representations that are necessary to present and/or decode the media content components contained in this dependent Representation.

For any dependent Representation X that depends on complementary Representation Y, the m -th Subsegment of X and the n -th Subsegment of Y shall be non-overlapping (as defined in subclause 4.5.3) whenever m is not equal to n . For dependent Representations, the concatenation of the Initialization Segment with the sequence of Subsegments of the dependent Representations, each being preceded by the corresponding Subsegment of each of the complementary Representations in order as provided in the `@dependencyId` attribute, shall represent a conforming Subsegment track as defined in subclause 4.5.4 conforming to the media format as specified in the `@mimeType` attribute for this dependent Representation.

NOTE 1 When decoding of a dependent Representation is started from a SAP in the (Sub)Segment with number i , the decoding process does not need to access data from the complementary Representation(s) from any earlier (sub)segments than (sub)Segment with number i of the complementary Representation(s).

NOTE 2 In case a dependent Representation X depends on at least two complementary Representations Y_i , that are also dependent Representations depending, directly or indirectly, on the same complementary Representation Z, then the concatenation as defined above results in a sequence conforming to the media format if the Segments of Representation Z are concatenated only the first time encountered following the order in the `@dependencyId` attributes starting from X. If following recursively the dependencies indicated in the `@dependencyId` of Representations, certain Representations can be encountered more than once. In this case, the corresponding Segments or Subsegments are expected to only be concatenated once, namely when first encountered.

Associated Representations are described by a **Representation** element that contains an `@associationId` attribute and optionally an `@associationType` attribute. Associated Representations are Representations that provide information on their relationships with other Representations. As opposed to complementary Representations, the segments of an associated Representation may be optional for decoding and/or presentation of the Representations identified by `@associationId`. They can be considered as supplementary or descriptive information, the type of the association being specified by the `@associationType` attribute.

NOTE 3 `@associationId` and `@associationType` attributes can only be used between Representations that are not in the same Adaptation Sets.

If a Representation is offered in a Media Presentation with `MPD@type= 'dynamic'`, it is recommended that means to compensate such drift be included. For more details, refer to A.8.

The semantics of the attributes and elements within a Representation are provided in Table 9 in subclause 5.3.5.2. The XML syntax of the Representation type is provided in subclause 5.3.5.3.

5.3.5.2 Semantics

Table 9 — Semantics of Representation element

Element or Attribute Name	Use	Description
Representation		This element contains a description of a Representation.
@id	M	<p>specifies an identifier for this Representation. The identifier shall be unique within a Period unless the Representation is functionally identical to another Representation in the same Period.</p> <p>The identifier shall not contain whitespace characters.</p> <p>If used in the template-based URL construction as defined in subclause 5.3.9.4.4, the string shall only contain characters that are permitted within an HTTP-URL according to IETF RFC 3986.</p>
@bandwidth	M	<p>Consider a hypothetical constant bitrate channel of bandwidth with the value of this attribute in bits per second (bps). Then, if the Representation is continuously delivered at this bitrate, starting at any SAP that is indicated either by @startWithSAP or by any Segment Index box, a client can be assured of having enough data for continuous playout providing playout begins after</p> $@minBufferTime * @bandwidth \text{ bits}$ <p>have been received (i.e. at time @minBufferTime after the first bit is received).</p> <p>For dependent Representations, this value specifies the bandwidth according to the above definition for the aggregation of this Representation and all complementary Representations.</p> <p>For details, see subclause 5.3.5.4.</p>
@qualityRanking	O	specifies a quality ranking of the Representation relative to other Representations in the same Adaptation Set. Lower values represent higher quality content. If not present, then no ranking is defined.
@dependencyId	O	<p>specifies all complementary Representations the Representation depends on in the decoding and/or presentation process as a whitespace-separated list of values of @id attributes.</p> <p>If not present, the Representation can be decoded and presented independently of any other Representation.</p> <p>This attribute shall not be present where there are no dependencies.</p>
@associationId	O	specifies all Representations the Representation is associated with in the decoding and/or presentation process as a whitespace-separated list of values of Representation @id attributes.

Element or Attribute Name	Use	Description
@associationType	0	<p>specifies, as a whitespace-separated list of values, the kind of association for each Representation the Representation has been associated with through the @associationId attribute. Values taken by this attribute are 4 character codes (4CCs) for track reference types registered in MP4 registration authority.</p> <p>This attribute shall not be present when @associationId is not present.</p> <p>When present, this attribute shall have as many values as the number of identifiers declared in the @associationId attribute.</p>
@mediaStreamStructureId	0	<p>The attribute may be present for Representations containing video and its semantics are unspecified for any other type of Representations.</p> <p>If present, the attribute @mediaStreamStructureId specifies a whitespace-separated list of media stream structure identifier values. If media streams share the same media stream structure identifier value, the media streams shall have the following characteristics:</p> <ul style="list-style-type: none"> — The media streams have the same number of stream access points of type 1 to 3. — The values of T_{SAP}, T_{DEC}, T_{EPT}, and T_{PTF} of the i-th SAP of type 1 to 3 in one media stream are identical to the values of T_{SAP}, T_{DEC}, T_{EPT}, and T_{PTF}, respectively, of the i-th SAP of type 1 to 3 in the other media streams for any value of i from 1 to the number of SAPs of type 1 to 3 in any of the media streams. — A media stream formed by concatenating the media stream of a first Representation until I_{SAU} (exclusive) of the i-th SAP of type 1 to 3 and the media stream of a second Representation (having the same media stream structure identifier value as for the first Representation) starting from the I_{SAU} (inclusive) of the i-th SAP of type 1 to 3 conforms to the specification in which the media stream format is specified for any value of i from 1 to the number of SAPs of type 1 to 3 in either media stream. Furthermore, the decoded pictures have an acceptable quality regardless of type of the stream access point access unit used.

Element or Attribute Name	Use	Description
		<p>All media stream structure identifier values for one Adaptation Set shall differ from those of another Adaptation Set.</p> <p>If not present, then for this Representation no similarities to other Representations are known.</p> <p>Indicating multiple media stream structure identifier values for a Representation can be useful in cases where switching between Representations A and B as well as between Representations B and C is allowed at non-IDR intra pictures, but switching between Representations A and C would cause too severe a degradation in the quality of the leading pictures and is hence not allowed. To indicate these permissions and restrictions, Representation A would contain @mediaStreamStructureId equal to "1", Representation B would contain @mediaStreamStructureId equal to "1 2", and Representation C would contain @mediaStreamStructureId equal to "2"</p>
<i>CommonAttributesElements</i>	—	Common Attributes and Elements (attributes and elements from base type RepresentationBaseType). For more details, see subclause 5.3.7.
BaseURL	0...N	specifies a Base URL that can be used for reference resolution and alternative URL selection. For more details, refer to the description in subclause 5.6.
ExtendedBandwidth	0...N	<p>specifies an extended bandwidth model with more detailed information on the characteristics of the Representation.</p> <p>For more details, see subclause 5.3.5.6.</p>
SubRepresentation	0 ... N	<p>specifies information about a Sub-Representation that is embedded in the containing Representation.</p> <p>For more details, see subclause 5.3.5.6.</p>
SegmentBase	0...1	<p>specifies default Segment Base information.</p> <p>For more details, see subclause 5.3.9.</p>
SegmentList	0 ... 1	<p>specifies the Segment List information.</p> <p>For more details, see subclause 5.3.9.</p>
SegmentTemplate	0 ... 1	<p>specifies the Segment Template information.</p> <p>For more details, see subclause 5.3.9.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @; list of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.</p>		

5.3.5.3 XML syntax

```

<xs:complexType name="RepresentationType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Representation
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="BaseURL" type="BaseURLType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="ExtendedBandwidth" type="ExtendedBandwidthType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SubRepresentation" type="SubRepresentationType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SegmentBase" type="SegmentBaseType" minOccurs="0"/>
        <xs:element name="SegmentList" type="SegmentListType" minOccurs="0"/>
        <xs:element name="SegmentTemplate" type="SegmentTemplateType"
minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="id" type="StringNoWhitespaceType" use="required"/>
      <xs:attribute name="bandwidth" type="xs:unsignedInt" use="required"/>
      <xs:attribute name="qualityRanking" type="xs:unsignedInt"/>
      <xs:attribute name="dependencyId" type="StringVectorType"/>
      <xs:attribute name="associationId" type="StringVectorType"/>
      <xs:attribute name="associationType" type="ListOf4CCType"/>
      <xs:attribute name="mediaStreamStructureId" type="StringVectorType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="StringNoWhitespaceType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      String without white spaces
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="^[^r\n\t \p{Z}]*"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StringVectorType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Whitespace-separated list of strings
    </xs:documentation>
  </xs:annotation>
  <xs:list itemType="xs:string"/>
</xs:simpleType>
<xs:simpleType name="ListOf4CCType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Whitespace separated list of 4CC
    </xs:documentation>
  </xs:annotation>
  <xs:list itemType="FourCCType"/>
</xs:simpleType>
<xs:simpleType name="FourCCType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      4CC as per latest 14496-12
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>

```

5.3.5.4 Relation of Bandwidth and Minimum Buffer Time attributes

The MPD contains a pair of values for a bandwidth and buffering description, namely the Minimum Buffer Time (MBT) expressed by the value of **MPD**@minBufferTime and bandwidth (BW) expressed by the value of **Representation**@bandwidth. The following holds:

- The value of the minimum buffer time does not provide any instructions to the client on how long to buffer the media. The value however describes how much buffer a client should have under ideal network conditions. As such, MBT is not describing the burstiness or jitter in the network, it is describing the burstiness or jitter in the content encoding. Together with the BW value, it is a property of the content. Using the "leaky bucket" model, it is the size of the bucket that makes BW true, given the way the content is encoded.
- The minimum buffer time provides information that for each representation, the following shall be true: if the Representation (starting at any segment) is delivered over a constant bitrate channel with bitrate equal to value of the BW attribute then each access unit with presentation time PT is available at the client latest at time with a delay of at most $PT + MBT$.
- In the absence of any other guidance, the MBT should be set to the maximum GOP size (coded video sequence) of the content, which quite often is identical to the maximum segment duration for the live profile or the maximum subsegment duration for the On-Demand profile. The MBT may be set to a smaller value than maximum (sub)segment duration but should not be set to a higher value.

5.3.5.5 Random Access to Representations

Random Access refers to start processing, decoding and presenting the Representation from the random access point at time t onwards by initializing the Representation with the Initialization Segment, if present and decoding and presenting the Representation from the signalled Segment onwards. Random Access point may be signalled with the **RandomAccess** element as defined in Table 10.

Table 11 provides different random access point types.

Table 10 — Random Access Signalling

Element or Attribute Name	Use	Description
RandomAccess		Random Access Information
@interval	M	<p>specifies the position of the random access points in the Representations. The information is specified in the scale of the @timescale on Representation level. Any Segment for which the MPD start time minus the @t value of the S element describing the segment is an integer multiple of the product of @timescale and @interval is a random access opportunity, i.e. it enables randomly access to this Representation with the random access strategy as defined by the @type value.</p> <p>The value should be chosen such that the resulting time matches MPD start time of segments, otherwise no random access will be described.</p>
@type	OD default: "closed"	<p>specifies the random access strategy for the random access points in by the @interval attribute.</p> <p>The value shall use a type present in Table 11.</p> <p>If the value of the type is unknown, the DASH Client is expected to ignore the containing Random Access element.</p>
@minBufferTime	O	<p>specifies a common duration used in the definition of the Representation data rate (see @bandwidth attribute in subclauses 0 and 5.3.5.4).</p> <p>If not present, then the value of the MPD level is inherited.</p>
@bandwidth	O	<p>Consider a hypothetical constant bitrate channel of bandwidth with the value of this attribute in bits per second (bps). Then, if the Representation is continuously delivered at this bitrate, starting at any RAP indicated in this element a client can be assured of having enough data for continuous playout providing playout begins after $@minBufferTime * @bandwidth$ bits have been received (i.e. at time @minBufferTime after the first bit is received).</p> <p>For dependent Representations, this value specifies the bandwidth according to the above definition for the aggregation of this Representation and all complementary Representations.</p> <p>For details, see subclause 5.3.5.4.</p> <p>If not present, the value of the Representation is inherited.</p>

Table 11 — Random Access Strategies

Type	Informative description
closed	Closed GOP random access. This implies that the segment is a Random Access Segment as well as the segment starts with a SAP type of 1 or 2. SAP type 1 or 2 is a necessary condition, but not sufficient. In addition, all requirements of a Random Access Segment need to be fulfilled.
open	Open GOP random access. This implies that the segment is a Random Access Segment as well as the segment starts with a SAP type of 1, 2 or 3. SAP type 1, 2 or 3 is a necessary condition, but not sufficient. In addition, all requirements of a Random Access Segment need to be fulfilled.
gradual	Gradual decoder refresh random access. This implies that the segment is a Random Access Segment as well as the segment starts with a SAP type of 1, 2, 3 or 4. SAP type 1, 2, 3 or 4 is a necessary condition, but not sufficient. In addition, all requirements of a Random Access Segment need to be fulfilled.

The XML schema snippet is as follows:

```

<xs:complexType name="RandomAccessType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Random Access
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="interval" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="type" type="RandomAccessTypeType" default="closed"/>
  <xs:attribute name="minBufferTime" type="xs:duration"/>
  <xs:attribute name="bandwidth" type="xs:unsignedInt"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="RandomAccessTypeType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Random Access Type type enumeration
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="closed"/>
    <xs:enumeration value="open"/>
    <xs:enumeration value="gradual"/>
  </xs:restriction>
</xs:simpleType>

```

5.3.5.6 Extended Bandwidth Signalling

Subclause 5.3.5.4 provides a detailed overview on the usage of @minBufferTime and @bandwidth for determining the bitrate properties of the associated Representation. However, this model does not expose certain extended bandwidth properties of the Representation, for example the nature of the encoding being variable bitrate, or any bitrate properties over different time windows.

For this purpose, each Representation have assigned an extended bandwidth signalling providing additional properties of the Representation. This information is provided as part of the ExtendedBandwidth element as defined in Table 12.

Table 12 — Semantics of Extended Bandwidth signalling

Element or Attribute Name	Use	Description
ExtendedBandwidth		
@vbr	OD default=false	<p>If set to true, the content is encoded to primarily have a constant or consistent quality, while at the same time when the signal restrictions on MPD@minBufferTime and Representation@bandwidth attributes are reached, the quality of the content may drop. In cases where is cap is not reached, the quality of the content is expected to be consistent.</p> <p>If set to false or not present, no information on the nature of the encoding is present.</p> <p>NOTE The signalling set to true implies that the actual instantaneous bitrate can frequently be below the maximum.</p>
ModelPair	0..N	defines an extended bandwidth model pair that applies for the Representation.
@bufferTime	M	<p>specifies a duration used in the definition of the Representation data rate together with @bandwidth attribute within the same element.</p> <p>NOTE There are cases for which a real or expected long-term average bitrate may be usefully signaled. Examples include (i) the signaling of the average bitrate of a Representation in On-Demand case. In this case the @buffer may be set to the duration of the Period and the @bandwidth may be set to the size of the Representation. (ii) the signaling of the maximum expected long-term bitrate in Live-Edge Periods: In this case the @buffer is set to a large value, for example 1 hour, or 24 hours and the bandwidth expresses the maximum amount of bytes over this period.</p>

Element or Attribute Name	Use	Description
@bandwidth	M	<p>Consider a hypothetical constant bitrate channel of bandwidth with the value of this attribute in bits per second (bps). Then, if the Representation is continuously delivered at this bitrate, starting at any signalled SAP of type 1 or 2 included in this Representation element, a client can be assured of having enough data for continuous playout providing playout begins after @bufferTime * @bandwidth bits have been received (i.e. at time @bufferTime after the first bit is received).</p> <p>For dependent Representations, this value specifies the bandwidth according to the above definition for the aggregation of this Representation and all complementary Representations.</p> <p>For details, see subclause 5.3.5.4.</p>
<p>Key</p> <p>For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory</p> <p>For elements: <minOccurs>..<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

The XML schema snippet is as follows:

```

<xs:complexType name="ExtendedBandwidthType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Extended Bandwidth Model
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ModelPair" type="ModelPairType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="vbr" type="xs:boolean" default="false"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="ModelPairType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Model Pair
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="bufferTime" type="xs:duration" use="required"/>
  <xs:attribute name="bandwidth" type="xs:unsignedInt" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.6 Sub-Representation

5.3.6.1 Overview

Sub-Representations are embedded in regular Representations and are described by the **SubRepresentation** element. **SubRepresentation** elements are contained in a **Representation** element.

The **SubRepresentation** element describes properties of one or several media content components that are embedded in the Representation. It may for example describe the exact properties of an embedded audio component (e.g. codec, sampling rate, etc.), an embedded sub-title (e.g. codec) or it may describe some embedded lower quality video layer (e.g. some lower frame rate, etc.).

Sub-Representations and Representation share some common attributes and elements.

In case the `@level` attribute is present in the **SubRepresentation** element,

- Sub-Representations provide the ability for accessing a lower quality version of the Representation in which they are contained. In this case, Sub-Representations for example allow extracting the audio track in a multiplexed Representation or may allow for efficient fast-forward or rewind operations if provided with lower frame rate;
- the Initialization Segment and/or the Media Segments and/or the Index Segments shall provide sufficient information such that the data can be easily accessed through HTTP partial GET requests. The details on providing such information shall be defined by the media format in use. For media formats defined in this document, the Subsegment Index as defined in subclause 6.3.2.4 shall be used.

If the `@level` attribute is absent, then the **SubRepresentation** element is solely used to provide a more detailed description for media streams that are embedded in the Representation.

The semantics of the attributes and elements within a Sub-Representation are provided in Table 13 of subclause 5.3.6.2. The XML syntax of the Sub-Representation type is provided in subclause 5.3.6.3.

5.3.6.2 Semantics

Table 13 — Semantics of SubRepresentation element

Element or Attribute Name	Use	Description
SubRepresentation		specifies a Sub-Representation.
<code>@level</code>	0	specifies the Sub-Representation level. If <code>@level</code> attribute is present and for media formats used in this document, a Subsegment Index as defined in subclause 6.3.2.4 shall be available for each Media Segment in the containing Representation. If not present, then the SubRepresentation element is solely used to provide a more detailed description for media streams that are embedded in the Representation.
<code>@dependencyLevel</code>	0	specifies the set of Sub-Representations within this Representation that this Sub-Representation depends on in the decoding and/or presentation process as a whitespace-separated list of <code>@level</code> values. If not present, the Sub-Representation can be decoded and presented independently of any other Representation.

Element or Attribute Name	Use	Description
@bandwidth	CM shall be present if @level is present	Identical to the @bandwidth definition in Representation but applied to this Sub-Representation. This attribute shall be present if the @level attribute is present.
@contentComponent	0	if present, specifies the set of all media content components that are contained in this Sub-Representation as a whitespace-separated list of values of ContentComponent @id values. if not present, the Sub-Representation is not assigned to a media content component.
CommonAttributesElements	-	Common Attributes and Elements (attributes and elements from base type RepresentationBaseType). For details, see subclause 5.3.7.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold preceded with an @, List of elements and attributes is in <i>italics</i> bold referring to those taken from the Base type that has been extended by this type.		

5.3.6.3 XML syntax

```

<xs:complexType name="SubRepresentationType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      SubRepresentation
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:attribute name="level" type="xs:unsignedInt"/>
      <xs:attribute name="dependencyLevel" type="UIntVectorType"/>
      <xs:attribute name="bandwidth" type="xs:unsignedInt"/>
      <xs:attribute name="contentComponent" type="StringVectorType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

5.3.7 Common attributes and elements

5.3.7.1 Overview

The elements **AdaptationSet**, **Representation** and **SubRepresentation** have assigned common attributes and elements. The attributes and elements listed in Table 14 of subclause 5.3.7.2 may be present in all three elements.

The semantics of the common attributes and elements are provided in Table 14 in subclause 5.3.7.2, the syntax is provided in subclause 5.3.7.3.

The 'Use' column in Table 14 shall be interpreted that an attribute marked with 'M' shall be available for a Representation, i.e. it shall either be present in the **Representation** element, or if not, it shall be in the containing **AdaptationSet** element. An attribute marked with 'O' may be absent in both.

5.3.7.2 Semantics

Table 14 — Common Adaptation Set, Representation and Sub-Representation attributes and elements

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
<i>Common attributes and elements</i>		
@profiles	0	<p>specifies the profiles which the associated Representation(s) conform to of the list of Media Presentation profiles as described in 8. The value shall be a subset of the respective value in any higher level of the document hierarchy (Representation, Adaptation Set, MPD).</p> <p>If not present, the value is inferred to be the same as in the next higher level of the document hierarchy. For example, if the value is not present for a Representation, then @profiles at the Adaptation Set level is valid for the Representation.</p> <p>The same syntax as defined in 5.3.1.2 shall be used.</p>
@width	0	<p>specifies the horizontal visual presentation size of the video media type on a grid determined by the @sar attribute.</p> <p>In the absence of @sar width and height are specified as if the value of @sar were "1 : 1"</p> <p>NOTE The visual presentation size of the video is equal to the number of horizontal and vertical samples used for presentation after encoded samples are cropped in response to encoded cropping parameters, "overscan" signalling, or "pan/scan" display parameters, e.g. SEI messages.</p> <p>If not present on any level, the value is unknown.</p>
@height	0	<p>specifies the vertical visual presentation size of the video media type, on a grid determined by the @sar attribute.</p> <p>If not present on any level, the value is unknown.</p>
@sar	0	<p>specifies the sample aspect ratio of the video media component type, in the form of a string consisting of two integers separated by ':', e.g. "10 : 11". The first number specifies the horizontal size of the encoded video pixels (samples) in arbitrary units. The second number specifies the vertical size of the encoded video pixels (samples) in the same units as the horizontal size.</p> <p>If not present on any level, the value is unknown.</p>

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
@frameRate	0	<p>specifies the output frame rate (or in the case of interlaced, half the output field rate) of the video media type in the Representation. If the frame or field rate is varying, the value is the average frame or half the average field rate over the entire duration of the Representation.</p> <p>The value is coded as a string, either containing two integers separated by a "/", ("F/D"), or a single integer "F". The frame rate is the division F/D, or F, respectively, per second (i.e. the default value of D is "1").</p> <p>If not present on any level, the value is unknown.</p>
@audioSamplingRate	0	<p>Either a single decimal integer value specifying the sampling rate or a whitespace-separated pair of decimal integer values specifying the minimum and maximum sampling rate of the audio media component type. The values are in samples per second.</p> <p>If not present on any level, the value is unknown.</p>
@mimeType	M	<p>specifies the MIME type of the concatenation of the Initialization Segment, if present, and all consecutive Media Segments in the Representation.</p>
@segmentProfiles	0	<p>specifies the profiles of Segments that are essential to process the Representation. The detailed semantics depend on the value of the @mimeType attribute.</p> <p>The contents of this attribute shall conform to either the pro-simple or pro-fancy productions of 4.5 of IETF RFC 6381:2011, without the enclosing DQUOTE characters, i.e. including only the unencodedv or encodedv elements respectively. As profile identifier, the brand identifier for the Segment as defined in 6 shall be used.</p> <p>4CC may contain characters that must be escaped in XML. 4CC strings shall be escaped according to W3C Extensible Markup Language (XML):2008, Section 2.4.</p> <p>If not present on any level, the value may be deducted from the value of the @profiles attribute.</p>

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
@codecs	0	<p>specifies the codecs present within the Representation. The codec parameters shall also include the profile and level information where applicable.</p> <p>For segment formats defined in this document, this element shall be present and the contents of this attribute shall conform to either the <code>simp-list</code> or <code>fancy-list</code> productions of IETF RFC 6381:2011, subclause 3.2, without the enclosing <code>DQUOTE</code> characters. The codec identifier for the Representation's media format, mapped into the name space for codecs as specified in IETF RFC 6381:2011, subclause 3.3, shall be used.</p>
@containerProfiles	0	<p>specifies the container profiles of Representations that are essential to process it. The detailed semantics depend on the value of the @mimeType attribute.</p> <p>The contents of this attribute shall conform to either the <code>pro-simple</code> or <code>pro-fancy</code> productions of IETF RFC 6381:2011, subclause 4.5, without the enclosing <code>DQUOTE</code> characters, i.e. including only the <code>unencodedv</code> or <code>encodedv</code> elements respectively.</p> <p>4CC may contain characters that must be escaped in XML. 4CC strings shall be escaped according to W3C Extensible Markup Language (XML):2008, Section 2.4.</p>
@maximumSAPPeriod	0	<p>when present, specifies the maximum SAP interval in seconds of all contained media streams, where the SAP interval is the maximum time interval between the T_{SAP} of any two successive SAPs of types 1 to 3 inclusive of one media stream in the associated Representations.</p> <p>If not present on any level, the value is unknown.</p>
@startWithSAP	0	<p>when present and greater than 0, specifies that in the associated Representations, each Media Segment starts with a SAP of type less than or equal to the value of this attribute value in each media stream.</p> <p>A Media Segment starts with a SAP in a media stream if the stream contains a SAP in that Media Segment, I_{SAU} is the index of the first access unit that follows I_{SAP} and I_{SAP} is contained in the Media Segment.</p> <p>If not present on any level, the value is unknown.</p>

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
@maxPlayoutRate	0	specifies the maximum playout rate as a multiple of the regular playout rate, which is supported with the same decoder profile and level requirements as the normal playout rate. If not present on any level, the value is 1.
@codingDependency	0	When present and 'true', for all contained media streams, specifies that there is at least one access unit that depends on one or more other access units for decoding. When present and 'false', for any contained media stream, there is no access unit that depends on any other access unit for decoding (e.g. for video all the pictures are intra coded). If not specified on any level, there may or may not be coding dependency between access units.
@scanType	0	specifies the scan type of the source material of the video media component type. The value may be equal to one of "progressive", "interlaced" and "unknown". If not specified on any level, the scan type is "progressive".
@selectionPriority	OD default=1	specifies the selection priority for the described data structures, i.e. the one described by the containing element. In the absence of other information, higher numbers are the preferred selection over lower numbers.
@tag	0	specifies the tag of the Representation, Adaptation Set or Preselection which may be used for selection purposes towards the decoder. NOTE This attribute is primarily introduced for the usage of Pre-Selections and Adaptation Sets, but future use for Representation and Sub-Representations is not precluded.
FramePacking	0 ... N	specifies frame-packing arrangement information of the video media component type. When no FramePacking element is provided for a video component, frame-packing shall not used for the video media component. For details, see subclauses 5.8.1 and 5.8.4.6.
AudioChannelConfiguration	0 ... N	specifies the audio channel configuration of the audio media component type. For details, see subclauses 5.8.1 and 5.8.4.7.
ContentProtection	0 ... N	specifies information about content protection schemes used for the associated Representations. For details, see subclauses 5.8.1 and 5.8.4.1.

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
OutputProtection	0 ... 1	specifies information about output protection schemes required for presenting the associated Representations For details, see subclauses 5.8.4.12.
EssentialProperty	0 ... N	specifies information about the containing element that is considered essential by the Media Presentation author for processing the containing element. For details, see subclause 5.8.4.8.
SupplementalProperty	0 ... N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause 5.8.4.9.
InbandEventStream	0 ... N	specifies the presence of an inband event stream in the associated Representations. For details, refer to subclause 5.10.
Switching	0 ... N	Specifies a switch-to times and types for the associated Representations. For more details, refer to subclause 0. These elements shall only be present if the @timescale value is the same for all Representations in one Adaptation Set and if the Segment Timeline is used for segment duration signalling.
RandomAccess	0 ... N	Specifies a random access times and types for the associated Representations. For more details, refer to subclause 5.3.5.5. These elements shall only be present if the @timescale value is the same for all Representations in one Adaptation Set and if the Segment Timeline is used for segment duration signalling.
GroupLabel	0 ... N	specifies a summary label for a group of Labels. For more details, refer to subclause 5.3.10.
Label	0 ... N	specifies a textual description of the element that may be used for annotation and selection purposes. For more details, refer to subclause 5.3.10.
ProducerReferenceTime	0 ... N	specifies the presence of and possibly values of producer reference time in the associated Representations. For details refer to subclause 5.12.

Element or Attribute Name	Use (see subclause 5.3.7.1)	Description
ContentPopularityRate	0 ... N	indicates a level of popularity of the containing entity (i.e., the Adaptation Set, Representation or Preselection) within the Media Presentation. For details, see subclause 5.14. NOTE This element is primarily introduced for the usage of Pre-Selections and Adaptation Sets but use for Representation and Sub-Representations is not precluded.
Resync	0 ... N	Specifies information on Segments' resynchronization points. For details refer to subclause 5.3.12.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.3.7.3 XML syntax

```

<xs:complexType name="RepresentationBaseType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Representation base (common attributes and elements)
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FramePacking" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="AudioChannelConfiguration" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="ContentProtection" type="ContentProtectionType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OutputProtection" type="DescriptorType" minOccurs="0"/>
    <xs:element name="EssentialProperty" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="SupplementalProperty" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="InbandEventStream" type="EventStreamType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Switching" type="SwitchingType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="RandomAccess" type="RandomAccessType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="GroupLabel" type="LabelType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Label" type="LabelType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ProducerReferenceTime" type="ProducerReferenceTimeType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ContentPopularityRate" type="ContentPopularityRateType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Resync" type="ResyncType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="profiles" type="ListOfProfilesType"/>
  <xs:attribute name="width" type="xs:unsignedInt"/>
  <xs:attribute name="height" type="xs:unsignedInt"/>
  <xs:attribute name="sar" type="RatioType"/>
  <xs:attribute name="frameRate" type="FrameRateType"/>

```

```

<xs:attribute name="audioSamplingRate" type="AudioSamplingRateType"/>
<xs:attribute name="mimeType" type="xs:string"/>
<xs:attribute name="segmentProfiles" type="ListOf4CCType"/>
<xs:attribute name="codecs" type="CodecsType"/>
<xs:attribute name="containerProfiles" type="ListOf4CCType"/>
<xs:attribute name="maximumSAPPeriod" type="xs:double"/>
<xs:attribute name="startWithSAP" type="SAPType"/>
<xs:attribute name="maxPlayoutRate" type="xs:double"/>
<xs:attribute name="codingDependency" type="xs:boolean"/>
<xs:attribute name="scanType" type="VideoScanType"/>
<xs:attribute name="selectionPriority" type="xs:unsignedInt" default="1"/>
<xs:attribute name="tag" type="TagType"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="AudioSamplingRateType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Audio Sampling Rate
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="UIntVectorType">
    <xs:minLength value="1"/>
    <xs:maxLength value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="VideoScanType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Video Scan type enumeration
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="progressive"/>
    <xs:enumeration value="interlaced"/>
    <xs:enumeration value="unknown"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CodecsType">
  <xs:restriction base="xs:string">
    <xs:pattern value="&charSet;&language;&id_list;">
      <xs:annotation>
        <xs:documentation>RFC6381 fancy-list without enclosing double
quotes</xs:documentation>
      </xs:annotation>
    </xs:pattern>
    <xs:pattern value="&id_simple;(,&id_simple;)*">
      <xs:annotation>
        <xs:documentation>RFC6381 simp-list without enclosing double
quotes</xs:documentation>
      </xs:annotation>
    </xs:pattern>
  </xs:restriction>
</xs:simpleType>

```

5.3.8 Subsets

5.3.8.1 Overview

Subsets are described by the **Subset** element contained in the **Period** element.

Subsets provide a mechanism to restrict the combination of active Adaptation Sets where an active Adaptation Set is one for which the DASH Client is presenting at least one of the contained Representations.

A Subset defines a set of one or more Adaptation Sets. The presence of a **Subset** element within a **Period** element expresses the intention of the creator of the Media Presentation that a client should act as follows: at any time, the set of active Adaptation Sets shall be a subset of the Adaptation Sets of one of the specified Subsets. Any Adaptation Set not explicitly contained in any Subset element is implicitly contained in all specified Subsets.

This implies that

- empty Subsets are not allowed.
- no Subset should contain all the Adaptation Sets.

Each Adaptation Set for which the value of the @id is provided in the @contains attribute is contained in this Subset.

The semantics of the attributes and elements within a Subset are provided in Table 15 of subclause 5.3.8.2. The XML syntax of the Subset type is provided in subclause 5.3.8.3.

5.3.8.2 Semantics

Table 15 — Subset element semantics

Element or Attribute Name	Use	Description
Subset		specifies a Subset
@contains	M	specifies the Adaptation Sets contained in a Subset by providing a white space separated list of the @id values of the contained Adaptation Sets.
@id	O	specifies a unique identifier for the Subset.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>.. <maxoccurs> (n="unbounded)<br/"> Elements are bold; attributes are non-bold and preceded with an @. </maxoccurs>>		

5.3.8.3 XML syntax

```

<xs:complexType name="SubsetType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Subset
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="contains" type="UIntVectorType" use="required"/>
  <xs:attribute name="id" type="xs:string"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.9 Segments and Segment information

5.3.9.1 General

This clause defines the MPD information for Segments. Segment formats are defined in 6.

Specifically, a Segment shall either be

- referenced by an HTTP-URL included in the MPD, where an HTTP-URL is defined as an <absolute-URI> according to IETF RFC 3986:2005 subclause 4.3, with a fixed scheme of “http” or “https”, possibly restricted by a byte range if a range attribute is provided together with the URL. The byte range shall be expressed as a byte-range-spec as defined in IETF RFC 2616 subclause 14.35.1. It is restricted to a single expression identifying a contiguous range of bytes, or

- included in the MPD with a data URL as defined in IETF RFC 2397. Data URLs shall only be used if explicitly permitted by the profile. In addition, the guidelines and recommendations in IETF RFC 2397 should be taken into account.

Each Segment referenced through an HTTP-URL in the MPD is associated with a Segment availability interval, i.e. a time window in wall-clock time at which the Segment can be accessed via the HTTP-URL. The Segment availability interval window is described by a Segment availability start time and a Segment availability end time.

Representations are assigned *Segment Information* through the presence of the elements **BaseURL**, **SegmentBase**, **SegmentTemplate** and/or **SegmentList**. The *Segment Information* provides information on the location, availability and properties of all Segments contained in one Representation. Specifically, information on the presence and location of Initialization, Media, Index and Bitstream Switching Segments is provided.

The elements **SegmentBase**, **SegmentTemplate** and **SegmentList** may be present in the **Representation** element itself. In addition, to express default values, they may be present in the **Period** and **AdaptationSet** element. At each level at most one of the three, **SegmentBase**, **SegmentTemplate** and **SegmentList**, shall be present. Further, if **SegmentTemplate** or **SegmentList** is present on one level of the hierarchy, then the other one shall not be present on any lower hierarchy level.

SegmentBase, **SegmentTemplate** and **SegmentList** shall inherit attributes and elements from the same element on a higher level. If the same attribute or element is present on both levels, the one on the lower level shall take precedence over the one on the higher level.

Several mechanisms are available to specify the *Segment Information*. Specifically, each Representation shall have assigned exactly one of the following choices to determine the *Segment Information*, either by direct presence in the **Representation** element or by inheritance from the higher levels:

- one **SegmentList** element — for syntax and semantics, refer to subclause 5.3.9.3,
- one **SegmentTemplate** element — for syntax and semantics, refer to subclause 5.3.9.4,
- one or more **BaseURL** elements, at most one **SegmentBase** element, and no **SegmentTemplate** or **SegmentList** element. The **SegmentBase** element is defined in subclause 5.3.9.2.

NOTE These rules do not prohibit the usage of the **BaseURL** element together with **SegmentList** or **SegmentTemplate**. If the **BaseURL** is present together with either the **SegmentList** or the **SegmentTemplate**, then processing according to subclause 5.6 applies. All three elements **SegmentBase**, **SegmentTemplate** and **SegmentList** share common elements based on the **SegmentBase** element. Furthermore, **SegmentTemplate** and **SegmentList** share common attributes and elements. The common information is defined in subclause 5.3.9.2.

The derivation and details of Initialization, Media, Index and Bitstream Switching Segment Information based on the above information are provided in subclause 5.3.9.5.

5.3.9.2 Segment base information

5.3.9.2.1 Overview

The **SegmentBase** element is sufficient to describe the *Segment Information* if and only if a single Media Segment is provided per Representation and the Media Segment URL is included in the **BaseURL** element.

In case multiple Media Segments are present, either a **SegmentList** or a **SegmentTemplate** shall be used to describe the *Segment Information*. **SegmentList** or a **SegmentTemplate** share the multiple Segment base information as provided in subclause 5.3.9.2.2, Table 17.

If the Representation contains more than one Media Segment, then either the attribute @duration or the element **SegmentTimeline** shall be present. The attribute @duration and the element **SegmentTimeline** shall not be present at the same time.

Segments described by the Segment base information are referenced by an HTTP-URL conforming to the type URLType as defined in Table 18.

The semantics of the attributes and elements for the **SegmentBase** element and the Segment base information are provided in subclause 5.3.9.2.2, Table 16 and the multiple Segment base information in Table 17 in subclause 5.3.9.2.2. The XML syntax of the Segment Base Information is provided in subclause 5.3.9.2.3.

5.3.9.2.2 Semantics

Table 16 — Semantics of SegmentBase element and Segment Base Information type

Element or Attribute Name	Use	Description
SegmentBase <i>Segment Base Information</i>		specifies Segment base element. This element also specifies the type for the Segment base information that is the base type for other elements.
@timescale	0	specifies the timescale in units per seconds to be used for the derivation of different real-time duration values in the Segment Information. If not present on any level, it shall be set to 1. NOTE This can be any frequency but typically is the media clock frequency of one of the media streams (or a positive integer multiple thereof).
@presentationTimeOffset	0	specifies the presentation time offset of the Representation relative to the start of the Period, i.e. the presentation time value of the media stream that shall be presented at the start of this Period. The value of the presentation time offset in seconds is the division of the value of this attribute and the value of the @timescale attribute. If not present on any level, the value of the presentation time offset is 0.

Element or Attribute Name	Use	Description
@eptDelta	0	<p>specifies the difference between the earliest presentation time in the Representation and the value of the @presentationTimeOffset.</p> <p>The value of the earliest presentation time of the first Media Segment in this Representation in seconds is computed as the sum of the value of this attribute and the value of the @presentationTimeOffset in units of the @timescale attribute.</p> <p>If the media is contained in a Self-Initializing Media Segment, the value of @presentationTimeOffset is used to identify the Sub-Segment, that is the first one of the Representation. In this case, the value of @eptDelta shall be set to the difference of the @presentationTimeOffset and the earliest presentation time of the Subsegment containing the media time provided by the value of @presentationTimeOffset.</p> <p>If not present, the value is unknown, but specific profiles may require it to be zero, if not signalled.</p> <p>NOTE If the value if @eptDelta is smaller than zero then this results in an overlap prior to the Period. If the value is positive, it results in a gap.</p>
@pdDelta	0	<p>specifies the difference between the presentation duration of this Representation in units of @timescale and the Period duration.</p> <p>If the media is contained in a Self-Initializing Media Segment, the sum of the value of @presentationTimeOffset and the Period duration is used to identify the Subsegment, that is the last one of the Representation. In this case, @pdDelta shall be set according to the presentation duration of this last Subsegment.</p> <p>If not present, the value is unknown.</p> <p>NOTE If the value if @pdDelta is smaller than zero then this results in a gap at the end of the Period. If the value is positive, it results in an overlap.</p>

Element or Attribute Name	Use	Description
@presentationDuration	0	<p>specifies the presentation duration of the Representation in the Period.</p> <p>The value of the presentation duration in seconds is the division of the value of this attribute and the value of the @timescale attribute. Specifically, the sum of the value of the @presentationTimeOffset, if present, or 0 otherwise and the value of this attribute is the last presentation time to be presented for this Representation.</p> <p>If not present on any level, the value of this attribute is unknown and the Representation should be presented until the end of the Period, i.e. until the presentation is terminated or until the next Period starts.</p>
@timeShiftBufferDepth	0	<p>specifies the duration of the time shifting buffer for this Representation that is guaranteed to be available for a Media Presentation with type 'dynamic'. When not present, the value is of the @timeShiftBufferDepth on MPD level applies. If present, this value shall be not smaller than the value on MPD level. This value of the attribute is undefined if the @type attribute is equal to 'static'.</p> <p>NOTE When operating in a time-shift buffer on a Representation with value larger than the time-shift buffer signalled on MPD level, not all Representations are necessarily available for switching.</p>
@indexRange	0	<p>specifies the byte range that contains the Segment Index in all Media Segments of the Representation.</p> <p>The byte range shall be expressed and formatted as a byte-range-spec as defined in IETF RFC 7233:2014, subclause 2.1. It is restricted to a single expression identifying a contiguous range of bytes.</p> <p>If not present, the value is unknown.</p>
@indexRangeExact	OD default "false"	<p>when set to 'true' specifies that for all Segments in the Representation, the data outside the prefix defined by @indexRange contains the data needed to access all access units of all media streams syntactically and semantically.</p> <p>This attribute shall not be present if @indexRange is absent.</p>

Element or Attribute Name	Use	Description
@availabilityTimeOffset	0	<p>specifies an offset to define the adjusted segment availability time. The value is specified in seconds, possibly with arbitrary precision.</p> <p>The offset provides the time how much earlier these segments are available compared to their computed availability start time for all Segments of all associated Representation.</p> <p>The segment availability start time defined by this value is referred to as adjusted segment availability start time. For details on computing the adjusted segment availability start time, refer to subclause 5.3.9.5.</p> <p>If not present, no adjusted segment availability start time is defined.</p> <p>NOTE The value of "INF" implies availability of all segments starts at MPD@availabilityStartTime.</p>
@availabilityTimeComplete	0	<p>specifies if all Segments of all associated Representation are complete at the adjusted availability start time. The attribute shall be ignored if @availabilityTimeOffset is not present on any level.</p> <p>If not present on any level, the value is inferred to true.</p> <p>NOTE If the value is set to false, then it can be inferred by the client that the Segment is available at its announced location prior being complete.</p>
Initialization	0 ... 1	<p>specifies the URL including a possible byte range for the Initialization Segment.</p> <p>For the type definition, refer to Table 18.</p>
RepresentationIndex	0 ... 1	<p>specifies the URL including a possible byte range for the Representation Index Segment.</p> <p>For the type definition, refer to Table 18.</p>
FailoverContent	0 ... 1	<p>specifies times where the content has been replaced by failover content, for example because of an encoder error.</p> <p>For details refer to subclause 5.3.9.7.</p>
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

Table 17 — Semantics of *MultipleSegmentBaseInformation* type

Element or Attribute Name	Use	Description
<i>MultipleSegmentBaseInformation</i>		specifies multiple Segment base information.
@duration	0	<p>If present, specifies the constant approximate Segment duration.</p> <p>All Segments within this Representation element have the same duration unless it is the last Segment within the Period, which can be significantly shorter.</p> <p>The value of the duration in seconds is the division of the value of this attribute and the value of the @timescale attribute associated to the containing Representation.</p> <p>For more details, refer to subclause 5.3.9.5.3.</p>
@startNumber	0	<p>specifies the number of the first Media Segment in this Representation in the Period.</p> <p>For details, refer to subclause 5.3.9.5.3.</p>
@endNumber	0	<p>specifies the number of the last Media Segment in this Representation in the Period.</p> <p>If not present, the number is inferred from the duration of the Period.</p> <p>For details refer to 5.3.9.5.3.</p>
<i>Segment Base Information</i>		specifies Segment base information.
SegmentTimeline	0...1	<p>specifies the timeline of arbitrary Segment durations</p> <p>For more details, see subclause 5.3.9.6.</p>
BitstreamSwitching	0 ... 1	<p>specifies the URL including a possible byte range for the Bitstream Switching Segment.</p> <p>For the type definition, refer to Table 18.</p>
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

Table 18 — Semantics of elements of type `URLType`

Element or Attribute Name	Use	Description
Element of type <code>URLType</code>		defines an HTTP-URL
<code>@sourceURL</code>	0	<p>specifies the source URL part and shall be formatted either as an <code><absolute-URI></code> according to IETF RFC 3986:2005, subclause 4.3, with a fixed scheme of “http” or “https” or as a <code><relative-ref></code> according to IETF RFC 3986:2005 subclause 4.2.</p> <p>If not present, then any BaseURL element is mapped to the <code>@sourceURL</code> attribute and the range attribute shall be present.</p>
<code>@range</code>	0	<p>specifies the byte range restricting the above HTTP-URL.</p> <p>The byte range shall be expressed and formatted as a <code>byte-range-spec</code> as defined in IETF RFC 7233:2014 subclause 2.1. It is restricted to a single expression identifying a contiguous range of bytes.</p> <p>If not present, the element refers to the entire resource referenced in the <code>@sourceURL</code> attribute.</p>

5.3.9.2.3 XML-Syntax

```

<xs:complexType name="SegmentBaseType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Segment information base
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Initialization" type="URLType" minOccurs="0"/>
    <xs:element name="RepresentationIndex" type="URLType" minOccurs="0"/>
    <xs:element name="FailoverContent" type="FailoverContentType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="timescale" type="xs:unsignedInt"/>
  <xs:attribute name="eptDelta" type="xs:integer"/>
  <xs:attribute name="pdDelta" type="xs:integer"/>
  <xs:attribute name="presentationTimeOffset" type="xs:unsignedLong"/>
  <xs:attribute name="presentationDuration" type="xs:unsignedLong"/>
  <xs:attribute name="timeShiftBufferDepth" type="xs:duration"/>
  <xs:attribute name="indexRange" type="SingleRFC7233RangeType"/>
  <xs:attribute name="indexRangeExact" type="xs:boolean" default="false"/>
  <xs:attribute name="availabilityTimeOffset" type="xs:double"/>
  <xs:attribute name="availabilityTimeComplete" type="xs:boolean"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="MultipleSegmentBaseType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Multiple Segment information base
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="SegmentBaseType">
      <xs:sequence>
        <xs:element name="SegmentTimeline" type="SegmentTimelineType"
minOccurs="0"/>
        <xs:element name="BitstreamSwitching" type="URLType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="duration" type="xs:unsignedInt"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="startNumber" type="xs:unsignedInt"/>
        <xs:attribute name="endNumber" type="xs:unsignedInt"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="URLType">
    <xs:annotation>
        <xs:documentation xml:lang="en">
            Segment Info item URL/range
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="sourceURL" type="xs:anyURI"/>
    <xs:attribute name="range" type="SingleRFC7233RangeType"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="SingleRFC7233RangeType">
    <xs:restriction base="xs:string">
        <xs:pattern value="([0-9]*) (\- ([0-9]*) )?" />
    </xs:restriction>
</xs:simpleType>

```

5.3.9.3 Segment list

5.3.9.3.1 Overview

The Segment list is defined by a **SegmentList** element. Each **SegmentList** element itself contains a list of **SegmentURL** elements for a consecutive list of Segment URLs. Each Segment URL may contain the Media Segment URL and possibly a byte range. The Segment URL element may also contain an Index Segment.

The semantics of the attributes and elements for the Segment list are provided in subclause 5.3.9.3.2, Table 19. The XML syntax of the Segment Information is provided in subclause 5.3.9.3.3.

5.3.9.3.2 Semantics

Table 19 — Semantics of **SegmentList** element

Element or Attribute Name	Use	Description
SegmentList		specifies Segment information.
@xlink:href	0	specifies a reference to a remote element entity that contains one or multiple elements of type SegmentList .
@xlink:actuate	0D default: "onRequest"	specifies the processing set, can be either "onLoad" or "onRequest"
MultipleSegmentBaseInformation		Multiple Segment base information as defined in subclause 5.3.9.2, Table 17.
SegmentURL	0 ... N	specifies a Media Segment URL and a possibly present Index Segment URL

Element or Attribute Name	Use	Description
@media	0	<p>in combination with the @mediaRange attribute, specifies the HTTP-URL for the Media Segment.</p> <p>It shall be formatted as an <absolute-URI> according to IETF RFC 3986:2005, subclause 4.3, with a fixed scheme of “http” or “https” or as a <relative-ref> according to IETF RFC 3986:2005, subclause 4.2.</p> <p>If not present, then any BaseURL element is mapped to the @media attribute and the range attribute shall be present.</p>
@mediaRange	0	<p>specifies the byte range within the resource identified by the @media corresponding to the Media Segment.</p> <p>The byte range shall be expressed and formatted as a byte-range-spec as defined in IETF RFC 7233:2014, subclause 2.1. It is restricted to a single expression identifying a contiguous range of bytes.</p> <p>If not present, the Media Segment is the entire resource referenced by the @media attribute.</p>
@index	0	<p>in combination with the @indexRange attribute, specifies the HTTP-URL for the Index Segment.</p> <p>It shall be formatted as an <absolute-URI> according to IETF RFC 3986:2005, subclause 4.3, with a fixed scheme of “http” or “https” or as a <relative-ref> according to IETF RFC 3986:2005, subclause 4.2.</p> <p>If not present and the @indexRange not present either, then no Index Segment information is provided for this Media Segment.</p> <p>If not present and the @indexRange present, then the @media attribute is mapped to the @index. If the @media attribute is not present either, then any BaseURL element is mapped to the @index attribute and the @indexRange attribute shall be present.</p>

Element or Attribute Name	Use	Description
@indexRange	0	<p>specifies the byte range within the resource identified by the @index corresponding to the Index Segment. If @index is not present, it specifies the byte range of the Segment Index in Media Segment.</p> <p>The byte range shall be expressed and formatted as a byte-range-spec as defined in IETF RFC 7233:2014, subclause 2.1. It is restricted to a single expression identifying a contiguous range of bytes.</p> <p>If not present, the Index Segment is the entire resource referenced by the @index attribute.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>The conditions only hold without using @xlink:href. If linking is used, then all attributes are "optional" and <minOccurs=0>.</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.3.9.3.3 XML-Syntax

```

<xs:complexType name="SegmentListType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Segment List
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="MultipleSegmentBaseType">
      <xs:sequence>
        <xs:element name="SegmentURL" type="SegmentURLType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="xlink:href"/>
      <xs:attribute ref="xlink:actuate" default="onRequest"/>
      <xs:attribute ref="xlink:type" fixed="simple"/>
      <xs:attribute ref="xlink:show" fixed="embed"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SegmentURLType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Segment URL
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="media" type="xs:anyURI"/>
  <xs:attribute name="mediaRange" type="SingleRFC7233RangeType"/>
  <xs:attribute name="index" type="xs:anyURI"/>
  <xs:attribute name="indexRange" type="SingleRFC7233RangeType"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.9.4 Segment template

5.3.9.4.1 Overview

The Segment template is defined by the **SegmentTemplate** element. In this case, specific identifiers that are substituted by dynamic values assigned to Segments, to create a list of Segments. The substitution rules are provided in subclause 5.3.9.4.4.

The semantics of the attributes and elements for the Segment list are provided in subclause 5.3.9.4.2, Table 20. The XML syntax of the Segment Information is provided in subclause 5.3.9.4.3.

5.3.9.4.2 Semantics

Table 20 — Semantics of **SegmentTemplate** element

Element or Attribute Name	Use	Description
SegmentTemplate		specifies Segment template information.
<i>MultipleSegmentBaseInformation</i>		Provides the Multiple Segment base information as defined in subclause 5.3.9.2.
@media	0	specifies the template to create the Media Segment List. For more details, refer to subclause 5.3.9.4.4.
@index	0	specifies the template to create the Index Segment List. If neither the \$Number\$ nor the \$Time\$ identifier is included, this provides the URL to a Representation Index. For more details, refer to subclause 5.3.9.4.4.
@initialization	0	specifies the template to create the Initialization Segment. Neither \$Number\$ nor the \$Time\$ identifier shall be included. For more details, refer to subclause 5.3.9.4.4.
@bitstreamSwitching	0	specifies the template to create the Bitstream Switching Segment. Neither \$Number\$ nor the \$Time\$ identifier shall be included. For more details, refer to subclause 5.3.9.4.4.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.3.9.4.3 XML syntax

```

<xs:complexType name="SegmentTemplateType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Segment Template
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="MultipleSegmentBaseType">
      <xs:attribute name="media" type="xs:string"/>
      <xs:attribute name="index" type="xs:string"/>
      <xs:attribute name="initialization" type="xs:string"/>
      <xs:attribute name="bitstreamSwitching" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:extension>
</xs:complexContent>
</xs:complexType>

```

5.3.9.4.4 Template-based Segment URL construction

The **SegmentTemplate@media** attribute, the **SegmentTemplate@index** attribute, the **SegmentTemplate@initialization** attribute and the **SegmentTemplate@bitstreamSwitching** attribute each contain a string that may contain one or more of the identifiers as listed in Table 21.

In each URL, the identifiers from Table 20 shall be replaced by the substitution parameter defined in Table 16. Identifier matching is case-sensitive. If the URL contains unescaped \$ symbols which do not enclose a valid identifier, then the result of URL formation is undefined. In this case, it is expected that the DASH Client ignores the entire containing **Representation** element and the processing of the MPD continues as if this **Representation** element was not present. The format of the identifier is also specified in Table 21.

Each identifier may be suffixed, within the enclosing '\$' characters, with an additional format tag aligned with the `printf` format tag as defined in IEEE 1003.1-2008^[10] following this prototype:

```
%0[width]d
```

The *width* parameter is an unsigned integer that provides the minimum number of characters to be printed. If the value to be printed is shorter than this number, the result shall be padded with zeros. The value is not truncated even if the result is larger.

The Media Presentation shall be authored such that the application of the substitution process results in valid Segment URLs.

Strings outside identifiers shall only contain characters that are permitted within URLs according to IETF RFC 3986.

Table 21 — Identifiers for URL templates

\$<Identifier>\$	Substitution parameter	Format
\$\$	Is an escape sequence, i.e. "\$\$" is replaced with a single "\$"	not applicable
<i>\$RepresentationID\$</i>	This identifier is substituted with the value of the attribute Representation@id of the containing Representation .	The format tag shall not be present.
<i>\$Number\$</i>	This identifier is substituted with the <i>number</i> of the corresponding Segment, if <i>\$SubNumber\$</i> is not present in the same string. If <i>\$SubNumber\$</i> is present, this identifier is substituted with the <i>number</i> of the corresponding Segment sequence. For details, refer to subclauses 5.3.9.6.4 and 5.3.9.6.5.	The format tag may be present. If no format tag is present, a default format tag with <i>width=1</i> shall be used.
<i>\$Bandwidth\$</i>	This identifier is substituted with the value of Representation@bandwidth attribute value.	The format tag may be present. If no format tag is present, a default format tag with <i>width=1</i> shall be used.

\$<Identifier>\$	Substitution parameter	Format
\$Time\$	This identifier is substituted with the value of the MPD start time of the Segment being accessed. For the Segment Timeline, this means that this identifier is substituted with the value of the SegmentTimeline@t attribute for the Segment being accessed. Either \$Number\$ or \$Time\$ may be used but not both at the same time.	The format tag may be present. If no format tag is present, a default format tag with width=1 shall be used.
\$SubNumber\$	This identifier is substituted with the <i>number</i> of the corresponding Segment in a Segment Sequence. This identifier shall only be present if either \$Number\$ or \$Time\$ are present as well. For details, refer to subclauses 5.3.9.6.4 and 5.3.9.6.5.	The format tag may be present. If no format tag is present, a default format tag with width=1 shall be used.

5.3.9.5 Segment information

5.3.9.5.1 Overview

The *Segment Information* provides the following information:

- the presence or absence of Initialization, Index and Bitstream Switching Segment information,
- the HTTP-URL and possibly a byte range for each accessible Segment in each Representation, or the Segment directly in the MPD as a data URL as defined in IETF RFC 2397,
- all valid Segment URLs declared by the containing MPD,
- for services with **MPD@type='dynamic'**, the Segment availability start time and Segment availability end time of each Segment,
- an approximate Media Presentation start time of each Media Segment in the Media Presentation timeline within the Period.

The derivation of Initialization, Media, Index and Bitstream Switching Segment Information from the elements **SegmentBase**, **SegmentList** and **SegmentTemplate** is provided in subclauses 5.3.9.5.2, 5.3.9.5.3, 5.3.9.5.4 and 5.3.9.5.5. Reference resolution as defined in subclause 5.6.4 and base URL selection as defined in subclause 5.6.5 using **BaseURL** elements as defined in subclause 5.6 shall be applied to any URLs.

5.3.9.5.2 Initialization Segment information

Each Representation has assigned at most one Initialization Segment.

The presence of an Initialization Segment is indicated by the presence of **SegmentBase.Initialization**, **SegmentList.Initialization**, the **SegmentTemplate.Initialization** element or the **SegmentTemplate@initialization** attribute that may contain URL and byte range information or URL construction rules for the Initialization Segment.

If neither **Initialization** element nor **SegmentTemplate@initialization** attribute are present for a Representation then each Media Segment within the Representation shall be self-initializing.

For services with **MPD@type='dynamic'**, the Segment availability start time of the Initialization Segment is the sum of the value of the **MPD@availabilityStartTime** and the *PeriodStart* time as defined in subclause 5.3.2.1 of the containing Period and the Segment availability end time of the

Initialization Segment is the largest Segment availability end time of any Media Segment in this Representation. For Segment availability of Media Segments, refer to subclause 5.3.9.5.3.

The data structures retrieved from the Initialization URL are defined in subclause 6.2.2.

5.3.9.5.3 Media Segment information

If a Representation consists of more than one Media Segment, then this Representation has assigned a list of consecutive Media Segments. The list may be specified explicitly by one or more **SegmentList** elements or implicitly by a **SegmentTemplate** element.

Each entry in this Media Segment list has assigned the following parameters:

- a valid Media Segment URL and possibly a byte range,
- the number and position of the Media Segment in the Representation,
- the MPD start time of the Media Segment in the Representation providing an approximate presentation start time of the Segment in the Period,
- the MPD duration of the Media Segment providing an approximate presentation duration of the Segment.

The MPD start time and the MPD duration may be approximate and do not necessarily reflect the exact Media Presentation time. For more details on the relation of MPD start times and Media Presentation time, refer to subclause 7.2.1.

In order to obtain the list of Media Segment URLs, i.e. the URL for each Segment at a specific position k in the list based on the Segment Information, the following shall apply:

- If **SegmentTemplate** element is present, the Template-based Segment URL construction in subclause 5.3.9.4.4 shall be applied as follows.
 - If the Representation contains or inherits a **SegmentTemplate** element with $\$Number\$$ then the URL of the Media Segment at position k in the Representation is determined by replacing the $\$Number\$$ identifier by $(k-1) + (k_{start}-1)$ with k_{start} the value of the `@startNumber` attribute, if present, or 1 otherwise.
 - If the Representation contains or inherits a **SegmentTemplate** element with $\$Time\$$ then the URL of the media segment at position k is determined by replacing the $\$Time\$$ identifier by MPD start time of this segment, as described below.
- If one or more **SegmentList** elements are present then they contain a list of **SegmentURL** elements for a consecutive list of Media Segment URLs. The number of the first Segment in the list within this Period is determined by the value of the `SegmentList@startNumber` attribute, if present, or it is 1 in case this attribute is not present. The sequence of multiple **SegmentList** elements within a Representation shall result in Media Segment List with consecutive numbers.
- None of the above: in this case, only a single Media Segment shall be present with the URL provided by a **BaseURL** element and the **SegmentBase** element may be present.

For the derivation of the MPD start time and duration of each Media Segment in the list of Media Segments, the position k of the Media Segment and the following information are used.

- If neither `@duration` attribute nor **SegmentTimeline** element is present, then the Representation shall contain exactly one Media Segment. The MPD start time is 0 and the MPD

duration is obtained in the same way as for the last Media Segment in the Representation (see below for more details).

- If `@duration` attribute is present, then the MPD start time of the Media Segment is determined as $(Number - Number_{Start})$ times the value of the attribute `@duration` with $Number_{Start}$ the value of the `@startNumber` attribute, and $Number$ the segment number [e.g. $(k-1) + Number_{Start}$]. The MPD duration of the Media Segment is the value of the attribute `@duration` unless the Media Segment is the last one the Representation (see below for more details).
- If `@duration` attribute is not present and the **SegmentTimeline** element is present then rules in subclause 5.3.9.6 apply to determine the start time and duration of each Media Segment in the Media Segment list.
- To determine the duration of the only or the last Media Segment of any Representation in a Period, the MPD shall include sufficient information to determine the duration of the containing Period. For example, the **MPD@mediaPresentationDuration**, or **Period@duration**, or next **Period@start** may be present. The Media Presentation author should always provide a Segment as indicated by the duration of the Period, even if the Segment may not contain any media samples. If the content author provides a different number of Segments as nominally derived, it should add the `@endNumber` attribute to provide the number of the last segment in the Period. The value of the `@endNumber` should differ by at most 1 from the nominally derived value. A Segment with a number greater than the value of `@endNumber` is not valid.

For services with **MPD@type** = 'dynamic', the Segment availability start time of a Media Segment is the sum of

- the value of the **MPD@availabilityStartTime**,
- the *PeriodStart* time of the containing Period as defined in subclause 5.3.2.1,
- the MPD start time of the Media Segment, and
- the MPD duration of the Media Segment.

NOTE By adding the MPD duration of the segment to the segment availability start time of the segment, the segment availability start time of the first segment of each Period depends on the segment duration. This enables to provide segments in Representations with shorter MPD duration earlier, for example to reduce latency for certain Representations.

The Segment availability end time of a Media Segment is the sum of the Segment availability start time, the MPD duration of the Media Segment and the value of the attribute `@timeShiftBufferDepth` for this Representation.

If the `@availabilityTimeOffset` attribute is present for a Representation in the Segment Information or the **BaseURL** element, then the parameter `availabilityTimeOffset` is determined as the sum of all values of `@availabilityTimeOffset` on all levels that are processed in determining the URL for the corresponding segment. Then the adjusted segment availability start time is determined by subtracting the value of `availabilityTimeOffset` from the Segment availability start time. This adjusted segment availability start time provides a time instant in wall-clock time at which a Segment becomes an available Segment. If the `@availabilityTimeComplete` flag is set to false for such a Representation on any level, then the entire Segment may not yet be fully available at the adjusted segment availability start time. No promise how the Segment will be made available is provided in this case. To signal the gradual availability of Segments over time, refer to subclause 5.3.12.1.

The MPD shall include URL information for all Segments with an availability start time less than both (i) the end of the Media Presentation and (ii) the sum of the latest time at which this version of the MPD is available on the server and the value of the **MPD@minimumUpdatePeriod**.

The data structures retrieved from the URL referring to a Media Segment are defined in subclause 6.2.3.

5.3.9.5.4 Index Segment information

Each Segment typically has assigned Segment Index information that may be provided in an explicitly declared Index Segment.

The presence of explicit Index Segment information is indicated

- by the presence of one **RepresentationIndex** element providing the Segment Index for the entire Representation, or
- by the presence of at least one of the two attributes **@index** and **@indexRange** in the **SegmentList.SegmentURL** element, or
- by the presence of **SegmentTemplate@index** attribute. If either **\$Number\$** or **\$Time\$** are present the Template-based Segment URL construction in subclause 5.3.9.4.4 shall be applied with number set to the number of the corresponding Media Segment. If not present, the **SegmentTemplate@index** attribute constitutes a reference to Representation Index.

The **@indexRange** attribute may also be used to provide the byte range for an index within a Media Segment, where this is allowed by the Media Segment format. In this case the **@index** attribute shall not be present and the range specified shall lie completely within any byte range specified for the Media Segment.

The availability of Index Segments is identical to the availability to the Media Segments they correspond to.

The data structures retrieved from the URL referring to an Index Segment are defined in subclause 6.2.4.

5.3.9.5.5 Bitstream Switching Segment information

Each Representation has assigned at most one Bitstream Switching Segment. The Bitstream Switching Segment is only relevant in case the **@bitstreamSwitching** flag is set to 'true' and may enable the creation of a conforming Segment track for Segments from different Representations.

The presence of a Bitstream Switching Segment is indicated by the presence of the **BitstreamSwitching** element or the **SegmentTemplate@bitstreamSwitching** attribute that may contain URL and byte range information or construction rules for the URL.

If neither **BitstreamSwitching** element nor **SegmentTemplate@bitstreamSwitching** attribute are present for a Representation and the **@bitstreamSwitching** flag is set to 'true', there are no Bitstream Switching Segments.

The Segment availability time of the Bitstream Switching Segment is identical to the one specified for the Initialization Segment in subclause 5.3.9.5.2.

The data structures retrieved from the URL referring to a Bitstream Switching Segment are defined in subclause 6.2.5.

5.3.9.6 Segment timeline

5.3.9.6.1 General

The **SegmentTimeline** element expresses the earliest presentation time and presentation duration (in units based on the @timescale attribute) for each Segment in the Representation. The use is an alternative to providing the @duration attribute and provides four additional features:

- the specification of arbitrary Segment durations,
- the specification of accurate Segment durations for one media stream where the duration expresses presentation duration of the Segment, and
- the signalling of discontinuities of the Media Presentation timeline for which no Segment data are present in a specific Representation.
- the ability to signal Segment sequences. For more details, refer to subclause 5.3.9.6.4. Segment sequences shall only be used if explicitly permitted by the profile in use.

For compactness, the syntax of this element includes run-length compression to express a sequence of Segments having constant duration.

The **SegmentTimeline** element shall contain a list of **S** elements each of which describes a sequence of contiguous segments of identical MPD duration. The **S** element contains a mandatory @d attribute specifying the MPD duration, an optional @r repeat count attribute specifying the number of contiguous Segments with identical MPD duration minus one and an optional @t time attribute. The value of the @t attribute minus the value of the @presentationTimeOffset specifies the MPD start time of the first Segment in the series.

The @r attribute has a default value of zero (i.e., a single Segment in the series) when not present. For example, a repeat count of three means there are four contiguous Segments, each with the same MPD duration. The value of the @r attribute of the **S** element may be set to a negative value indicating that the duration indicated in @d repeats until the **S**@t of the next **S** element or if it is the last **S** element in the **SegmentTimeline** element until the end of the Period or the next update of the MPD, i.e. it is treated in the same way as the @duration attribute for a full period.

Any @d value shall not exceed the value of **MPD**@maxSegmentDuration.

The @d attribute shall represent the accurate presentation duration of the Segment except for the following conditions are met:

- the @availabilityTimeOffset is present for the Representation,
- the @availabilityTimeComplete is set to FALSE for the Representation,
- the **S** element is the one with the largest **S**@t value in the MPD, and
- the non-adjusted segment availability time of the Segment is greater than or equal to the publish time of the MPD.

This means that for such a Segment, the @d attribute may be updated with a new version of the MPD, or a new **S** element may be added to reflect a different duration of the Segment than the nominal one. A DASH Client should not rely on nominal segment durations for the derivation of Segment addresses in the case the \$Time\$ identifier is present, but instead the duration of the Segment may be derived by parsing the Segment and determining the value or by updating the MPD after the non-adjusted segment availability time.

The textual order of the **S** elements within the **SegmentTimeline** element shall match the numbering (and thus time) order of the corresponding Media Segments.

When the **SegmentTemplate** is in use and the `$Time$` identifier is present in the **SegmentTemplate@media** then

- If a Segment Index ('`sidx`') box is present, then the values of the **SegmentTimeline** shall describe accurate timing of each Media Segment. Specifically, these values shall reflect the information provided in the Segment index ('`sidx`') box, i.e.
 - the value of `@timescale` shall be identical to the value of the `timescale` field in the first '`sidx`' box,
 - the value of `S@t` shall be identical to the value of the `earliest_presentation_time` in the first '`sidx`' box of the Media Segment described in **S**,
 - the value of `S@d` shall be identical to sum of the values of all `Subsegment_duration` fields in the first '`sidx`' box of the Media Segment described in **S**.
- If a Segment Index ('`sidx`') box is not present, then the derivation of the earliest presentation time shall be based on the media internal data. The details depend on the segment format in use and further restriction on the segment format may apply.
- The Segment URL for a Media Segment is obtained by replacing the `$Time$` identifier by the earliest presentation time obtained from the **SegmentTimeline**.

NOTE As the earliest presentation time of the next Media Segment in the same Representation can be derived from the actual Media Segment, e.g. by the use of the Segment Index, the Segment URL can be generated without reading of the updated MPD that contains the update to the Segment Timeline.

The semantics of the attributes and elements for Segment Timeline are provided in subclause 5.3.9.6.2, Table 22. The XML syntax of the Segment Timeline is provided in subclause 5.3.9.6.3.

5.3.9.6.2 Semantics

Table 22 — Semantics of **SegmentTimeline** element

Element or Attribute Name	Use	Description
SegmentTimeline		specifies the Segment timeline information
S	0 .. N	specifies Segment start time and duration for a contiguous sequence of segments of identical durations, referred to as series in the following. NOTE The S elements are ordered in sequence of increasing values of the attribute <code>@t</code> .

Element or Attribute Name	Use	Description
@t	0	<p>this value of this attribute minus the value of the @presentationTimeOffset specifies the MPD start time, in @timescale units, of the first Segment in the series. The MPD start time is relative to the beginning of the Period.</p> <p>The value of this attribute shall be equal to or greater than the sum of the previous S element earliest presentation time and the sum of the contiguous Segment durations.</p> <p>If the value of the attribute is greater than what is expressed by the previous S element, it expresses discontinuities in the timeline.</p> <p>If not present, then the value shall be assumed to be zero for the first S element and for the subsequent S elements, the value shall be assumed to be the sum of the previous S element's earliest presentation time and contiguous duration [i.e. previous S@t + @d * (@r + 1)].</p>
@n	0	<p>specifies the Segment number of the first Segment in the series.</p> <p>The value of this attribute shall be at least one greater than the number of previous S elements plus the @startNumber attribute value, if present. If the value of @n is greater than one plus the previously calculated Segment number, it expresses that one or more prior Segments in the timeline are unavailable.</p>
@d	M	specifies the Segment duration or the duration of a Segment sequence, in units of the value of the @timescale.
@k	OD default: 1	<p>specifies the number of Segments that are included in a Segment Sequence. The attribute shall not be present unless explicitly permitted by the profile.</p> <p>For more details, refer to subclause 5.3.9.6.3.</p>
@r	OD default: 0	specifies the repeat count of the number of following contiguous Segments or Segment Sequences with the same duration expressed by the value of @d. This value is zero-based (e.g. a value of three means four Segments or Segment Sequences in the contiguous series). A negative value of the @r attribute of the S element indicates that the duration indicated in @d attribute repeats until the start of the next S element, the end of the Period or until the next MPD update.

Element or Attribute Name	Use	Description
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.3.9.6.3 XML syntax

```

<xs:complexType name="SegmentTimelineType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Segment Timeline
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="S" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="t" type="xs:unsignedLong"/>
        <xs:attribute name="n" type="xs:unsignedLong"/>
        <xs:attribute name="d" type="xs:unsignedLong" use="required"/>
        <xs:attribute name="r" type="xs:integer" default="0"/>
        <xs:attribute name="k" type="xs:unsignedLong" default="1"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.9.6.4 Segment Sequences

Segment sequences in the Segment Timeline may be signalled with including a @k attribute in the **S** element of the **SegmentTimeline**. The @k shall only be present if all of the following requirements are fulfilled:

- the addressing scheme for the associated Representation is using Segment template with either \$Number\$ or hierarchical templating and sub-numbering as defined in subclause 5.3.9.6.5,
- the profile explicitly allows the usage of Segment sequences.

If @k is present and greater than 1, then it specifies that sequence described by @d is accurate in timing but contains @k Segments.

The MPD duration of the Segment is determined as the integer value of @d divided by the value of @k and determines the MPD start time and therefore the Segment availability start time. MPD duration of the Segments is not required to exactly match the media duration of the Segments.

The integer of the quotient of the value of @d and the value of @k of any **S** element shall not exceed the quotient of @d and the value of @k minus 1 of any other **S** element in the Segment Timeline.

The concatenation of all Segments in a Segment sequence shall have an accurate segment duration according of the value of @d.

Any Segment in a Segment sequence may carry inband events.

5.3.9.6.5 Hierarchical Templating and Sub-Numbering

If the Segment template contains a \$SubNumber\$ value and a Segment Timeline signalling with Segment sequence is used, then

- if `$Time$` is present, the `$Time$` is replaced with the earliest presentation time of the Segment sequence for all Segments in the Segment Sequence,
- if `$Number$` is present, the `$Number$` is replaced with the number of the Segment sequence, i.e. with the number as if every Segment sequence in the Segment timeline is treated as single Segment,
- and in both cases the `$SubNumber$` is replaced with the Segment number of the Segment Sequence, with 1 being the number of the first Segment in the sequence.

NOTE The earliest presentation time of the next Segment Sequence in the same Representation can be derived from the sum of the earliest presentation time of the current Segment Sequence and the duration of the Segment resulting from the concatenation of all Media Segments in a Segment Sequence. In case of ISO BMFF, this can be accomplished by summing the track runs of segments in the segment sequence.

5.3.9.7 Failover Content Signalling

5.3.9.7.1 General

In certain cases, a time-continuous section of a Representation or an Adaptation Set is not properly represented, for example due to errors in the content ingest or encoding. However, at least at the DASH level, the content may be offered properly by adding a Missing Segments as defined in 6.2.6 or by including content that is encoded but representing some failover content instead of the main content.

The **FailoverContent** element enables identification of such sections of Representations that include alternative “failover content” and should be avoided by the DASH Client, for example by switching to a Representation that does not include such failover content.

The **FailoverContent** element contains a list of failover content sections, each expressed by one **FCS** element. The **FCS** element contains an optional `@d` attribute specifying the MPD duration of the alternative content section and a mandatory `@t` time attribute whereby the value of the `@t` attribute minus the value of the `@presentationTimeOffset` specifies the MPD start time of the alternative content section.

The semantics of the attributes and elements for failover content are provided in 5.3.9.7.2, Table 23. The XML syntax of the Failover Content signalling is provided in 5.3.9.7.3.

5.3.9.7.2 Semantics

Table 23 — Semantics of **FailoverContent** element

Element or Attribute Name	Use	Description
FailoverContent		specifies sections that contain failover content.
<code>@valid</code>	OD default: <code>true</code>	specifies whether the corresponding data results in a valid HTTP request or if the request is invalid and it is expected that a 404 is received when requesting the associated Media Segment.
FCS	1 .. N	specifies one continuous section of failover content.
<code>@t</code>	M	this value of this attribute minus the value of the <code>@presentationTimeOffset</code> specifies the MPD start time, in <code>@timescale</code> units, of the first sample in the alternative content section. The value of this attribute shall be equal to or greater than the sum of the previous FCS element

		earliest presentation time and the sum of the contiguous section durations.
@d	0	<p>specifies the alternative content section duration in units of the value of the @timescale.</p> <p>If not present, the alternative content section lasts until the start of the next FCS element, or until the end of the Period or until the end of MPD duration, whichever occurring first in the timeline.</p>
<p>Legend:</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory.</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.3.9.7.3 XML syntax

```

<xs:complexType name="FailoverContentType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Failover Content
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FCS" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="t" type="xs:unsignedLong" use="required"/>
        <xs:attribute name="d" type="xs:unsignedLong"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="valid" type="xs:boolean" default="true"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.10 Label and Group Label

5.3.10.1 Overview

Labels provide the ability to annotate data structures in a DASH Media Presentation by providing a short textual description of the parent element. Labels can be used for the purpose of interaction and configuration settings within a DASH Client or associated application. For example, the **Label** element can be used to provide a selection menu to the user.

GroupLabel may be added on a higher level to provide a summary or title of labels collected in a group. For example, **GroupLabel** can be used as the title on a selection menu containing a collection of labels.

Labels with the same value for the @id attribute belong to a label group. All labels with the same @id shall be on the same level in the MPD.

The @lang attribute assigned to a label describes the language of that label. For example, the @lang attribute can be used by the DASH Client to filter labels according to user language preferences.

The semantics of the label element and the group label element are provided in Table 24 and Table 25, respectively, in subclause 0, the XML syntax is provided in subclause 5.3.10.3.

5.3.10.2 Semantics

Table 24 — Semantics of Label element

Element or Attribute Name	Use	Description
Label		The label and the actual text of the label that annotates the element in the DASH Media Presentation
@id	OD default: 0	An identifier for the label. Labels with the same value for this attribute belong to a label group.
@lang	0	Specifies the language of the label. If not present, the language of the label is unknown.

Table 25 — Semantics of GroupLabel element

Element or Attribute Name	Use	Description
GroupLabel		The actual text of the group label that annotates the element in the DASH Media Presentation
@id	OD default: 0	An identifier for the group label. The value for this attribute shall be unique for all group labels in the MPD. The value of this attribute shall be the same as the id attribute of all Label elements in the label group.
@lang	0	Specifies the language of the label. If not present, the language of the label is unknown.

5.3.10.3 XML Syntax

```

<xs:complexType name="LabelType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Label and Group Label
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="id" type="xs:unsignedInt" default="0"/>
      <xs:attribute name="lang" type="xs:language"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="ProducerReferenceTimeType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Producer Reference time
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="UTCTiming" type="DescriptorType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="inband" type="xs:boolean" default="false"/>
  <xs:attribute name="type" type="ProducerReferenceTimeTypeType" default="encoder"/>
  <xs:attribute name="applicationScheme" type="xs:string"/>
  <xs:attribute name="wallClockTime" type="xs:string" use="required"/>
  <xs:attribute name="presentationTime" type="xs:unsignedLong" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.3.11 Preselection

5.3.11.1 Overview

The concept of Preselections was initially considered for the purpose of enabling Next Generation Audio (NGA) codecs to signal suitable combinations of audio elements that are offered in different Adaptation Sets. However, the Preselection concept is introduced in a generic manner such that it can also be applicable to and used by other media types and codecs.

Preselections define user experiences that can be selected by the DASH Client. Each Preselection is uniquely identifiable and distinguishable, e.g. by language. A Preselection encompasses a subset of media components such that the media components can be selected and combined into a complete experience.

Preselections can be used to reference a set of Representations from multiple Adaptation Sets in order to produce a complete experience. Preselections can also be used to indicate a pre-defined experience at the elementary-stream level, i.e. the DASH Client can select a pre-defined experience and provides the selection to the media engine.

Preselections may be uniquely identified by a Preselection Tag. Users/Codecs using this Tag functionality are encouraged to provide more information on how tags defined in the MPD map to functionality in the specific codec.

Preselections have equivalent annotation parameters to Adaptation Sets and are always assigned exactly one media type.

Media components can be mapped to Adaptation Sets in multiple ways:

1. by a one-to-one mapping between media components and Adaptation Sets;
2. by the inclusion of multiple media components in a single Adaptation Set where all encoded versions of the media components are multiplexed on the file-container level;
3. by the inclusion of multiple media components in a single Adaptation Set where all encoded versions of the media components are multiplexed on the elementary-stream level.

If the Adaptation Set contains a single media component, then the media component can be referenced by the `@id` of the Adaptation Set.

If the Adaptation Set contains multiple media components multiplexed on the file-container level, then each media component is mapped to a Content Component as defined in 5.3.4. For example, in the ISO BMFF case, a Representation contains multiple tracks and each track is mapped to a Content Component. Therefore, media components can be referenced by the `@id` of an Adaptation Set or the `@id` of a Content Component. When Preselections reference Content Components, the `@id` of Adaptation Sets and Content Components shall be unique within the scope of a Period.

If the Adaptation Set contains multiple media components multiplexed at the elementary-stream level, then a pre-defined experience is referenced by the Preselection. For example, in the ISO BMFF case, a Representation contains a single track of multiple media components that is referenced by the `@id` of the Adaptation Set. Multiple Preselections can reference the Adaptation Set and select a pre-defined experience by passing the Presentation Tag to the media engine along with the media stream.

The Main Adaptation Set is the Adaptation Set that contains the Initialisation Segment for the complete experience. Each Preselection shall reference a Main Adaptation Set and may reference zero, one or more other Adaptation Sets.

NOTE In the context of Preselection the term "Main Adaptation Set" is used. This term is not to be confused with an Adaptation Set that has assigned the main Role.

Within a Preselection, two types of Adaptation Sets are differentiated:

- Main Adaptation Set: A Representation of this Adaptation Set is needed for playback of the Preselection. In particular for ISO BMFF, the Initialization Segment of such a Representation is needed for playback of the Preselection.
- Partial Adaptation Set: A Representation of this Adaptation Set is only consumable together with the Main Adaptation Sets within this Preselection. Again, in particular for ISO BMFF, the Initialization Segment of a Representation of the Main Adaptation Set is needed for playback.

Preselections, main Adaptation Set and partial Adaptation Sets may be defined by one of the two means:

- A preselection descriptor as specified in subclause 5.3.11.2. Such a descriptor enables simple configurations and preserves backward compatibility but may not be suitable for advanced use cases.
- A preselection element as specified in 5.3.11.3 and 5.3.11.4. The semantics of the Preselection element is provided in subclause 5.3.11.3 and the XML syntax is provided in 5.3.11.4.

5.3.11.2 Preselection Descriptor

The Preselection Descriptor is used for two purposes.

- To indicate that an Adaptation Set is part of a Preselection.
- Optionally, to provide instructions on how to combine the Adaptation Set with other Adaptation Sets to form a Preselection.

The Preselection Descriptor shall be either an Essential Property Descriptor or a Supplemental Property Descriptor with a `@schemeIdURI` of "urn:mpeg:dash:preselection:2016".

For simple use cases, the Preselection Descriptor may be used to provide instructions on how to combine the Adaptation Set with other Adaptation Sets to form a Preselection. The descriptor shall only be present at the Adaptation Set level.

The `@value` attribute of the descriptor provides two fields, separated by a comma:

- the Preselection Tag
- the id of the Adaptation Sets or Content Components referenced by this Preselection as a white space separated list in processing order. The first id references the Main Adaptation Set.

The syntax for the value attribute of the Preselection descriptor shall follow the `PRESELECTION-DESCRIPTOR-VALUE` as defined in the following ABNF notation according to IETF RFC 5234:

```
PRESELECTION-DESCRIPTOR-VALUE = TAG-VALUE "," ID-LIST

ID-LIST = ID-VALUE [ WHITESPACE ID-VALUE ]
TAG-VALUE = STRING
ID-VALUE = 1* DIGIT
STRING = *VCHAR
```

If the descriptor is present, but the value field is absent, the Adaptation Set shall be referenced by at least one Preselection.

If the value field is present, then this descriptor identifies a Preselection. The Tag is assigned, and the Preselection includes the ids of the Adaptation Sets that are part of this Preselection. In this case, the Multi-Segment Track Conformance rule as defined in subclause 5.3.11.5.2 applies.

If the descriptor is present and used with the Essential Descriptor, then this indicates that the Adaptation Set is only consumable as part of a Preselection.

If the descriptor is present and used with the Supplemental Descriptor, then this indicates that the Adaptation Set is also consumable independently of a Preselection.

5.3.11.3 Semantics of Preselection element

As an alternative to the Preselection descriptor, Preselections may also be defined through the Preselection element as provided in Table 26. The selection of Preselections is based on the contained attributes and elements in the Preselection element.

Table 26 — Semantics of PreSelection element

Element or Attribute Name	Use	Description
Preselection		
@id	OD default=1	specifies the id of the Preselection. This shall be unique within one Period.
@preselectionComponents	M	specifies the ids of the contained Adaptation Sets or Content Components that belong to this Preselection as white space separated list in processing order. The first id defines the Main Adaptation Set.
@lang	O	same semantics as in Table 5 for @lang attribute.
@order	OD Default: 'undefined'	specifies the conformance rules for Representations in Adaptation Sets within the Preselection. When set to 'undefined', the Preselection follows the conformance rules for Multi-Segment Tracks in subclause 5.3.11.5.1. When set to 'time-ordered', the Preselection follows the conformance rules for Time-Ordered Segment Tracks in subclause 5.3.11.5.2. When set to 'fully-ordered', the Preselection follows the conformance rules for Fully-Ordered Segment Tracks in subclause 5.3.11.5.3. In this case, order in the @preselectionComponents attribute specifies the component order.
Accessibility	0 ... N	specifies information about accessibility scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.3.
Role	0 ... N	specifies information on role annotation scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.2.
Rating	0 ... N	specifies information on rating scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.4.

Element or Attribute Name	Use	Description
Viewpoint	0 ... N	specifies information on viewpoint annotation scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.5.
<i>CommonAttributesElements</i>	-	specifies the common attributes and elements (attributes and elements from base type RepresentationBaseType). For details, see subclause 5.3.7.
Key For attributes: M=mandatory, O=Optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>..<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.3.11.4 XML Syntax

```

<xs:complexType name="PreselectionType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Preselection
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="Accessibility" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Role" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Rating" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Viewpoint" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="id" type="StringNoWhitespaceType" default="1"/>
      <xs:attribute name="preselectionComponents" type="StringVectorType"
use="required"/>
      <xs:attribute name="lang" type="xs:language"/>
      <xs:attribute name="order" type="PreselectionOrderType" default="undefined"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="PreselectionOrderType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Preselection Order type
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="undefined"/>
    <xs:enumeration value="time-ordered"/>
    <xs:enumeration value="fully-ordered"/>
  </xs:restriction>
</xs:simpleType>

```

5.3.11.5 Conformance rules

5.3.11.5.1 Conformance rules for Multi-Segment Tracks

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the regular conformance rules for multi-segment tracks as defined in subclause 5.3.5.1.

No additional conformance rules are defined for the Representations in different Adaptation Sets within Preselections.

5.3.11.5.2 Conformance rules for Time-Ordered Segment Track

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the conformance rules for multi-segment tracks as defined in subclause 5.3.11.5.1.

In addition, the concatenation of the following shall represent a conforming Segment track as defined in subclause 4.5.4 and conforming to the media type as specified in the `@mimeType` attribute for the Representation of the Main Adaptation Set:

- An Initialization Segment of one Representation of the Main Adaptation Set (specified by the first id in the `@preselectionComponents` attribute or the Preselection Descriptor, and
- media segments/subsegments of one Representation from each Adaptation Set referenced in the Preselection ordered by non-decreasing first decode times.

NOTE This does not constrain the order of segments with the same first decode time.

If Adaptation Sets within a Preselection are time-ordered as defined above, the Representations of all Adaptation Sets referenced by the Preselection should be segment/subsegment aligned as defined in subclause 5.3.3.5.

5.3.11.5.3 Conformance rules for Fully-Ordered Segment Track

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the conformance rules for multi-segment tracks as defined in subclause 5.3.11.5.1.

In addition, the concatenation of the following shall represent a conforming Segment track as defined in subclause 4.5.4 and conforming to the media type as specified in the `@mimeType` attribute for the Representation of the Main Adaptation Set:

- An Initialization Segment of one Representation of the Main Adaptation Set (specified by the first id in the `@preselectionComponents` attribute or the Preselection Descriptor), and
- media segments/subsegments of one Representation from each Adaptation Set referenced in the Preselection ordered first by non-decreasing decode times and then by position in the list given in `@preselectionComponents`.

If Adaptation Sets referenced by a Preselection are fully ordered as defined above, the Representations of all Adaptation Sets referenced by the Preselection shall be segment/subsegment aligned as defined in subclause 5.3.3.5.

5.3.12 Initialization Set, Group and Presentation

5.3.12.1 Initialization Set

An Initialization Set provides a common set of media properties across Periods in a Media Presentation. If an Initialization Set with certain properties is provided in an MPD, there shall be at least one Adaptation Set in at least one Period with these properties. More importantly, it can also be indicated that *all* Periods in the Media Presentations include at least one Adaptation Set that follows the properties of the Initialization Set. Especially in the latter case, an Initialization Set can be selected at the start of a Media Presentation in order to establish the relevant decryption, decoding and rendering environment. This provides expected consistent playback behaviour for the DASH Client if it selects only Adaptation Sets that are included in the selected Initialization Set.

The semantical definition of Initialization Sets shares all parameters of Adaptation Sets, but an Initialization Set may only include a subset of those parameters. An Adaptation Set referring to an Initialization Set that it conforms to may include additional information, for example:

- There may be an Initialization Set with media type audio and a `@codecs` parameter, but without a `@lang` attribute. Then each Period includes at least one Adaptation Set with the same media type, the same `@codecs` parameter, but each of the Adaptation Sets defines a different language for each Period. This ensures that initialization of an audio playback decoder is possible, but over Period boundaries, the language may change, or additional languages may be added.
- There may be an Initialization Set with media type video and `@maxWidth` and `@maxHeight` parameter and a `@codecs` parameter. Then at least one Period includes at least one Adaptation Set with the same media type, but its actual codecs parameters and the `@maxWidth` and `@maxHeight` may be different than the ones in the Initialization Set, but can be decoded and displayed within the Initialization Set constraints.

An Initialization Set may get assigned a specific profile using the `@profiles` parameter in order to express conformance to DASH profiles or Interoperability Points.

If an MPD has multiple Periods, there should be at least one Initialization Set be present for each media type.

The semantics of the attributes and elements within an **InitializationSet** element are provided in subclause 5.3.12.2, Table 27. The XML syntax of the **InitializationSet** element is provided in subclause 5.3.12.3.

5.3.12.2 Semantics

Table 27 — Semantics of InitializationSet element

Element or Attribute Name	Use	Description
InitializationSet		Initialization Set description
<code>@xlink:href</code>	0	specifies a reference to a remote element entity that shall contain exactly one element of type InitializationSet
<code>@xlink:actuate</code>	OD default: 'onRequest'	specifies the processing instructions, which can be either "onLoad" or "onRequest".
<code>@id</code>	M	specifies a unique identifier for this Initialization Set. The attribute shall be a unique unsigned integer value amongst all InitializationSet and InitializationGroup ids in the scope of the MPD.
<code>@inAllPeriods</code>	OD Default: TRUE	if set to true, then there is at least one Adaptation Set in each Period of the Media Presentation that conforms to this Initialization Set.
<code>@contentType</code>	0	specifies the media content component type for this Initialization Set. A value of the top-level Content-type 'type' value as defined in 4 of IETF RFC 6838:2013 shall be taken. If not present, the media content component type may be defined for each media component or it may be unknown.

Element or Attribute Name	Use	Description
@par	0	<p>specifies the picture aspect ratio of the video media component type, in the form of a string consisting of two integers separated by ':', e.g. "16:9". When this attribute is present, and the attributes @width and @height for the Representations of Adaptation Sets conforming to this Initialization Set are also present, the picture aspect ratio as specified by this attribute shall be the same as indicated by the values of @width, @height, and @sar, i.e. it shall express the same ratio as $(@width * sar_x) : (@height * sar_y)$, with sar_x the first number in @sar and sar_y the second number.</p> <p>If not present, the picture aspect ratio may be defined for each media component or it may be unknown.</p>
@maxWidth	0	<p>specifies the maximum @width value in all Representations in all Adaptation Sets associated to this Initialization Set. This value has the same units as the @width attribute.</p> <p>If not present, the value is unknown.</p>
@maxHeight	0	<p>specifies the maximum @height value in all Representations in all Adaptation Sets associated to this Initialization Set. This value has the same units as the @height attribute.</p> <p>If not present, the value is unknown.</p>
@maxFrameRate	0	<p>specifies the maximum @frameRate value in all Representations in all Adaptation Sets associated to this Initialization Set. This value is encoded in the same format as the @frameRate attribute.</p> <p>If not present, the value is unknown.</p>
@initialization	0	specifies the URL of an Initialization Segment.
CommonAttributesElements	-	specifies the common attributes and elements (attributes and elements from base type RepresentationBaseType). For details see 5.3.7.
Accessibility	0 ... N	<p>specifies information about accessibility scheme for all Adaptation Sets associated to this Initialization Set.</p> <p>For more details, refer to subclauses 5.8.1 and 5.8.4.3.</p>
Role	0 ... N	<p>specifies information on role annotation scheme for all Adaptation Sets associated to this Initialization Set.</p> <p>For more details, refer to subclauses 5.8.1 and 5.8.4.2.</p>
Rating	0 ... N	<p>specifies information on rating scheme for all Adaptation Sets associated to this Initialization Set</p> <p>For more details, refer to subclauses 5.8.1 and 5.8.4.4.</p>

Element or Attribute Name	Use	Description
Viewpoint	0 ... N	specifies information on viewpoint annotation scheme for all Adaptation Sets associated to this Initialization Set For more details, refer to subclauses 5.8.1 and 5.8.4.5.
Legend: For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using <code>xlink:href</code> . If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @. List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

5.3.12.3 XML syntax

```

<xs:complexType name="InitializationSetType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Initialization Set
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="Accessibility" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Role" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Rating" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Viewpoint" type="DescriptorType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="xlink:href"/>
      <xs:attribute ref="xlink:actuate" default="onRequest"/>
      <xs:attribute ref="xlink:type" fixed="simple"/>
      <xs:attribute name="id" type="xs:unsignedInt" use="required"/>
      <xs:attribute name="inAllPeriods" type="xs:boolean" default="true"/>
      <xs:attribute name="contentType" type="RFC6838ContentTypeType"/>
      <xs:attribute name="par" type="RatioType"/>
      <xs:attribute name="maxWidth" type="xs:unsignedInt"/>
      <xs:attribute name="maxHeight" type="xs:unsignedInt"/>
      <xs:attribute name="maxFrameRate" type="FrameRateType"/>
      <xs:attribute name="initialization" type="xs:anyURI"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

5.3.12.4 Initialization Group and Initialization Presentation

5.3.12.4.1 Overview

Initialization Group and Initialization Presentation signal the capabilities required for playback of the content as intended by the service provider.

The **InitializationGroup** element lists the minimum requirements for playback of a media content type in all Periods of the Media Presentation. Any Period contained in the Media Presentation shall have at least one Adaptation Set that conforms to one of the Initialization Sets included in this Initialization Group. Therefore, a client supporting all listed Initialization Sets in one **InitializationGroup** element is expected to be able to play that media type during the entire Media Presentation.

The **InitializationPresentation** element lists the minimum requirements for playback of a Media Presentation in all Periods. A client supporting all listed Initialization Sets and Groups of one

Initialization Presentation is expected to be able to play the entire Media Presentation as intended by the service provider.

An Initialization Group or Presentation may get assigned a specific profile using the `@profiles` parameter in order to express conformance to DASH Profiles or Interoperability Points.

5.3.12.4.2 Application of Initialization hierarchy

Figure 4 demonstrates the use of initialization hierarchy in signalling the required capabilities. In this example, the MPD has 3 different media types: video, audio and subtitle. The MPD has several Periods, the Adaptation Sets of which are described by different Initialization Sets for each media type: Initialization Sets (IS) 1-4 define the requirements for 4 different video decoders, Initialization Sets 5-7 define the requirements for 3 audio decoders, and Initialization Sets 8-9 define the requirements for 2 different subtitle parsers.

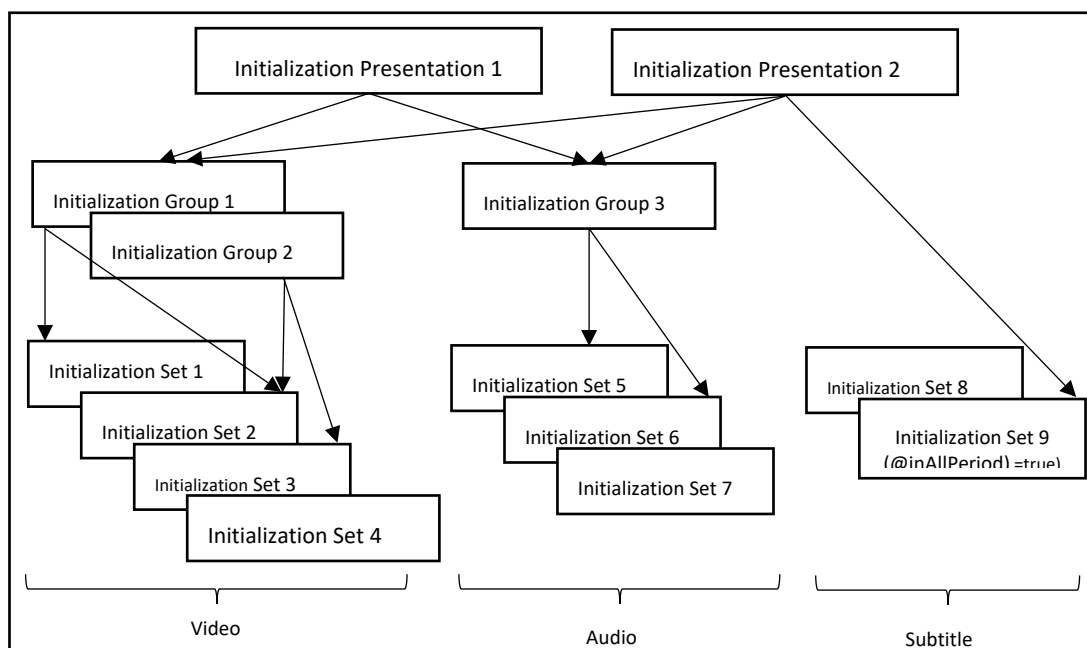


Figure 4 — An Example of initialization signaling.

In Figure 4, each Initialization Group defines the minimum requirement for playback of a media type for entire Media Presentation, while each Initialization Presentation defines the requirements to playback Media Presentation as intended by the service provider. Initialization Group (IG) 1 consists of IS1 and IS2, while Initialization Group 2 consists of IS2 and IS3, which means that a client can play the video during all periods if it supports the decoders defined in IG1 or IG2. Similarly, IG3 consists of IS5 and IS6, so a client supporting decoders defined by IS5 and IS6 can play the audio during all periods. And finally, since IS9 `@inAllPeriod` is set to true, a client supporting IS9 can play the existing subtitle in all periods.

To playback the content as intended by service provider, the MPD contains of two Initialization Presentations (IP) 1 and 2. IP1 defines a complete presentation that consists of audio and video (i.e. the client has to support IG1 and IG3). IP2 adds the support for playback of subtitle IS9 to its presentation requirements.

5.3.13 Resynchronization

5.3.13.1 Overview

In DASH context, in typical cases Segments are treated as a single unit for download, random access to Media Presentations, and they are also addressed by a single URL. However, Segments may have internal

structures that enable resynchronization on container level and random access to the respective Representation even within a Segment. The resynchronization mechanism is supported and signalled by the **Resync** element.

The **Resync** element signals Resynchronization Points in Segments. A Resynchronization Point marks the start (in byte position) of a well-structured continuous byte range within a Segment that contains media data of certain presentation duration and can be accessed independently on container format level. A Resynchronization Point may provide additional functionalities, such as access on decryption and decoding level.

A container format making use of the Resynchronization feature must define a Resynchronization Point and associated properties.

A Resynchronization Point in a Segment is defined as follows:

1. A Resynchronization Point enables to start parsing and processing on the container level.
2. A Resync Point has assigned the following properties:
 - It has a byte offset or index *Index* from the start of the Segment, pointing to the Resynchronization Point.
 - It has an earliest presentation time *Time* in the Representation, i.e. the smallest presentation time of any sample included in the Representation when starting to process from the Resynchronization Pointer.
 - It has assigned a type *Type*, for example, defined by the SAP type in ISO/IEC 14496-12.
 - It has assigned a boolean marker property *Marker* indicating whether the Resynchronization Point can be detected while parsing Segment through a specific structure or if the resync point needs to be signalled by external means.
3. Starting to process the Segment from a Resynchronization Point, together with the information in the Initialization Segment, if present, allows container parsing. Whether and how to access the contained and potentially encrypted elementary stream may be indicated by the resynchronization access point type.

Signalling each Resynchronization Point with all properties in the MPD can be done by providing a separate metadata Segment describing the Resynchronization Points in a Segment. However, not in all cases such side-car Segments may be provided, or at least provided in time. For example, in case of dynamic and live services, Resynchronization Points are added by the Segment packager independent of MPD updates. A Resynchronization Point may be generated by the encoder and packager independently from the MPD. Also, in low-latency cases, the MPD signalling may not be available to the DASH client.

Hence, there are two non-mutually exclusive ways specified to signal Resync Points provided in a Segment in an MPD:

1. By providing a binary map for each Resynchronization Point in a Representation that includes timed metadata track. This is most easily used for Segments that are fully available on the network.
2. By signalling the existence of Resynchronization Points in a Media Segment with additional information that permits to easily locate the Resynchronization Points in terms of the byte position and the presentation time, as well as providing the type of the Resynchronization Point.

In case the **Resync** element is present with *@dImin* and *@dT* attributes included and the derived from these values are adjusted values *dImin* in bytes and *dT* in seconds, respectively, and the *@availabilityTimeComplete* attribute is set to false, then the following shall hold:

- At the adjusted availability start time of the Segment, the first chunk is available.

- At the sum of the adjusted availability start time of the Segment and $i \cdot dT$, the $(i+1)$ th chunk is available with $i=1, \dots, N$ and N the total amount of chunks in the Segment.
- If the @rangeAccess on **BaseURL** attribute associated to the Representation is set to true, available chunks may be accessed with byte ranges. If set to false, the client should not expect a response corresponding to the requested byte range.

In order to signal the above properties, a **Resync** element is defined with different attributes, that are explained in more detail in subclause 5.3.13.2, Table 28. The XML Syntax is provided in subclause 5.3.13.3.

5.3.13.2 Semantics

Table 28 — Semantics of Resynchronization Point

Element or Attribute Name	Use	Description
Resync		Resync Point signalling if present, specifies that Resync Points are present in the Segments of the corresponding Representation.
@type	OD default: "0"	specifies the type of the Resync Point. if 0, specifies that on Segment format level the Resync Points allow parsing and decryption to access the Segments, in combination with the Initialization Segment of the corresponding Representation (if present). For all Segment formats defined in this document, the following holds: if greater than 0, specifies that one or multiple Resync Points are present with the properties of @type=0 and the properties defined in the element in each corresponding Segment with SAP type being equal to or smaller than the one indicated in the value. For details on SAP types, refer to subclause 4.5.2.
@dT	0	specifies the maximum difference of the Time values of any two consecutive Resynchronization Points that are included in the @type definition above in any Media Segment for this Representation. The value is expressed in scale of the @timescale of the corresponding Representation. If not present, the value is unknown.
@dImax	0	specifies the maximum difference of the Index values of any two consecutive Resync Points that are included in the @type definition above in any Media Segment for this Representation normalized by the @bandwidth value of the Representation. To obtain the maximum difference in octets, the value of @dImax is multiplied with the @bandwidth value. If not present, the value is unknown.

Element or Attribute Name	Use	Description
@dImin	OD Default: 0	<p>specifies the minimum difference of the <code>Index</code> values of any two consecutive Resynchronization Points that are included in the <code>@type</code> definition of this element in any Media Segment for this Representation normalized by the <code>@bandwidth</code> value of the Representation.</p> <p>To obtain the minimum difference in octets, the value of <code>@dImin</code> is multiplied with the <code>@bandwidth</code> value.</p> <p>If not present, the value is assumed to be 0.</p>
@marker	OD Default: FALSE	<p>If set to <code>true</code>, it specifies that every Resynchronization Point includes a specific marker for Resynchronization Point detection.</p> <p>If not present or <code>FALSE</code>, the presence of a marker for every signalled Resynchronization Point cannot be expected.</p> <p>Every Segment format making use of this functionality must specify a Resynchronization Marker. For details refer to the Segment formats in Clause 6.</p>

5.3.13.3 XML Syntax

```

<xs:complexType name="ResyncType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Resynchronization Point
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="type" type="SAPType" default="0"/>
  <xs:attribute name="dT" type="xs:unsignedInt"/>
  <xs:attribute name="dImax" type="xs:float"/>
  <xs:attribute name="dImin" type="xs:float" default="0"/>
  <xs:attribute name="marker" type="xs:boolean" default="false"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.4 Media Presentation Description updates

5.4.1 General

If **MPD**`@type` is set to 'dynamic', the MPD may be updated during the Media Presentation. Updates typically extend the accessible Segment list for each Representation, introduce a new Period, update Segment locations or terminate the Media Presentation.

When the MPD is updated

- the updated MPD shall be made available as follows:
 - At the same location as the original MPD, if no `MPD.Location` element is present in the original MPD at the same location provided by the `MPD.Location` elements are present in the original MPD
 - at all locations provided by the `MPD.Location` elements;
 - the updated MPD may be made available as an MPD patch document by providing one or several `MPD.PatchLocation` elements that can be combined and processed with the

original MPD to generate an updated MPD. The MPD patch document is updated at the same time when the MPD is updated. Hence, timing related to MPD patch document is expected to be identical the operation of MPD updates.

- the value of **MPD@id**, if present, shall be the same in the original and the updated MPD;
- the values of any **Period@id** attributes shall be the same in the original and the updated MPD, unless the containing **Period** element has been removed;
- the values of any **AdaptationSet@id** attributes shall be the same in the original and the updated MPD unless the containing **Period** element has been removed;
- any Representation with the same **@id** and within the same Period as a Representation appearing in the previous MPD shall provide functionally equivalent attributes and elements and shall provide functionally identical Segments with the same indices in the corresponding Representation in the new MPD.

If the attribute **MPD@minimumUpdatePeriod** is not present, no update to the MPD is expected, the attribute **MPD@mediaPresentationDuration** or the **Period@duration** of the last Period shall be present and the MPD shall remain valid until the Media Presentation end time.

If the attribute **MPD@minimumUpdatePeriod** is present, updates to the MPD are expected and restricted in a sense that at the location where the MPD is available at a certain time, the MPD is also valid for the duration of the value of the **MPD@minimumUpdatePeriod** attribute. Specifically, the following shall hold.

If the i -th version of the MPD is the last version of MPD till the end of the Media Presentation, let $Texp(i)$ be the Media Presentation end time.

Otherwise, let $Texp(i)$ be the sum of the value of **MPD@minimumUpdatePeriod** and the wall-clock time at which the i -th version of the MPD is updated [and replaced with the $(i+1)$ -th version] at the location where the MPD is available. The i -th MPD shall remain valid until $Texp(i)$ in the following sense:

- all Segments with availability start time less than $Texp(i)$ shall be available at their availability start times at the location advertised in the i -th MPD.
- all Representations have a Segment with an availability start time, $Tavail$, which is less than $Texp(i)$ and with duration not less than $[Texp(i) - Tavail]$.

NOTE 1 The actual duration of this Segment is not known at the client until this Segment or the updated MPD is fetched and this Segment duration can be less than the previous Segment duration if it is the last Segment in the Period or if the Segment Timeline is in place.

NOTE 2 It is possible that the clients do not know $Texp(i)$, but they can each calculate a lower bound on $Texp(i)$ by adding **MPD@minimumUpdatePeriod** to the wall-clock time at which they request the MPD, or the MPD patch document.

NOTE 3 The second condition above ensures that sufficient media is contained in each Representation to present the Media Presentation up to $Texp(i)$ for a client that begins playing each Segment at the earliest possible time (its availability start time).

NOTE 4 The result of the MPD validity requirement is that all items a client expects to be able to retrieve (both segments and MPD elements) are guaranteed to be available for retrieval during the periods that the client can expect them to be accessible.

NOTE 5 An MPD can contain no Period element or only an early available Period can be provided. In this case, updates to the MPD are expected in order to provide the start time of the first Period, which coincides with the start of the actual Media Presentation.

NOTE 6 An update of the MPD does not necessarily change the MPD but can only extend the validity of this MPD.

NOTE 7 If the `@minimumUpdatePeriod` is set to 0, then as a consequence of NOTE 2, all segments with availability start time less than the request time of the MPD are available at the location advertised in the MPD.

5.4.2 MPD Reset

In certain cases, the DASH server may fail to produce an MPD update and may have to reset the entire content on the server, but still offer the MPD at the same URL as part of an MPD update. Whereas such an update may be inconsistent with the rules in subclause 5.4.1, it is expected that the client will continue the service properly when such a reset occurs.

If the MPD author does such a reset, it shall signal the reset by providing either an Essential Descriptor or a Supplemental Descriptor on MPD level with `@schemeIdURI` set to `urn:mpeg:dash:reset:2016`. If this is signalled, the content author indicates that the MPD that results from this update does not need to comply with MPD update rules in subclause 5.4.1. If an Essential Descriptor is used, the DASH Client may terminate presentation, if the signal is not supported by the client.

DASH Clients receiving such an MPD as part of an MPD update are expected to continue using the previous MPD as long as possible, but at most up to the time expressed by the sum of the availability start time and the Period start time of the new MPD. At this time, the new MPD is expected to be used for playout. Should the client not have available any media data until the updated MPD can be used for playout, it is expected that the DASH Client conceals the gap in media time appropriately.

The updated MPD may be valid upon reception of the MPD including the reset signal, if the sum of the availability start time and the Period start time of the new MPD is less than the current MPD presentation time. In this case, a DASH Client may cancel decoding and presentation of media data from the previous MPD overlapping the updated one.

5.5 MPD assembly

5.5.1 General

A mechanism for referencing a *remote element entity* from within a local MPD is defined. A subset of W3C XLINK simple links is defined consisting of

- restricted syntax and semantics as specified in subclause 5.5.2, and
- the processing model as specified in subclause 5.5.3.

If the MPD is updated, then the rules in subclause 5.2.3.5 apply to the MPD after dereferencing all remote element entities.

5.5.2 Syntax and semantics

Table 29 provides the XLINK attributes that are used in this document and shall be supported accordingly.

Table 29 — XLINK attributes used in this document

Attribute	Comments and Usage
<code>@xlink:type</code>	Identifies the type of W3C XLINK being used. In the context of this document, all references shall be W3C XLINK simple links. The attribute <code>@xlink:type</code> is optional with fixed setting <code>@xlink:type="simple"</code> .

Attribute	Comments and Usage
@xlink:href	<p>references the remote element entity by a URI as defined in IETF RFC 3986.</p> <p>In the context of this document, such URIs shall exclusively be HTTP-URLs.</p> <p>For example, cookies as defined in IETF RFC 6265 may be used in order to enable targeted resolution of the same HTTP-URL provided in this attribute. Detailed requirements on the support and usage of such technologies are outside the scope of this document.</p>
@xlink:show	<p>Defines the desired behaviour of the remote element entity once dereferenced from within an MPD as defined in W3C XLINK.</p> <p>In the context of this document, the attribute <code>@xlink:show</code> is optional with fixed setting <code>@xlink:show="embed"</code>.</p> <p>NOTE In W3C XLINK, the behaviour of conforming XLink applications when embedding as a remote element entity is not defined. Thus, the actual behaviour for this document is defined in subclause 5.5.3.</p>
@xlink:actuate	<p>Defines the desired timing of dereferencing a remote element entity from within an MPD as defined in W3C XLINK. In the context of this document, the attribute values shall only take one of the following:</p> <ol style="list-style-type: none"> 1) <code>onLoad</code>: an application should dereference the remote element entity immediately on loading the MPD. The remote element entity shall not contain another <code>@xlink:href</code> with <code>@xlink:actuate</code> set to <code>onLoad</code>. 2) <code>onRequest</code> (default): according to W3C Xlink, an application should dereference the remote element entity only on a post-loading event triggered for the purpose of dereferencing. In the context of this document, the application dereferences the link only for those resources it needs (or anticipates it probably will need) and at the time when it needs the content of the remote element entity for playout. Examples include dereferencing a link in a <code>Period</code> element when the play-time is expected to enter that <code>Period</code>, dereferencing an <code>Adaptation Set</code> link when it appears to contain <code>Representations</code> that will be needed, and so on.

5.5.3 Processing

The following rules apply to the processing of URI references within `@xlink:href`:

- 1) URI references to remote element entities that cannot be resolved shall be treated as invalid references and invalidate the URI and all `@xlink` attributes included in the element containing `@xlink:href` shall be removed.
- 2) URI references to remote element entities that are inappropriate targets for the given reference shall be treated as invalid references (see below for the appropriate targets) and invalidate the URI and all `@xlink` attributes included in the element containing `@xlink:href` shall be removed.
- 3) URI references to remote element entities that contain another `@xlink:href` attribute with `xlink:actuate` set to `onLoad` are invalid circular references and invalidate the URI. In this case, all `@xlink` attributes included in the element containing `@xlink:href` shall be removed.
- 4) The `xlink:href` may contain a URN as `urn:mpeg:dash:resolve-to-zero:2013`. If this value is present, the element containing the `xlink:href` attribute and all `@xlink` attributes

included in the element containing `@xlink:href` shall be removed at the time when the resolution is due.

- 5) If a URI reference is relative then reference resolution as defined in subclause 5.6.4 shall apply.

The remote element entity referenced with `@xlink:href` within an element of the MPD (referred to as MPD element) shall be formatted according to the following rules:

- 1) Only a single top-level element type of the same type as the MPD element shall be obtained from a remote element entity. However, multiple top-level elements of the same type may be obtained from a remote element entity unless explicitly restricted. All these elements shall have the same type as the element of the MPD containing `xlink`. If multiple top-level elements are obtained from the remote element entity, the elements shall be in appropriate order and the first element shall replace the element of the MPD containing the `xlink`. All additional top-level elements shall be inserted immediately after this element in the order in which they appear.

NOTE The remote element entity can be a non-valid XML document, when containing multiple top-level elements.

- 2) The remote element entity may contain another `@xlink:href` attribute with `@xlink:actuate` set to `onRequest`. In this case, the resolution to the referenced remote element entity is expected to happen only when the latter is needed again after processing the content in the returned document.

The remote element entity referenced from within an MPD (referred to as appropriate targets) shall be embedded into the MPD by applying the following rules:

- 1) If the remote element entity is empty, all `@xlink` attributes shall be removed from the element in the MPD and the remaining attributes and child elements shall not be changed.
- 2) If the remote element entity is non-empty, the original element in the MPD that contains `@xlink:href` shall be replaced with the content in the remote element entity. If multiple top-level elements are obtained from the remote element entity, the elements shall be in appropriate order and the first element shall replace the MPD element. All other top-level elements shall be inserted immediately after this element in the order in which they are declared.
- 3) All `XLINK` attributes initially present in the MPD shall be removed after dereferencing is completed.
- 4) All resources in the remote element entity referenced by `@xlink:href` shall have an availability end time as specified by **MPD**`@availabilityEndTime`.

Remote element entities are not valid MPDs and, when stored in a file system, should not use the `.mpd` file extension.

5.6 Base URL Processing

5.6.1 Overview

The **BaseURL** element may be used to specify one or more common locations for Segments and other resources. Reference resolution as defined in subclause 5.6.4 shall be applied to each URL in the MPD. Handling of multiple alternative base URLs is addressed in subclause 5.6.5.

The semantics of the attributes and elements for the Base URL are provided in 5.6.2, Table 30. The XML syntax of the Base URL is provided in subclause 5.6.3.

5.6.2 Semantics

Table 30 — Semantics of **BaseURL** element

Element or Attribute Name	Use	Description
BaseURL		A URL that can be used as Base URL. The content of this element is a URI string as described in subclause 5.6.4.
@serviceLocation	0	<p>This attribute specifies a relationship between Base URLs such that BaseURL elements with the same @serviceLocation value are likely to have their URLs resolve to services at a common network location, for example a common Content Delivery Network.</p> <p>If not present, no relationship to any other Base URL is known.</p>
@byteRange	0	<p>If present, specifies HTTP partial GET requests may alternatively be issued by adding the byte range into a regular HTTP-URL based on the value of this attribute and the construction rules in E.2.</p> <p>If not present, HTTP partial GET requests may not be converted into regular GET requests.</p> <p>NOTE Such alternative requests are expected to not be used unless the DASH application requires this. For more details, refer to Annex E.</p>
@availabilityTimeOffset	0	<p>specifies an offset to define the adjusted segment availability time. For semantics, refer to Table 14.</p> <p>If the value is present in SegmentBase then this attribute is additive to the one in SegmentBase. For details on processing, refer to 5.3.9.5.3.</p>
@availabilityTimeComplete	0	<p>specifies if all Segments of all associated Representation are complete at the adjusted availability start time. For semantics, refer to Table 14.</p> <p>If the value is present in SegmentBase then this attribute should not be present. If present in SegmentBase and BaseURL, the value in BaseURL shall be ignored.</p>
@timeShiftBufferDepth	0	<p>specifies the duration of the smallest time shifting buffer for any Representation in the MPD that is guaranteed to be available for a Media Presentation with type 'dynamic'.</p> <p>This value overrides MPD@timeShiftBufferDepth for the resources that use this BaseURL.</p> <p>This value of the attribute is undefined if the MPD@type attribute is equal to 'static'.</p>

Element or Attribute Name	Use	Description
@rangeAccess	OD Default: FALSE	<p>If set to <code>true</code>, partially available Segments may be accessed with byte range request. If a client is making a byte-range request against a partially available Segment and the first-byte position of that range request is non-zero and the client is expecting an aggregating response, then the client should signal that expectation which shall follow the convention of IETF RFC 8673. Specifically, it should use a <code>last-byte</code> value of 9007199254740991. This will signal the server to respond with a 206 aggregating response instead of waiting for the end of the Segment and responding with a 200 response code and a content-length response header.</p> <p>If set to <code>false</code>, the client should not expect a response corresponding to the requested byte range.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.6.3 XML syntax

```

<xs:complexType name="BaseURLType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Base URL
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attribute name="serviceLocation" type="xs:string"/>
      <xs:attribute name="byteRange" type="xs:string"/>
      <xs:attribute name="availabilityTimeOffset" type="xs:double"/>
      <xs:attribute name="availabilityTimeComplete" type="xs:boolean"/>
      <xs:attribute name="timeShiftBufferDepth" type="xs:duration"/>
      <xs:attribute name="rangeAccess" type="xs:boolean" default="false"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

5.6.4 Reference resolution

URLs at each level of the MPD are resolved according to IETF RFC 3986 with respect to the **BaseURL** element specified at that level of the document or the level above in the case of resolving base URLs themselves (the document "base URI" as defined in IETF RFC 3986:2005, subclause 5.1 is considered to be the level above the MPD level). If only relative URLs are specified and the document base URI cannot be established according to IETF RFC 3986 then the MPD should not be interpreted. URL resolution applies to all URLs found in MPD documents.

In addition to the document level (the level above the MPD level), base URL information may be present on the following levels:

- On MPD level in **MPD.BaseURL** element. For details, refer to subclause 5.3.1.2.
- On Period level in **Period.BaseURL** element. For details, refer to subclause 5.3.2.2.

- On Adaptation Set level in **AdaptationSet.BaseURL** element. For details, refer to subclause 5.3.3.2.
- On Representation level in **Representation.BaseURL**. For details, refer to subclause 5.3.5.2.

When a **BaseURL** element is present on the same level in the MPD as an actual URL, resolution of the actual URL includes the value of the **BaseURL**.

NOTE URL resolution is defined in RFC 3986:2005, section 5.2 and is *not* a simple concatenation. In particular, when a URL in the MPD is specified as a relative URL, then as per IETF RFC 3986 for the URL resulting from resolving the value of the **BaseURL** element, all characters following the last '/' part of its path part are replaced by the value of the actual URL attribute.

5.6.5 Alternative base URLs

If alternative base URLs are provided through the **BaseURL** element at any level, identical Segments shall be accessible at multiple locations. In the absence of other criteria, the DASH Client may use the first **BaseURL** element as "base URI". The DASH Client may use base URLs provided in the **BaseURL** element as "base URI" and may implement any suitable algorithm to determine which URLs it uses for requests.

If a **BaseURL** element containing an absolute URL is present on any level, it overwrites any **BaseURL** information present on a higher level.

5.7 Program information

5.7.1 Overview

Descriptive information on the program may be provided for a Media Presentation within the **ProgramInformation** element.

When multiple **ProgramInformation** elements are present, the `@lang` attribute shall be present and each element shall describe the Media Presentation sufficiently in the language defined by the value of the `@lang` attribute.

For each language, the program information may specify title, source of the program, copyright information, and a URL to more information.

The semantics of the attributes within the **ProgramInformation** element are provided in subclause 5.7.2, Table 31. The XML syntax of **ProgramInformation** element is provided in subclause 5.7.3.

5.7.2 Semantics

Table 31 — Program information semantics

Element or Attribute Name	Use	Description
ProgramInformation		specifies descriptive information about the program
<code>@lang</code>	0	Declares the language code(s) for this Program Information. The syntax and semantics according to IETF RFC 5646 shall be applied. If not present, the value is unknown.
<code>@moreInformationURL</code>	0	If provided, this attribute specifies an absolute URL which provides more information about the Media Presentation. If not present, the value is unknown.
Title	0 ... 1	specifies the title for the Media Presentation.

Element or Attribute Name	Use	Description
Source	0 ... 1	specifies information about the original source (for example content provider) of the Media Presentation.
Copyright	0 ... 1	specifies a copyright statement for the Media Presentation, usually starting with the copyright symbol, unicode U+00A9.
Key For attributes: M=mandatory, O=optional, OD=optional with default Value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.7.3 XML syntax

```

<xs:complexType name="ProgramInformationType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Program Information
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Title" type="xs:string" minOccurs="0"/>
    <xs:element name="Source" type="xs:string" minOccurs="0"/>
    <xs:element name="Copyright" type="xs:string" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="lang" type="xs:language"/>
  <xs:attribute name="moreInformationURL" type="xs:anyURI"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.8 Descriptors

5.8.1 General

The descriptor elements are all structured in the same way, namely they contain a @schemeIdUri attribute that provides a URI to identify the scheme and an optional attribute @value and an optional attribute @id. The semantics of the element are specific to the scheme employed. The URI identifying the scheme may be a URN or a URL.

In this document, specific elements for descriptors are defined in subclause 5.8.4.

The MPD does not provide any specific information on how to use these elements. It is up to the application that employs DASH formats to instantiate the description elements with appropriate scheme information. However, this document defines some specific schemes in subclause 5.8.5.

DASH applications that use one of these elements must first define a Scheme Identifier in the form of a URI and must then define the value space for the element when that Scheme Identifier is used. The Scheme Identifier appears in the @schemeIdUri attribute.

In the case that a simple set of enumerated values is required, a text string may be defined for each value and this string must be included in the @value attribute. If structured data is required then any extension element or attribute may be defined in a separate namespace.

The @id value may be used to refer to a unique descriptor or to a group of descriptors. In the latter case, descriptors with identical values for the attribute @id shall be synonymous, i.e. the processing of one of the descriptors with an identical value for @id is sufficient.

Two elements of type `DescriptorType` are *equivalent*, if the element name, the value of the `@schemeIdUri` and the value of the `@value` attribute are equivalent. If the `@schemeIdUri` is a URN, then equivalence shall refer to lexical equivalence as defined in IETF RFC 8141:2017, Clause 3. If the `@schemeIdUri` is a URL, then equivalence shall refer to equality on a character-for-character basis as defined in IETF RFC 3986:2005 subclause 6.2.1. If the `@value` attribute is not present, equivalence is determined by the equivalence for `@schemeIdUri` only. Attributes and element in extension namespaces are not used for determining equivalence. The `@id` attribute may be ignored for equivalence determination.

The semantics of the attributes within an element of the type `DescriptorType` are provided in subclause 5.8.2, Table 32. The XML schema definition of `DescriptorType` is provided in subclause 5.8.3. The specific descriptors follow these syntax and semantics.

5.8.2 Semantics of generic descriptor

Table 32 — Semantics of elements of type `DescriptorType`

Element or Attribute Name	Use	Description
Element of type <code>DescriptorType</code>		specifies a descriptor.
<code>@schemeIdUri</code>	M	specifies a URI to identify the scheme. The semantics of this element are specific to the scheme specified by this attribute. The <code>@schemeIdUri</code> may be a URN or URL. When a URL is used, it should also contain a month-date in the form <code>mm/yyyy</code> ; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date, to avoid problems when domain names change ownership.
<code>@value</code>	0	specifies the value for the descriptor element. The value space and semantics must be defined by the owners of the scheme identified in the <code>@schemeIdUri</code> attribute.
<code>@id</code>	0	specifies an identifier for the descriptor. Descriptors with identical values for this attribute shall be synonymous, i.e. the processing of one of the descriptors with an identical value is sufficient.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.8.3 XML syntax of generic descriptor

```
<xs:complexType name="DescriptorType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Descriptor
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="schemeIdUri" type="xs:anyURI" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
  <xs:attribute name="id" type="xs:string"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

5.8.4 Specific descriptors

5.8.4.1 Content protection

5.8.4.1.1 General

Content protection descriptors are signalled by the **ContentProtection** element and are used to provide content protection, encryption, and DRM related information in order to access encrypted and/or DRM-protected content. The **ContentProtection** element is an extended descriptor type.

For the element **ContentProtection**, the @schemeIdUri attribute is used to identify a content protection descriptor scheme.

The **ContentProtection** descriptors is expected to provide sufficient information, possibly in conjunction with the @value and/or extension attributes and elements, such as the DRM system(s), encryption algorithm(s), and key distribution scheme(s) employed, to enable a client to determine whether it can possibly play the protected content. The **ContentProtection** element can be extended in a separate namespace to provide information specific to a content protection scheme (e.g. particular key management systems or encryption methods).

When no **ContentProtection** element is present, the content shall not be encrypted.

Subclause 5.8.5.2 exclusively defines schemes indicating the encryption scheme for different MPEG formats. In subclause 5.8.5.2.2, the MP4 protection scheme identified with @schemeIdUri set to "urn:mpeg:dash:mp4protection:2011" is defined. In subclause 5.8.5.2.3, the MPEG-2 TS protection scheme identified with @schemeIdUri set to "urn:mpeg:dash:13818:1:CA_descriptor:2011" is defined. These schemes merely indicate the encryption mode used in conjunction with one or more DRMs. Hence, it is recommended to provide the client with additional content protection information about the available DRMs.

Additional **ContentProtection** elements with the @schemeIdURI attribute set to a URN not defined in subclause 5.8.5 must describe a specific key management and protection scheme that is sufficient to access and present the Representation. For such schemes, an optional @robustness attribute may be present, for details see subclause 5.8.4.1.2.

The content protection information may be provided explicitly in a **ContentProtection** element or may be referenced from another **ContentProtection** element. For details refer to subclause 5.8.4.1.3.

The semantics of the attributes within the **ContentProtection** element are provided in subclause 5.8.4.1.4, Table 30. The XML syntax of **ContentProtection** element is provided in subclause 5.8.4.1.5.

Subclause 5.8.4.1.6 provides information on how to define a content protection scheme.

5.8.4.1.2 Robustness

An optional @robustness attribute may be present in a **ContentProtection** element specifying the minimum robustness value required for playing back the associated Representations.

Robustness levels are generally used to differentiate implementations based on their level of robustness to attacks. Typically, the differentiation is based on whether decryption and decoding are performed in hardware or software. The set of robustness levels, their values, and the ordering are specific to each content protection scheme.

The robustness information may be provided to a DASH client for optimizing its processing. Typically, if the client ignores this information, it will identify whether its robustness level is sufficient for playing the associated Representations only after processing the DRM license server response.

For details defining robustness levels for a content protection scheme, refer to subclause 5.8.4.1.6.

NOTE Different DRMs can potentially have different robustness levels on the same device, and different codecs can have different robustness levels with the same DRM on the same device. A client may thus factor these in when selecting the DRM and codec.

5.8.4.1.3 Referencing

An MPD may contain multiple Representations or Adaptation Sets with identical **ContentProtection** descriptors. This redundancy may become significant in cases such as carriage of Protection System Specific Header ('pssh') information inside the **ContentProtection** descriptor.

For this purpose, a **ContentProtection** element may contain a @refId attribute that shall uniquely identify the element among all other **ContentProtection** elements in the MPD.

If a **ContentProtection** element with a @ref attribute is present, then exactly one **ContentProtection** element with a matching @refId attribute value shall be present.

A **ContentProtection** element, referred to as "referencing", may contain a @ref attribute that is set to the value of @refId of any **ContentProtection** element, referred to as "source". In this case the following holds:

- The "referencing" **ContentProtection** element inherits all attributes and elements from the "source" **ContentProtection** element.
- Any element or attribute present in the "referencing" **ContentProtection** element remains in the "referencing" **ContentProtection** element.
- The "source" **ContentProtection** element shall be included in a Period not later than the Period that contains in the "referencing" **ContentProtection**.

NOTE 1 The content author is expected to be careful about the consequences of a client not being able to dereference a "referencing" **ContentProtection** element. In this case, the client could either reject the associated Representations, or it could be forced to exchange additional information with the DRM server to identify the suitability of being able to access the associated Representations.

NOTE 2 A "source" **ContentProtection** element could be included in a Period which at some point is outside of the time shift buffer, yet still included in the MPD updates.

5.8.4.1.4 Semantics

Table 33 — Semantics of elements of Content Protection descriptor

Element or Attribute Name	Use	Description
ContentProtection		specifies information regarding a content protection or encryption scheme used to encrypt and/or protect the associated Representation(s)
@schemeIdUri	M	identifies a content protection or encryption scheme.
@value	0	provides additional information specific to the content protection or encryption scheme. For example, it may provide information such as DRM version, encryption mode, etc. For details, refer to subclause 5.8.4.1.6.

Element or Attribute Name	Use	Description
@ref	0	If present, makes this a referencing content protection descriptor that inherits from a “source” content protection descriptor which is identified by the equivalent value of @refId attribute. For details, refer to subclause 5.8.4.1.3. The attribute shall not be present if the @refId attribute is present.
@refId	0	specifies an identifier of this descriptor. The identifier shall be unique within an MPD. The attribute shall not be present if the @ref attribute is present.
@robustness	OD	specifies the robustness level required for this content protection scheme for accessing content represented by the associated Representation(s). For more details refer to subclause 5.8.4.1.2 and 5.8.4.1.6. If not present, then the lowest robustness level for the identified content protection scheme applies.
Key For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.8.4.1.5 Syntax

```

<xs:complexType name="ContentProtectionType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Content Protection
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="DescriptorType">
      <xs:attribute name="robustness" type="StringNoWhitespaceType"/>
      <xs:attribute name="refId" type="xs:ID"/>
      <xs:attribute name="ref" type="xs:IDREF"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

5.8.4.1.6 Definition of content protection schemes

Content Protection schemes may be defined and be used together with the content protection descriptor. In order to define a content protection scheme, this subclause defines several rules and recommendations for such a definition.

The following information is required for defining a content protection scheme:

- 1) A unique value of the scheme identifier @schemeIdUri
- 2) A list of permissible values for the @value attribute and their interpretation. There is no need to define a value if the scheme identifier itself is sufficient to convey the information.

The following information is recommended:

- 1) The usage and setting of the `@robustness` attribute. Robustness values, if defined, must be provided in an ordered list from low to high, representing the robustness level.
- 2) Example(s) of value discovery by the client using some frequently used APIs (e.g. W3C EME, Android, etc.).
- 3) How these values can be used by players in existing workflows.
- 4) Additional information that is important for the content protection scheme.

This document does not define content protection schemes in the sense as introduced above. However, this document provides in subclause 5.8.5.2 some well-defined encryption schemes to be used in context with external content protection schemes.

External organizations can define content protection schemes. If they do, they must provide the aforementioned required information and are encouraged to provide the recommended information.

Organizations are encouraged to document content protection schemes along with all additional information at DASH-IF Content Protection.

5.8.4.2 Role

For the element **Role**, the `@schemeIdUri` attribute is used to identify the role scheme employed to identify the role of the media content component. Roles define and describe characteristics and/or structural functions of media content components.

One Adaptation Set or one media content component may have assigned multiple roles even within the same scheme.

This document defines a simple role scheme in subclause 5.8.5.5.

In addition, this document defines other roles schemes to support signalling for multiple view signals in subclause 5.8.5.6.

5.8.4.3 Accessibility

For the element **Accessibility**, the `@schemeIdUri` attribute is used to identify the accessibility scheme employed. Accessibility is a general term used to describe the degree to which the DASH Media Presentation is available to as many people as possible.

NOTE **Accessibility** elements fulfil a very similar purpose with respect to media content components as for **Role** elements but are specifically intended for accessibility.

One Adaptation Set or one media content component may have assigned multiple accessibility purposes even within the same scheme.

This document does not define a specific accessibility scheme, but the simple role scheme in subclause 5.8.5.5 may be used to express a minimum amount of accessibility information.

5.8.4.4 Rating

For the element **Rating**, the `@schemeIdUri` attribute is used to identify the rating scheme employed.

Ratings specify that content is suitable for presentation to audiences for which that rating is known to be appropriate, or for unrestricted audiences.

NOTE If an audience has a rating restriction, it is intended that content that has associated ratings is not presented to that audience, unless at least one scheme is recognized and the rating value indicates that the content is appropriate to that audience.

This document does not define a rating scheme.

5.8.4.5 Viewpoint

For the element **Viewpoint**, the @schemeIdUri attribute is used to identify the viewpoint scheme employed.

Adaptation Sets containing non-equivalent **Viewpoint** element values contain different media content components. The **Viewpoint** elements may equally be applied to media content types that are not video.

Adaptation Sets with equivalent **Viewpoint** element values are intended to be presented together. This handling should be applied equally for recognized and unrecognized @schemeIdUri values.

This document does not define a viewpoint scheme.

5.8.4.6 Frame-packing

For the element **FramePacking**, the @schemeIdUri attribute is used to identify the frame-packing configuration scheme employed.

Multiple **FramePacking** elements may be present. If so, each element shall contain sufficient information to select or reject the described Representations.

NOTE If the scheme or the value for all **FramePacking** elements are not recognized, the DASH Client is expected to ignore the described Representations. A client can reject the Adaptation Set on the basis of observing a **FramePacking** element.

The descriptor may carry frame-packing schemes using the URN label and values defined for VideoFramePackingType in ISO/IEC 23091-3.

This document also defines frame-packing schemes in subclause 5.8.5.6. These schemes are maintained for backward-compatibility, but it is recommended to use the signalling as defined in ISO/IEC 23091-3.

5.8.4.7 Audio channel configuration

For the element **AudioChannelConfiguration**, the @schemeIdUri attribute is used to identify the audio channel configuration scheme employed.

Multiple **AudioChannelConfiguration** elements may be present indicating that the Representation supports multiple audio channel configurations. For example, it may describe a Representation that includes MPEG Surround audio supporting stereo and multichannel.

NOTE 1 If the scheme or the value for this descriptor is not recognized, the DASH Client is expected to ignore the descriptor.

The descriptor may carry audio channel configuration using the URN label and values defined for ChannelConfiguration in ISO/IEC 23001-8.

NOTE 2 In addition, a scheme for audio channel configuration is also defined in subclause 5.8.5.4. This scheme is maintained for backward-compatibility, but it is preferable to use the signalling as defined in ISO/IEC 23001-8.

5.8.4.8 Essential Property Descriptor

For the element **EssentialProperty**, the Media Presentation author expresses that the successful processing of the descriptor is essential to properly use the information in the parent element that contains this descriptor unless the element shares the same @id with another **EssentialProperty** element.

If **EssentialProperty** elements share the same @id, then processing one of the **EssentialProperty** elements with the same value for @id is sufficient. At least one **EssentialProperty** element of each distinct @id value is expected to be processed.

NOTE 1 If the scheme or the value for this descriptor is not recognized, the DASH Client is expected to ignore the parent element that contains the descriptor.

Multiple **EssentialProperty** elements with the same value for @id and with different values for @id may be present.

If one or more **EssentialProperty** elements sharing the same @id appear at the MPD level, this means that successful processing of at least one of these descriptors is essential to properly access and/or present the content described by this MPD.

NOTE 2 In the case when none of the **EssentialProperty** elements sharing the same @id can be successfully processed, the DASH Client is expected to terminate the media presentation.

The removal of an **EssentialProperty** descriptor in an MPD shall not change the conformance of this MPD to its associated schema and profile.

5.8.4.9 Supplemental Property Descriptor

For the element **SupplementalProperty**, the Media Presentation author expresses that the descriptor contains supplemental information that may be used by the DASH Client for optimized processing.

NOTE If the scheme or the value for this descriptor is not recognized, the DASH Client is expected to ignore the descriptor.

Multiple **SupplementalProperty** elements may be present.

5.8.4.10 Asset Identifier

The **AssetIdentifier** is used to identify the asset on Period level. If two different Periods contain equivalent Asset Identifiers then the content in the two Periods belongs to the same asset.

NOTE If the scheme or the value for this descriptor is not recognized, the **AssetIdentifier** element can still be used to understand the equivalence of Asset Identifiers across Periods. Processing of the descriptor scheme and value by the DASH Client is not essential for normal operation.

5.8.4.11 UTC Timing Descriptor

Using the **UTCTiming** element, the Media Presentation author provides additional information for the client to optionally obtain wall-clock time to be used in Media Presentation. If multiple schemes are specified by the Media Presentation author, their order indicates their relative preference, first having the highest, and the last having the least priority. However, the client may choose any method, potentially having to deal with reduced accuracy.

The value of the @schemeIdURI for this descriptor shall be restricted to one of the identifiers documented in the first column "@schemeIdURI" in subclause 5.8.5.7, Table 35.

5.8.4.12 Output protection

5.8.4.12.1 General

Output protection schemes, such as High-bandwidth Digital Content Protection, Digital Content Protection, LLC HDCP, are frequently used to protect the rendered output in transit from the decoding device to the displaying device (for example, from a set-top box connected to a TV set using an HDMI cable). These output protection schemes and their properties are frequently a pre-requisite for the ability to present digital media content.

The **OutputProtection** element is a descriptor type and this descriptor may be used to identify an output protection descriptor scheme that is required to present the associated Representation(s).

The ultimate decision for whether a Representation can be rendered by a client is typically defined by the content usage rules, usually documented in the license provided by the DRM license server. However, the information contained in the output protection descriptor can be used by the client to select the associated Representation or Adaptation Set and start downloading without waiting for the license server response.

This information is provided to the DASH client to optimize its processing. Typically, if the client ignores this information, it will identify whether its output environment is capable to play the associated Representations after processing detailed DRM information.

When an **OutputProtection** element is not present, the client should not make any assumptions on whether output protection is indeed a prerequisite for content presentation.

NOTE The authoritative answer to the question “can the client present this representation” is still given by the DRM license server response. Hence, even if the descriptor is not understood, incorrect, or ignored, it would not result in incorrect behaviour, only in reduced efficiency.

5.8.4.12.2 Output protection schemes definition

Output protection schemes may be defined and be used in DASH together with the output protection descriptor. In order to define an output protection scheme, this subclause defines several rules and recommendations for the definition of this.

For defining an output protection scheme the following information is required:

- 1) A unique value of `@schemeIdUri`
- 2) The list of permissible values for the `@value` attribute in non-decreasing order, and their interpretation

The following information is recommended:

- 1) Example(s) of value discovery by the client using some frequently used APIs (e.g. W3C EME, Android, etc.)
- 2) Example(s) of how these values would be used by the players to reject potentially unplayable associated representations

Output protection schemes defined in this document are provided in subclause 5.8.5.14.

External organizations can define output protection schemes. If they do, they must provide the required information from above and are encouraged to provide the recommended versions.

Organizations are encouraged to document content protection schemes along with all additional information at DASH-IF Output Protection.

5.8.5 Specific scheme definitions

5.8.5.1 General

The definition of specific schemes (both syntax and semantics) to be used in any of the descriptor elements requires the definition of the URI by the authors to link the content description to the Media Presentation. In subclause 5.8.5, some schemes and scheme identifiers are defined to enable usage of existing code points in combination with this document as well as to provide simple means to support different functionalities.

5.8.5.2 Content protection

5.8.5.2.1 General

The remainder of subclause 5.8.5.2 defines a set of URIs that identify specific encryption schemes. Note that these defined schemes do not follow the rules of subclause 5.8.4.1.6 as they are exclusively used to identify the encryption scheme rather than a full content protection scheme. However, they are supposed to be used with the **ContentProtection** descriptor.

5.8.5.2.2 The MP4 Protection Scheme

For Representations based on ISO/IEC 14496-12, the following URI is defined to indicate protection schemes identified by the Scheme Type within the Scheme Type Box of the Protection Scheme Information Box of the file:

```
urn:mpeg:dash:mp4protection:2011
```

In this scheme, the @value attribute shall present and shall be the 4CC contained in the Scheme Type Box, suitably percent-encoded according to IETF RFC 8141, and may include the version number. The 4CC and the version number, if present, shall be separated by a ":". The version number shall be encoded as up to 8 hexadecimal digits, where the leading '0's may be omitted.

The syntax for the value field of the content protection descriptors shall follow the **MP4-PROTECTION-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
MP4-PROTECTION-VALUE = VERSIONED-4CC

VERSIONED-4CC = CCCC [ CCCC-VERSION ]
CCCC = ALPHA ALPHA ALPHA ALPHA
CCCC-VERSION = ":" 1*8HEX

CA-DESCRIPTOR-VALUE = LHEX LHEX LHEX LHEX

LHEX = DIGIT / "a" / "b" / "c" / "d" / "e" / "f"
```

The value space of the @robustness attribute is not defined for this scheme and therefore the attribute shall not be present.

If the urn:mpeg:dash:mp4protection:2011 content protection scheme is assigned to a Representation, it merely indicates the encryption mode used in conjunction with one or more DRMs. Hence, it is recommended to provide the client with additional information about the available DRMs.

If the value of the Scheme Type Box (and hence the value of the @value attribute of the MP4 protection scheme) is defined in ISO/IEC 23001-7, then as defined in 11 of ISO/IEC 23001-7, the following applies:

- 1) The MPD author is encouraged to provide additional DRM-specific **ContentProtection** elements. These typically use the @schemeIdUri property to carry a UUID, as defined in IETF RFC 4122, of the specific content protection scheme and may include additional scheme-specific information.
- 2) The presence, syntax, and semantics of the @value attributes are governed by the specific content protection scheme.
- 3) The @cenc:default_KID attribute as defined in ISO/IEC 23001-7 should be present in the **ContentProtection** element with cenc defined as the namespace xmlns:cenc="urn:mpeg:cenc:2013" as defined in ISO/IEC 23001-7.
- 4) A media player application may read the @cenc:default_KID value to determine whether that key has been acquired, and may acquire a license using the information in a cenc:pssh element in advance of media availability.

- 5) The carriage of the scheme-specific `cenc:pssh` element is recommended.

Organizations are encouraged to document content protection schemes along with all additional information at DASH-IF Content Protection.

5.8.5.2.3 The CA_descriptor scheme

For Representations based on ISO/IEC 13818-1 (MPEG-2 Transport Stream), the following URI is defined to indicate the Conditional Access System used:

```
urn:mpeg:dash:13818:1:CA_descriptor:2011
```

In this scheme, the value of the `@value` attribute shall be the 4-digit lower-case hexadecimal Representation of the 16-bit `CA_system_ID` from the `CA_descriptor` as defined in ISO/IEC 13818-1.

The value space of the `@robustness` attribute is not defined for this scheme and therefore the attribute shall not be present.

5.8.5.3 Frame-packing

The following defines a set of URIs that identify specific frame-packing arrangements, i.e. schemes contained in the **FramePacking** element:

- For Adaptation Sets or Representations that contain a video component that conforms to ISO/IEC 14496-10^[19], the URI `urn:mpeg:dash:14496:10:frame_packing_arrangement_type:2011` is defined. The `@value` shall be value as defined for `VideoFramePackingType` in ISO/IEC 23091-2.
- For Adaptation Sets or Representations that contain a video component that conforms to ISO/IEC 13818-1, the URI `urn:mpeg:dash:13818:1:stereo_video_format_type:2011` is defined. The `@value` shall be value as defined for `VideoFramePackingType` in ISO/IEC 23091-2.

The syntax for the value field of the frame packing descriptors shall follow the **FRAME-PACKING-ARRANGEMENT-TYPE-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
FRAME-PACKING-ARRANGEMENT-TYPE-VALUE = VIDEO-FRAME-PACKING-TYPE
STEREO-VIDEO-FORMAT-TYPE = VIDEO-FRAME-PACKING-TYPE
VIDEO-FRAME-PACKING-TYPE = 1*2DIGIT
```

5.8.5.4 Audio channel configuration schemes

The following defines a URI that identifies channel configuration signalling for Representations that contain an audio component. The URI `"urn:mpeg:dash:23003:3:audio_channel_configuration:2011"` is defined to indicate the channel configuration as defined by Table 68 (Channel Configurations, meaning of `channelConfigurationIndex`, mapping of channel elements to loudspeaker positions') of ISO/IEC 23003-3. The `@value` shall be the value as defined for `ChannelConfiguration` in ISO/IEC 23091-3.

The URN `"urn:mpeg:dash:outputChannelPositionList:2012"` defines a list of output channel positions to signal individual speaker positions. The `@value` shall be a space-delimited list of values as defined of the `OutputChannelPosition` as defined in ISO/IEC 23091-3. For example, the `@value` for the subclause 7.1 channel configuration 2 high as 2/0/0, 5 mid as 3/0/2 and 0.1 low, where a/b/c indicates speaker count in front, side and back, respectively and 0.1 indicates a subwoofer channel), is `"2 0 1 4 5 3 17 18"`.

5.8.5.5 DASH role scheme

The URN "urn:mpeg:dash:role:2011" is defined to identify the role scheme defined in Table 34. **Role@value** shall be assigned to Adaptation Sets that contain a media component type to which this role is associated.

Table 34 — Role@value attribute for scheme with a value "urn:mpeg:dash:role:2011"

Role@value	Description	Applicable Media Type(s)
caption	Captions (see NOTE 3 below).	video, text
subtitle	Subtitles (see NOTE 3 below).	video, text
Main	Main media component(s) which is/are intended for presentation if no other information is provided.	any
alternate	Media content component(s) that is/are an alternative to (a) main media content component(s) of the same media component type (see NOTE 2 below).	any
supplementary	Media content component that is supplementary to a media content component of a different media component type (see NOTE 1 below).	any
commentary	Experience that contains a commentary (e.g. director's commentary) (typically audio).	audio, text
dub	Experience that contains an element that is presented in a different language from the original (e.g. dubbed audio, translated captions).	audio, text
description	Textual or audio media component containing a textual description (intended for audio synthesis) or an audio description describing a visual component.	audio, text
sign	Visual media component representing a sign-language interpretation of an audio component.	video
metadata	Media component containing information intended to be processed by application specific elements.	text, application
enhanced-audio-intelligibility	Experience containing an element for improved intelligibility of the dialogue.	audio
emergency	Experience that provides information, about a current emergency, that is intended to enable the protection of life, health, safety, and property, and may also include critical details regarding the emergency and how to respond to the emergency.	any
forced-subtitle	Textual information meant for display when no other text representation is selected. It is used to clarify dialogue, alternate languages, texted graphics or location/person IDs that are not otherwise covered in the dubbed/localized audio.	text
easyreader	Simplified or reduced captions as specified in [United States Code Title 47 CFR 79.103(c)(9)].	text, video

Role@value	Description	Applicable Media Type(s)
karaoke	Textual representation of a songs' lyrics, usually in the same language as the associated song. See SMPTE ST 2067-2.	any

NOTE 1 A normal audio/video program labels both the primary audio and video as "main". However, when the two media component types are not equally important, for example (a) video providing a pleasant visual experience to accompany a music track that is the primary content or (b) ambient audio accompanying a video showing a live scene such as a sports event, that is the primary content, the accompanying media can be assigned a "supplementary" role.

NOTE 2 Alternate media content components are expected to carry other descriptors to indicate in what way it differs from the main media content components (e.g. a Viewpoint descriptor or a Role descriptor), especially when multiple alternate media content components including multiple supplementary media content components are available.

NOTE 3 Open ("burned in") captions or subtitles would be marked as media type component "video" only, but having a descriptor saying "caption" or "subtitle".

NOTE 4 Role descriptors with values such as "subtitle", "caption", "description", "sign" or "metadata" can be used to enable assignment of a "kind" value in W3C HTML 5 applications for tracks exposed from a DASH MPD.

The syntax for the value field of the role descriptor shall follow the **ROLE-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
ROLE-VALUE = "caption" / "subtitle" / "main" / "alternate" / "supplementary"
ROLE-VALUE =/ "commentary" / "dub" / "description" / "sign" / "metadata"
ROLE-VALUE =/ "enhanced-audio-intelligibility" / "emergency"
ROLE-VALUE =/ "forced-subtitle" / "easyreader" / "karaoke"
```

5.8.5.6 DASH Multiple views scheme

This scheme is defined for multiple views media content description.

This scheme may be used with the **Role** descriptor applied to a **ContentComponent** element of type video or to an **AdaptationSet** element. If this scheme is applied to an **AdaptationSet** element containing multiple views, each media content component of type video shall have a **ContentComponent** element that contains the **Role** descriptor using this scheme. A **Role** element of this scheme is used to indicate which views amongst the views comprising multiple presentable stereo pairs the contained media content component or components represent. If N views are available that can be combined into M valid stereo pairs, the **Role** with @schemeIdURI equal to "urn:mpeg:dash:stereoid:2011" signals which views form a stereo pair and which one is the left view and which one is the right view of each stereo pair. The @value of the Role element shall contain a space-delimited list of view indicators 'li' or 'rj' where i, j are non-negative decimal integers. A stereo pair i ($0 \leq i < M$) is formed by using a view whose **Role** element includes the view indicator 'li' as the left view and a view whose **Role** element contains the view indicator 'ri' as the right view. Within the @value attribute, view indicators shall be ordered with all left view indicators preceding all right view indicators and within each group in ascending order of view index.

The syntax for the value field of the role descriptor shall follow the **STEREID-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
STEREID-VALUE = LEFT-VIEW-INDICATOR * [ SP LEFT-VIEW-INDICATOR ]
RIGHT-VIEW-INDICATOR * [ SP RIGHT-VIEW-INDICATOR ]

LEFT-VIEW-INDICATOR = "l" DECIMAL_DIGITS
RIGHT-VIEW-INDICATOR = "r" DECIMAL_DIGITS

DECIMAL_DIGITS = 1 * DIGIT
```

5.8.5.7 DASH UTC Timing Schemes

This document defines several methods, specified in Table 35, by which DASH Clients can obtain wall-clock times as used by the Media Presentation. Specifically, this clock is synchronized to the one used to generate the MPD.

Table 35 — Different UTC timing Methods

@schemeIdURI	Description
urn:mpeg:dash:utc:ntp:2014	<p>The identifier indicates that the @value contains a white space separated list of servers that are recommended to be used in combination with the NTP protocol as defined in IETF RFC 5905 for getting the appropriate time.</p> <p>NOTE Multiple servers can be used to improve accuracy.</p> <p>Use of NTP servers not specified in the @value attribute is allowed.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>
urn:mpeg:dash:utc:sntp:2014	<p>The identifier indicates that the @value contains a white space separated list of servers that are recommended to be used in combination with the SNTP protocol as defined in IETF RFC 5905 for getting the appropriate time.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>
urn:mpeg:dash:utc:http-head:2014	<p>The identifier indicates that the @value contains a white space separated list of HTTP URLs that are recommended to be used in combination with the HTTP protocol as defined in IETF RFC 7230 for getting the appropriate time.</p> <p>The value of the @value attribute contains a white space separated list of HTTP URLs to which HTTP HEAD requests can be made to obtain the Date information in the HTTP Header providing the wall-clock time for this Media Presentation.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>
urn:mpeg:dash:utc:http-xsdate:2014	<p>The identifier indicates that the @value contains a white space separated list of HTTP URLs that are recommended to be used in combination with the HTTP protocol as defined in IETF RFC 7230 for getting the appropriate time.</p> <p>The value of the @value attribute contains a white space separated list of HTTP URLs to which HTTP GET requests can be made to obtain the timing information. The timing information is contained in the message body of the HTTP response to the above HTTP GET request and contains the time value which shall be formatted according to xs:dateTime as defined in W3C XML Schema Part 2: Datatypes specification. This value is based on a wall clock synchronized to the one used to generate the MPD.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>

@schemeIdURI	Description
urn:mpeg:dash:utc:http-iso:2014	<p>The identifier indicates that the @value contains a white space separated list of HTTP URLs that are recommended to be used in combination with the HTTP protocol as defined in IETF RFC 7230 for getting the appropriate time.</p> <p>The value of the @value attribute contains a white space separated list of HTTP URLs to which HTTP GET requests can be made to obtain the timing information. The timing information is contained in the message body of the HTTP response to the above HTTP GET request and contains time value formatted according to ISO time code as defined in ISO/IEC 8601. This value is based on a wall clock synchronized to the one used to generate the MPD.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>
urn:mpeg:dash:utc:http-ntp:2014	<p>The identifier indicates that the @value contains a white space separated list of HTTP URLs that are recommended to be used in combination with the HTTP protocol as defined in IETF RFC 7230 for getting the appropriate time.</p> <p>The value of the @value attribute contains a white space separated list of HTTP URLs to which HTTP GET requests can be made to obtain the timing information. The timing information is contained in the message body of the HTTP response to the above HTTP GET request and contains time value formatted according to formatted according to NTP timestamp format in IETF RFC 5905. This value is based on a wall clock synchronized to the one used to generate the MPD.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>
urn:mpeg:dash:utc:direct:2014	<p>The identifier indicates that the @value field, contains the time value which shall be formatted according to xs:dateTime as defined in W3C XML Schema Part 2: Datatypes. This value is based on a wall clock synchronized to the one used to generate the MPD.</p> <p>For exact syntax of the value of the @value attribute, see below.</p>

The syntax for the value field of the UTC Timing descriptor with @schemeIdURI set to "urn:mpeg:dash:utc:ntp:2014" or set to "urn:mpeg:dash:utc:sntp:2014" shall follow the **NTP-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
NTP-VALUE = TIME-SERVER *[ WS TIME-SERVER ]
TIME-SERVER = host [ ":" port ] ; host and port are declared in
; augmented IETF RFC 6874
```

The syntax for the value field of the UTC Timing descriptor with @schemeIdURI set to "urn:mpeg:dash:utc:http-head:2014", set to "urn:mpeg:dash:utc:http-xsdate:2014", set to "urn:mpeg:dash:utc:http-iso:2014" or set to "urn:mpeg:dash:utc:ntp:2014" shall follow the **HTTP-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:


```
HTTP-VALUE = httpurl * [ WS httpurl ] ; httpurl is defined in
IETF RFC 1738
```

The syntax for the value field of the UTC Timing descriptor with @schemeIdURI set to "urn:mpeg:dash:utc:direct:2014" shall follow the **DIRECT-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
DIRECT-VALUE = [ "-" ] 4DIGIT "-" 2DIGIT "-" 2DIGIT "T" 2DIGIT ":"
2DIGIT ":" 2DIGIT [ "Z" / ( "+" / "-" ) 2DIGIT ":" 2DIGIT ]
```

5.8.5.8 Audio Receiver Mix

This clause defines a scheme for use in **EssentialProperty** or **SupplementaryProperty** to indicate that two audio Adaptation Sets need to be mixed by the media engine prior to playback.

The @schemeIdUri attribute identifying the scheme is urn:mpeg:dash:audio-receiver-mix:2014.

The @value attribute shall contain the value of the **AdaptationSet@id** from an Adaptation Set with content type audio attribute with which the current Adaptation Set needs to be mixed with in order to provide complete audio experience.

An example of receiver mix is the case where a single audio Adaptation Set provides music and effects — i.e. complete experience without dialogues, and one or more Adaptation Sets provide dialogues in different languages. In this case, the dialogue Adaptation Sets depends on music and effects Adaptation Set.

The mixing requirement is unidirectional — i.e. requirement of mixing Representation A with Representation B when A is selected does not imply mixing the two is required if B is selected.

The syntax for the value field of the audio receiver mix descriptor shall follow the **AUDIO-RECEIVER-MIX-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
AUDIO-RECEIVER_MIX-VALUE = ADAPTATION-SET-ID
ADAPTATION-SET-ID = DECIMAL_DIGITS
DECIMAL_DIGITS = 1*DIGIT
```

5.8.5.9 DASH MPD Adaptation Set Linking scheme

The URN "urn:mpeg:dash:mpd-as-linking:2015" is defined in order to provide information that the same Adaptation Set can be found in another MPD. The scheme may be used with Essential Property Descriptors together with an **EmptyAdaptationSet** element or with Supplemental Property Descriptors. The value provides

- a URL to the MPD, including appropriate anchors for Periods and Adaptation Set as defined in C.4.2, and
- optionally, a timeline offset field to synchronize the data added as a white space separated second item. If the value is not present, then the media time on the original MPD and the linked MPD are identical. If a timeOffset field is added, then this value expresses the difference between the media time of the Adaptation Set of the linked MPD and the media time in the originating MPD. The number may be positive or negative. The timeoffset is in unit of @timescale of the linked Adaptation Sets.

The Descriptor shall only be used in combination with Adaptation Sets (regular or empty ones). If all information is contained in the Adaptation Set, then a Supplemental Descriptor or an Essential Descriptor may be used. If the Adaptation Set does neither contain a **Representation** element, nor an @xlink attribute, then an Empty Adaptation Set as well as an Essential Descriptor shall be used to indicate that the Adaptation Set is not fully described in this MPD.

NOTE This scheme can be used for many use cases. However, one specific use case is the server-based mosaic channel as described in ISO/IEC TR 23009-3^[20]. The use case can be fulfilled by the combination of the Spatial Relationship Description (SRD) as defined in Annex H and this scheme. In this case, the same value of `source_id` parameter in SRD scheme indicates the association among Adaptation Set(s) (regular and empty ones if present) even though Representation is not present in an Empty Adaptation Set.

The syntax for the value field of the adaptation set linking descriptor shall follow the **MPD-AS-LINKING-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
MPD-AS-LINKING-VALUE = MPD-URL [ WSP TIMELINE-OFFSET ]
MPD-URL = httpurl ; defined in IETF RFC 1738
TIMELINE-OFFSET = "timeOffset=" [ "-" ] DECIMAL_DIGITS
DECIMAL_DIGITS = 1 * DIGIT
```

5.8.5.10 Sub-Asset Identifier scheme

In DASH MPD, sub-assets across Periods can be identified using the sub asset Scheme Identifier. This scheme is signalled using a specific **SupplementalProperty** descriptor at the Adaptation Set or Sub-Representation level with `@schemeIdUri` attribute set to "urn:mpeg:dash:sai:2015".

If two different Adaptation Sets or Sub-Representation from different Periods contain Sub-Asset Identifiers descriptors with the same `@value` attribute, then the content in the Representation(s) contained in these Adaptation Sets represent, at least, the same sub-asset.

NOTE 1 The association between sub-assets and an Adaptation Set can change across Periods. For instance, an Adaptation Set can be associated with a sub-asset of an asset in one Period but with another sub-asset of the same asset in another Period.

NOTE 2 Sub-Asset Identifier descriptor can be used by DASH Clients to select Representation(s) to be processed after a Period change.

A given Sub-Representation or Adaptation Set can contain more than one Sub-Asset Identifier descriptor indicating that this Sub-Representation or the Representation(s) contained in this Adaptation Set, respectively, represent more than one identified part of the asset.

NOTE 3 If the value for this descriptor is not recognized, the SubAsset Identifier descriptor can still be used to understand the equivalence of sub-asset identifiers across Periods. Processing of the descriptor scheme and value by the DASH Client is not essential for normal operation.

NOTE 4 Different Adaptation Sets or Sub-Representations in a Period do not contain the same SubAsset-Identifier, i.e. the same `@schemeIdUri` and `@value` is dis-allowed.

The syntax for the value field of the Sub-Asset identifier descriptor shall follow the **SAI-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
SAI-VALUE = STRING
STRING = *VCHAR
```

5.8.5.11 Client Authentication and Content Access Authorization

When client authentication and/or content access authorization functionality is needed, DASH may be used with different schemes such as Open Authentication Technical Committee (OATC) Online Multimedia Authorization Protocol (OMAP)^[13], Open Standard for Authorization (OAuth) 2.0^[14], OASIS Security Assertion Markup Language (SAML)^[15], 3GPP Generic Authentication Architecture (GAA)^[16], or 3GPP Generic Bootstrapping Architecture (GBA)^[17]. This section describes generic signalling to support use of various authentication and authorization schemes.

Typical access control methods include blocking HTTP requests that do not include a security token obtained by the authorization protocol wherein the security token is validated by a CDN before

downloading the requested Media Segment, or encrypting Media Segments so that playback will be restricted unless the authorization protocol provides the client with a decryption key.

For client authentication, a service may limit content delivery to authenticated clients, and may use client identification information such as certificates, cookies, and embedded keys to determine subscription rights, etc. required to authorize playback of the Media Presentation. The details of such a scheme are outside the scope of this document. A client that does not support the signalled content access authorization would not be able to play the content.

NOTE This subclause does not provide any requirements on client authentication or content access authorization.

The signalling and setup of the specific scheme may be done outside the MPD level, e.g. a system applying such a scheme may only permit access to the MPD if the client is authenticated or the content access is authorized.

However, there are cases for which the MPD is provided without access control. In this case, client authentication and content access authorization methods may be signalled in the MPD using Essential Property Descriptors. The DASH Client may be offered with multiple options to access the entire Media Presentation or specific parts of the Media Presentation, e.g. a specific Adaptation Set. An Essential Property descriptor may be placed at the appropriate level, and, for example, the **EssentialProperty**@schemeIdUri may signal the URN of the appropriate authentication or authorization method and the **EssentialProperty**@value attribute may carry some scheme specific information. Other signalling may be used, but the detailed signalling and semantics remain specific to a particular scheme.

There may be cases in which multiple options are provided to the client for client authentication and/or content access authorization. In this case, the **EssentialProperty**@id value may be used to signal functional equivalence of descriptors. In the absence of other information, the **EssentialProperty**@id value may contain the following URNs:

- urn:mpeg:dash:client-authentication:2015 for client authentication.
- urn:mpeg:dash:content-authorization:2015 for content access authorization.

In this case,

- each **EssentialProperty** descriptor with **EssentialProperty**@id value of urn:mpeg:dash:client-authentication:2015 indicates a supported client authentication protocol. A client may select one of possibly multiple elements with that **EssentialProperty**@id value, and a scheme that it recognizes based on the @schemeIdUri attribute; and execute that protocol using any information included in the @value attribute and any extension elements defined by that particular authentication scheme.
- each **EssentialProperty** descriptor with **EssentialProperty**@id value of urn:mpeg:dash:content-authorization:2015 indicates a supported content access authorization protocol. A client may select one of possibly multiple elements with that **EssentialProperty**@id value, and a scheme that it recognizes based on the @schemeIdUri attribute; and execute that protocol using any information included in the @value attribute and any extension elements defined by that particular authorization scheme.

A DASH Client that is successfully authenticated as an authenticated player and authorized for playback of some or all Representations or Adaptation Sets in the MPD may request and play the authorized content.

5.8.5.12 Audio Interactivity Descriptor

A scheme is defined to be used with an Essential Property or Supplemental Property Descriptor as "urn:mpeg:dash:audio-interactivity:2016".

This descriptor indicates if the associated audio content (Adaptation Set, Preselection or Representation) contains media components that are enabled for user interactivity through associated metadata. The descriptor is used e.g. to facilitate user interface (UI) resource management in the receiving client. Interactivity involves user interaction with elements, i.e. the user can modify dynamically for example the gain, spatial position or mute/unmute status of audio elements. Therefore, a UI is required to enable this kind of personalization during playback. A supplemental descriptor should be used if a UI is not mandatory to select and play the corresponding audio elements. An essential descriptor should be used if a UI is mandatory in order to play the corresponding audio elements. The @value attribute is owned by the codec in use. The detailed semantics of the descriptor are also owned by the codec in use.

5.8.5.13 Quality Equivalence Descriptor

A scheme is defined to be used with a Supplemental Property Descriptor as "urn:mpeg:dash:qr-equivalence:2019". This descriptor enables an indication that the values provided with each @qualityRanking hold across Adaptation Sets that are included in the equivalence indicator. In this case the value of all provided @qualityRanking values provides an equivalence.

This descriptor may be declared at MPD, Period or Preselection level. Multiple quality equivalence descriptors may be present in an MPD.

The value of the descriptor provides a comma-separated list of Adaptation Set identifiers of the Adaptation Sets that have equivalent quality ranking. If the @value attribute is absent, all Adaptation Sets in the underlying hierarchy have equivalent quality ranking.

The syntax for the value field of the quality ranking equivalence descriptor shall follow the **QR-EQUIVALENCE-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
QR-EQUIVALENCE-VALUE = AS-ID [ WSP AS-ID ]
AS-ID = DECIMAL_DIGITS
DECIMAL_DIGITS = 1*DIGIT
```

5.8.5.14 Output protection schemes

5.8.5.14.1 General

The remainder of subclause 5.8.5.14 defines a set of URIs that identify specific output protection schemes following the rules of subclause 5.8.5.14.2.

5.8.5.14.2 HDCP Output Protection Scheme

The subclause defines an output protection scheme for HDCP, Digital Content Protection, LLC HDCP, with @schemeIdUri defines "urn:mpeg:dash:output-protection:hdc:2020" and the value as defined in Table 36. The value provides an association between the content and the device output requirements, namely the minimum HDCP version that a device output must comply with to output the associated content.

Table 36 defines an order list with increasing rows defining increasing levels.

Table 36 — Value definition of HDCP Output protection scheme with @schemeIdURI set to "urn:mpeg:dash:output-protection:hdc:2020"

@value	Interpretation (Output compliance requirement for content)
1.0	HDCP 1.0-compliant device
1.1	HDCP 1.1-compliant device
1.2	HDCP 1.2-compliant device
1.3	HDCP 1.3-compliant device
1.4	HDCP 1.4-compliant device
2.0	HDCP 2.0-compliant device
2.2	HDCP 2.2-compliant device
2.3	HDCP 2.3-compliant device

5.9 DASH metrics descriptor

5.9.1 Overview

This document does not define mechanisms for reporting metrics; however, it does define a set of metrics and a mechanism that may be used by the service provider to trigger metric collection and reporting at the clients, if a reporting mechanism is available. The trigger mechanism is based on the **Metrics** element in the MPD. The element contains the list of DASH Metrics for which the measurements are desired, the time interval and the granularity for the measurements, as well as the scheme according to which the metric reporting is desired.

The semantics of the attributes within the **Metrics** element are provided in subclause 5.9.2, Table 37. The XML syntax of **Metrics** element is provided in 5.9.3.

The semantics of the **Reporting** element are provided in subclause 5.9.4.

5.9.2 Semantics

Table 37 — Semantics of **Metrics** element

Element or Attribute Name	Use	Description
Metrics		DASH metric element
@metrics	M	specifies all DASH Metrics that the client is desired to report as a list of DASH Metric keys. The keys shall be used as defined in Annex D, separated by a comma.
Range	0 ... N	specifies the time period during which DASH Metrics collection is requested. When not present, DASH Metrics reporting is requested for the whole duration of the content.
@starttime	0	<p>specifies the start time of the DASH Metrics collection operation. When not present, DASH Metrics collection is requested from the beginning of content consumption.</p> <p>For services with MPD@type= 'dynamic ', the start time is indicated in wall clock time by adding the value of this attribute to the value of the MPD@availabilityStartTime attribute.</p> <p>For services with MPD@type= 'static ', the start time is indicated in Media Presentation time and is relative to the <i>PeriodStart</i> time of the first Period in this MPD.</p> <p>NOTE For example, if MPD@availabilityStartTime is 14:30 and the metrics collection is intended to start at 14:45, then @starttime is 0:15.</p>
@duration	0	<p>specifies the duration of the DASH metrics collection interval. The value of the attribute expresses in Media Presentation time.</p> <p>If not present, the value is identical to the Media Presentation duration.</p>
Reporting	1 ... N	<p>specifies information about the requested reporting method and formats.</p> <p>For more details, refer to subclause 5.9.4.</p>
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are bold ; attributes are non-bold and preceded with an @.		

5.9.3 XML syntax

```

<xs:complexType name="MetricsType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Metrics
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Range" type="RangeType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Reporting" type="DescriptorType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="metrics" type="xs:string" use="required"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <xs:complexType name="RangeType">
      <xs:annotation>
        <xs:documentation xml:lang="en">
          Metrics Range
        </xs:documentation>
      </xs:annotation>
      <xs:attribute name="starttime" type="xs:duration"/>
      <xs:attribute name="duration" type="xs:duration"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>

```

5.9.4 Metric reporting

DASH Clients should collect metrics based on the **Metric** element and report the collected metrics using one of the reporting schemes in the **Reporting** descriptor in the **Metrics** element.

It is expected that elements containing unrecognized reporting schemes are ignored by the DASH Client.

If multiple **Reporting** elements are present, it is expected that the client processes one of the recognized reporting schemes.

No reporting scheme is specified in this document. It is expected that external specifications may define formats and delivery for the reporting data. External specifications defining a reporting scheme should take specific care to respect privacy issues.

5.10 Events

5.10.1 Overview

Events may be provided in the MPD or within a Representation in order to signal aperiodic information to the DASH Client or to an application. Events are timed, i.e. each event starts at a specific media presentation time and may have a duration. Events include DASH specific signalling or application-specific events. DASH events are identified by scheme identifiers defined in this document. For application specific events, a scheme identifier identifies the application such that the DASH Client can forward the event to the proper application.

Events of the same type are clustered in Event Streams by the same scheme/value pair. This enables a DASH Client or the application to subscribe to an Event Stream of interest and ignore Event Streams that are of no relevance or interest.

Two ways of signalling events are provided, namely:

- events signalled in the MPD as defined in subclause 5.10.2,
- events signalled inband in the Representation as defined in subclause 5.10.3.

Generally, the Event Stream timing model follows the timing model of a media Representation in a Period.

DASH-specific events are defined in subclause 5.10.4.

5.10.2 MPD Events

5.10.2.1 Overview

Events may be signalled in the MPD. A sequence of events assigned to the media presentation time may be provided in the MPD on Period level. Events of the same type are summarized in an Event Stream that is specified by an **EventStream** element in a **Period** element. Events shall terminate at the end of a

Period even if the start time is after the Period boundary or duration of the event extends beyond the Period boundary.

The **EventStream** element is structured in a similar way as the descriptor defined in subclause 5.8, namely it contains a `@schemeIdUri` attribute that provides a URI to identify the scheme and an optional attribute `@value`. The semantics of the element are specific to the scheme employed. The URI identifying the scheme may be a URN or a URL.

A Period shall contain at most one **EventStream** element with the same value of the `@schemeIdUri` attribute and the value of the `@value` attribute, i.e. all Events of one type shall be clustered in one Event Stream.

As Event Streams contain timed events, also a time scale attribute `@timescale` is provided to assign events to a specific media presentation time within the Period. The timed events themselves are described by the **Event** element.

This document does not provide any specific information on how to use Event Streams. It is up to the application that employs DASH formats to instantiate the description elements with appropriate scheme information. However, this document defines some specific schemes in subclause 5.10.4.

NOTE A DASH application that uses one of these elements defines a Scheme Identifier in the form of a URI and then defines the value space for the element when that Scheme Identifier is used. The Scheme Identifier appears in the `@schemeIdUri` attribute.

The semantics of the attributes within the **EventStream** element are provided in subclause 5.10.2.2, Table 38 and the semantics of the attributes within the **Event** element are provided in subclause 5.10.2.2, Table 39. The XML syntax of **EventStream** and **Event** element is provided in subclause 5.10.2.3.

5.10.2.2 Semantics

Table 38 — Event Stream Semantics

Element or Attribute Name	Use	Description
EventStream		specifies event Stream
<code>@xlink:href</code>	O	specifies a reference to an external EventStream element
<code>@xlink:actuate</code>	OD default: onRequest	specifies the processing instructions, which can be either "onLoad" or "onRequest". This attribute shall not be present if the <code>@xlink:href</code> attribute is not present.
<code>@schemeIdUri</code>	M	identifies the message scheme. The string may use URN or URL syntax. When a URL is used, it is recommended to also contain a month-date in the form mmyyyy; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date. A URL may resolve to an Internet location, and a location that does resolve may store a specification of the message scheme.
<code>@value</code>	O	specifies the value for the event stream element. The value space and semantics must be defined by the owners of the scheme identified in the <code>@schemeIdUri</code> attribute.

Element or Attribute Name	Use	Description
@timescale	0	<p>specifies the timescale in units per seconds to be used for the derivation of different real-time duration values in the Event elements.</p> <p>If not present on any level, it shall be set to 1.</p>
@presentationTimeOffset	OD Default: 0	<p>specifies the presentation time offset of this Event Stream that aligns with the start of the Period. Any Event contained in this Event Stream is mapped to the Period timeline by using the Event presentation time subtracted by the value of the presentation time offset.</p> <p>This adjustment shall not be applied to Inband event message streams..</p> <p>The value of the presentation time offset in seconds is the division of the value of this attribute and the value of the @timescale attribute.</p>
Event	0 ... N	<p>specifies one event. For details see Table 39.</p> <p>Events in Event Streams shall be ordered such that their presentation time is non-decreasing.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

Table 39 — Event Semantics

Element or Attribute Name	Use	Description
Event		<p>specifies an Event and contains the message of the event. The content of this element depends on the event scheme. The contents shall be either:</p> <ul style="list-style-type: none"> • A string, optionally encoded as specified by @contentEncoding • XML content using elements external to the MPD namespace <p>For new event schemes string content should be used, making use of Base 64 encoding if needed.</p> <p>NOTE The schema allows “mixed” content within this element however only string data or XML elements are permitted by the above options, not a combination.</p>

Element or Attribute Name	Use	Description
@presentationTime	OD default: 0	<p>specifies the presentation time of the event relative to the start of the Period taking into account the @presentationTimeOffset of the Event Stream, if present.</p> <p>The value of the presentation time in seconds is the division of the value of this attribute and the value of the @timescale attribute.</p> <p>If not present, the value of the presentation time is 0.</p>
@duration	0	<p>specifies the presentation duration of the Event.</p> <p>The value of the duration in seconds is the division of the value of this attribute and the value of the @timescale attribute.</p> <p>The interpretation of the value of this attribute is defined by the scheme owner.</p> <p>If not present, the value of the duration is unknown.</p>
@id	0	<p>specifies an identifier for this instance of the event. Events with equivalent content and attribute values in the Event element shall have the same value for this attribute.</p> <p>The scope of the @id for each Event is with the same @schemeIdURI and @value pair.</p>
@contentEncoding	0	<p>specifies whether the information in the body and the information in the @messageData is encoded.</p> <p>If present, the following value is possible:</p> <ul style="list-style-type: none"> base64 the content is encoded as described in IETF RFC 4648 prior to adding it to the field. <p>If this attribute is present, the DASH Client is expected to decode the message data and only provide the decoded message to the application.</p>
@messageData	0	<p>specifies the value for the event stream element. The value space and semantics must be defined by the owners of the scheme identified in the @schemeIdUri attribute.</p> <p>NOTE the use of the message data is discouraged by content authors, it is only maintained for the purpose of backward-compatibility. Including the message in the Event element is recommended in preference to using this attribute. This attribute is expected to be deprecated in the future editions of this document.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.10.2.3 XML-Syntax

```

<xs:complexType name="EventStreamType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Event Stream
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Event" type="EventType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="xlink:href"/>
  <xs:attribute ref="xlink:actuate" default="onRequest"/>
  <xs:attribute ref="xlink:type" fixed="simple"/>
  <xs:attribute ref="xlink:show" fixed="embed"/>
  <xs:attribute name="schemeIdUri" type="xs:anyURI" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
  <xs:attribute name="timescale" type="xs:unsignedInt"/>
  <xs:attribute name="presentationTimeOffset" type="xs:unsignedLong" default="0"/>
</xs:complexType>
<xs:complexType name="EventType" mixed="true">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Event
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="presentationTime" type="xs:unsignedLong" default="0"/>
  <xs:attribute name="duration" type="xs:unsignedLong"/>
  <xs:attribute name="id" type="xs:unsignedInt"/>
  <xs:attribute name="contentEncoding" type="ContentEncodingType"/>
  <xs:attribute name="messageData" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        Deprecated in favor of carrying the message information in the
        value space of the event
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:simpleType name="ContentEncodingType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Event Coding
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="base64"/>
  </xs:restriction>
</xs:simpleType>

```

5.10.3 Inband Event Signalling

5.10.3.1 Overview

Event streams may be multiplexed with Representations by adding the event messages as part of the Segments. The event streams may be present in selected Representations, in one or several selected Adaptation Sets only or in all Representations. For example, one possible configuration is one where only the audio Adaptation Sets may contain inband events.

In order to identify the Representations that carry the event stream, the presence of Events shall be signalled in the MPD as defined in subclause 5.10.3.2.

If more than one Representation carries event streams with the same `@schemeIdUri` and the same `@value`, the streams shall be semantically equivalent, i.e. processing one Representation is sufficient.

The format of the box to signal events in the media stream is provided in subclause 5.10.3.3.

5.10.3.2 MPD Signalling

An inband event stream may be present in a Representation. If it is expected to be processed by the DASH Client it is indicated by an **InbandEventStream** element at the Adaptation Set or Representation level. The **InbandEventStream** type is defined in 5.10.2, Table 38 based on the **EventStream** semantics, but the following restrictions apply:

- The `@timescale` attribute may be absent.
- The `@presentationTimeOffset` attribute shall be absent.
- The **InbandEventStream** element shall not contain any MPD Events, i.e. the **Event** element shall not be included in any **InbandEventStream** element.

One Representation may contain multiple inband Event streams, each indicated by a separate **InbandEventStream** element.

Inband event messages may be present in Representations but a corresponding **InbandEventStream** element (with the identical `@schemeIdUri` and `@value` pair) is not present in MPD. Such inband event messages are expected to be ignored by the DASH Client. A DASH Client is only expected to process inband event messages if a corresponding **InbandEventStream** element is present in the MPD.

5.10.3.3 Event message box

5.10.3.3.1 General

The Event Message box (`'emsg'`) provides signalling for generic events related to the media presentation time. The same semantics as for an event defined in the MPD specified in subclause 5.10.2 applies.

The Event Message box (`'emsg'`) can carry signalling specific to the DASH operations. The event scheme identifier and the events for this are defined in subclause 5.10.4.

A Media Segment if based on the ISO BMFF container may contain one or more event message (`'emsg'`) boxes. If present, any `'emsg'` box shall be placed as follows:

- It may be placed before the first `'moof'` box of the segment
- It may be placed in between any `'mdat'` and `'moof'` box. In this case an equivalent `'emsg'` with the same `id` value shall be present before the first `'moof'` box of any Segment.

NOTE These placement options allow `emsg` boxes to be placed at the start of segments, or mid segment, whilst allowing clients to choose to only process `emsg` boxes which occur at the start of segments.

The carriage of event messages in MPEG-2 TS based segments is described in subclause 5.10.3.3.5.

Event message boxes with scheme identifier and value pairs that are not defined in the MPD should not be present. If a DASH Client detects an event message box with a scheme that is not defined in MPD, the client is expected to ignore it.

5.10.3.3.2 Definition

Box Type: `'emsg'`

Container: Segment

Mandatory: No

Quantity: Zero or more

5.10.3.3.3 Syntax

```
aligned(8) class DASHEventMessageBox extends FullBox('emsg', version,
flags=0){
    if (version==0) {
        string          scheme_id_uri;
        string          value;
        unsigned int(32) timescale;
        unsigned int(32) presentation_time_delta;
        unsigned int(32) event_duration;
        unsigned int(32) id;
    } else if (version==1) {
        unsigned int(32) timescale;
        unsigned int(64) presentation_time;
        unsigned int(32) event_duration;
        unsigned int(32) id;
        string          scheme_id_uri;
        string          value;
    }
    unsigned int(8)    message_data[];
}
```

5.10.3.3.4 Semantics

- `scheme_id_uri`: is a null-terminated ('C') string in UTF-8 characters that identifies the message scheme. The semantics and syntax of the `message_data[]` are defined by the owner of the scheme identified. The string may use URN or URL syntax. When a URL is used, it is recommended to also contain a month-date in the form `mmyyyy`; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date. A URL may resolve to an Internet location, and a location that does resolve may store a specification of the message scheme.
- `value`: is a null-terminated ('C') string in UTF-8 characters that specifies the value for the event. The value space and semantics must be defined by the owners of the scheme identified in the `scheme_id_uri` field.
- `timescale` provides the timescale, in ticks per second, for the `event_duration` and `presentation_time_delta` or `presentation_time` fields. The value should be identical to the timescale of a track contained in the carrying Segment. Furthermore, the value should be identical for all events in one Event Stream.
- `presentation_time_delta` provides the Media Presentation time delta of the media presentation time of the event and the earliest presentation time in this segment. If the segment index is present, then the earliest presentation time is determined by the field `earliest_presentation_time` of the first 'sidx' box. If the segment index is not present, the earliest presentation time is determined as the earliest presentation time of any access unit in the media segment. The timescale is provided in the `timescale` field.
- `presentation_time` provides the Media Presentation time of the event measured on the Movie timeline, in the timescale provided in the `timescale` field, and adjusted by **InbandEventStream**@`presentationTimeOffset`, in the time scale provided by **InbandEventStream**@`timescale`; the value shall not be less than the earliest presentation time of the carrying Segment.

- `event_duration` provides the duration of event in media presentation time. The timescale is indicated in the `timescale` field. The value `0xFFFFFFFF` indicates an unknown duration. The interpretation of this value must be defined by the owner of the event scheme.
- `id`: a field identifying this instance of the message. The scope of this identifier for each event is with the same `scheme_id_uri` and `value` pair. Messages with the same `id` within the scope of the same `scheme_id_uri` and `value` pair are equivalent, i.e. processing of any one event message box with the same `id` is sufficient.
- `message_data`: body of the message, which fills the remainder of the message box. This may be empty depending on the above information. The syntax and semantics of this field must be defined by the owner of the scheme identified in the `scheme_id_uri` field.

5.10.3.3.5 Carriage of the Event Message Box in MPEG-2 TS

A Media Segment if encapsulated in MPEG-2 Transport Stream may contain one or more event message ('`emsg`') boxes encapsulated into transport stream packets.

Transport stream packets carrying the '`emsg`' box shall use a reserved fixed PID value of `0x0004`.

The transport stream packet carrying the start of the '`emsg`' box shall have the `payload_unit_start_indicator` field set to '1', and the packet payload will start with the '`emsg`' box. The complete `Box.type` field shall be present in this first packet, and the payload size shall be at least 8 bytes.

The continuation of box data occupies the following transport stream packets from the same PID. The last packet carrying the end of the box is padded using adaptation field stuffing bytes.

A segment shall contain only complete boxes. If `@bitstreamSwitching` is set, and subsegments are used, a subsegment shall contain only complete '`emsg`' boxes.

For any packet with PID value of `0x0004`, the value of the `transport_scrambling_control` field shall be set to '00'.

5.10.3.3.6 Inband Event Alignment

If `AdaptationSet.InbandEventStream` element is present and `AdaptationSet@segmentAlignment` attribute is true, event message boxes in non-overlapping Segments shall be *aligned*. Let $S_{R1}(T)$ be a segment of Representation R_1 with earliest presentation time T , and let the Adaptation Set contain N representations. If $S_{R1}(T)$ contains one or more Event Message ('`emsg`') boxes, identical '`emsg`' boxes shall be contained in each of the non-overlapping Segments $S_{R2}(T) \dots S_{RN}(T)$.

NOTE 1 As a consequence, under the above constraints, all Representations in the Adaptation Set contain events.

NOTE 2 If Segments are non-overlapping, but their EPT differ, alignment described above still applies, and Event Message boxes are in the beginning of both Segments.

5.10.4 DASH-specific events

5.10.4.1 Overview

DASH specific events that are of relevance for the DASH Client are signalled in the MPD. The URN "`urn:mpeg:dash:event:2012`" is defined to identify the event scheme defined in Table 40.

For events using this schema, the 'emsg'.message_data[] field contains the DASHEvent structure defined below:

```
aligned(8) struct DASHEvent
{
    string publish_time; // MPD@publishTime
    if ( 'emsg'.value == 2 )
    {
        string mpd_patch; // MPD patch as defined in subclause 5.15
    }
    if ( 'emsg'.value == 3 )
    {
        string mpd; // full MPD, as defined in subclause 5.10.4.4
    }
}
```

Table 40 — InbandEventStream@value attribute for scheme with a value "urn:mpeg:dash:event:2012"

@value	Description
1	indicates that MPD validity expiration events as defined in subclause 5.10.4.2 are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause 5.10.4.2 at least in the last segment with earliest presentation time smaller than the event time.
2	indicates that MPD validity expiration events as defined in subclause 5.10.4.2 are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause 5.10.4.2 at least in the last segment with earliest presentation time smaller than the event time. In addition, the message includes an MPD Patch document as defined in subclause 5.15. DASHEvent.mpd field within the message_data field.
3	indicates that MPD validity expiration events as defined in subclause 5.10.4.4 are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause 5.10.4.2 at least in the last segment with earliest presentation time smaller than the event time. In addition, the message includes a <i>complete</i> MPD as defined in subclause 5.10.4.4 in DASHEvent.mpd field within the message_data field.

NOTE Additional values for InbandEventStream@value when @schemeIDURI is urn:mpeg:dash:event:2012 are reserved for ISO/IEC.

5.10.4.2 MPD validity expiration

MPD validity expiration events provide the ability to signal to the client that the MPD with a specific publish time can only be used up to a certain media presentation time.

MPD validity expiration shall be signalled for all updates causing an extension of the timeline, except for the following ones:

- The value of the **MPD@minimumUpdatePeriod** is changed,
- The value of a **SegmentTimeline.S@r** has changed,
- A new **SegmentTimeline.S** element is added.

If the `scheme_id_uri` is set to "urn:mpeg:dash:event:2012" and the value is set to 1, then the fields in the event message box shall document the following:

- the `DASHEvent.publish_time` field contains the publish time of an MPD, i.e. the value of the `MPD@publishTime`.
- The media presentation time beyond the event time (indicated time by `presentation_time_delta` in v0 and the `presentation_time` in v1) is correctly described only by MPDs with publish time greater than indicated value in the `message_data` field.
- the event duration expresses the remaining duration of Media Presentation from the event time. If the event duration is 0, the Media Presentation ends at the event time. If 0xFFFF, the media presentation duration is unknown.

This implies that clients attempting to process the Media Presentation at the event time or later are expected to operate on an MPD with a publish time that is later than the indicated publish time in this box.

Event boxes in different segments may have identical `id` fields, but different values for the `presentation_time` or `presentation_time_delta` if the earliest presentation time is different across segments.

Figure 5 shows an example for MPD validity expiration method. An MPD signals the presence of the scheme in one or several Representations. Once a new MPD gets available, that adds new information not present in the MPD with `@publishTime="2012-11-01T09:06:31.6"`, the expiration time of the current MPD is added to the segment by using the `emsg` box. The information may be present in multiple segments.

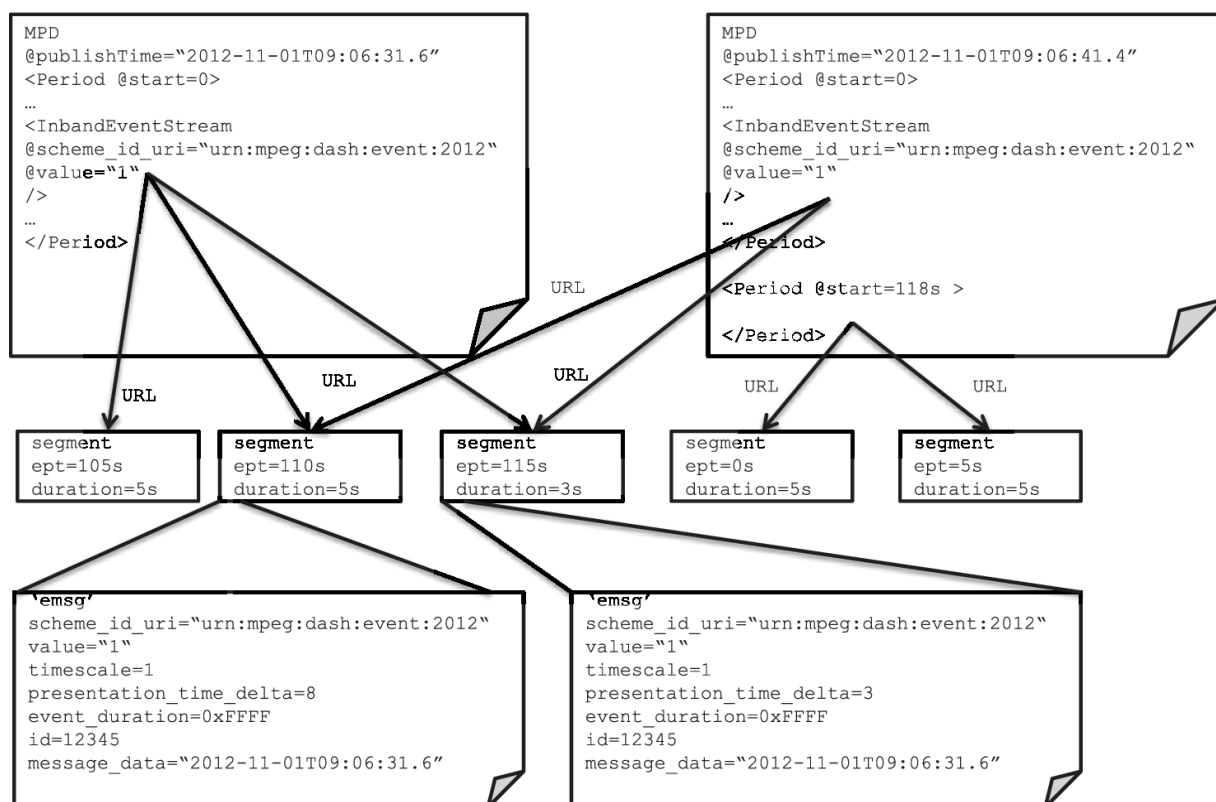


Figure 5 — Example for MPD validity expiration to signal new Period

5.10.4.3 MPD Patch event

For DASH events with value 2, an MPD patch shall be included in the `DASHEvent` structure, immediately following the `publish_time` field. The payload of this message shall be a valid MPD patch as defined in subclause 5.15.

The result of the patch application shall be parse-tree identical before any `xlink` resolution to the MPD with `publish_time` would have been retrieved at event time. For details on the processing model refer to subclause 5.15.4.

5.10.4.4 MPD Update Event

For DASH events with value 3, a complete MPD shall be included in the in the `DASHEvent` structure, immediately following the `publish_time` field. The content of the `mpd` field shall be the MPD.

5.10.4.5 DASH Callback Event

5.10.4.5.1 General

DASH Callback events are indications in the content that it is expected by a DASH Client to issue an HTTP GET request to a given URL and ignore the HTTP response. These event schemes are identified by the URN "urn:mpeg:dash:event:callback:2015".

A content author may use such an event for tracking play-back of specific content on a server that is not included in the media path.

The Status-Code field of the HTTP response should be 2xx; however, the client is expected to entirely ignore the response.

NOTE 1 HTTP GET (as opposed to HEAD) is used in alignment with IAB VAST^[18].

The message body of the HTTP response should be as small as possible or absent.

NOTE 2 The system adopting this functionality is expected to define appropriate means for secure handling of this feature.

5.10.4.5.2 Inband event

Table 41 defines the message data and the expected actions for different `@value` values when the DASH callback event is signalled as an Inband Event.

Table 41 — Message data and expected actions for DASH call back inband event

value	message_data[]	Action
1	Valid HTTP/HTTPS URL	<p>An HTTP GET request is expected to be issued to a URL contained in <code>message_data[]</code>.</p> <p>The URL shall be a NULL-terminated string.</p> <p>HTTP response shall either not be provided or be provided such that it can be discarded.</p>

5.10.4.5.3 MPD event

Table 42 defines the relevant parameters for a call back event signalled in the MPD.

Table 42 — Relevant parameters for a call back event signalled in the MPD

Attribute	Value
EventStream @schemeIdUri	"urn:mpeg:dash:event:callback:2015"
EventStream @value	1
Event @messageData	HTTP-URL HTTP response is expected to be discarded without parsing.

5.10.4.6 Presentation Termination Event

DASH Presentation Termination events are indications that the currently playing Media Presentation is ending at a time earlier than expected from the current MPD. This event can be either an inband event or an MPD event. These events are identified by the URN "urn:mpeg:dash:event:ttfn:2016".

NOTE The primary use case for this feature is when the client does not expect to get an MPD update.

Three values are defined according to Table 43.

Table 43 — interpretation of value field of a Presentation Termination event

value	Description
0	indicates the end of presentation, including these of chained-to and fallback MPDs
1	Indicates end of presentation. Presentation described in the chained-to MPD (if the latter is present) is still valid. NOTE If the value of <code>earliestTimeToResolve</code> in the MPD chaining descriptor is later than the time derived from <code>'emsg'.presentation_time_delta</code> , this is considered an error condition and triggers switch to the fallback presentation, if possible.
2	Indicates end of presentation due to an unspecified error condition. Fallback presentation is still valid; hence the client is expected to switch to it.

5.11 MPD Chaining

5.11.1 General

MPD chaining provides a mechanism to indicate that, at the end of one Media Presentation, a new Media Presentation starts. The end may be a regular end, or an early termination due to an error condition. In order to enable this mechanism, the “chained-from” MPD may include an Essential or Supplemental descriptor which points to the “chained-to” MPD location.

Two cases are differentiated, a regular chaining operation at the end of the chained-from MPD in 5.11.2 and a case for which the chained-to MPD is only played in case the “chained-from” MPD is terminated early due to error conditions as defined in subclause 5.11.3 and serves as a fallback.

5.11.2 Regular Chaining

Regular chaining refers to the case that a Media Presentation is played until the end, and once the Media Presentation is finished, a new chained Media Presentation is played instantaneously. A client receiving the chained-from MPD is expected to play the chained-to MPD right after the chained-from one. Each MPD has its own independent media timeline, but the DASH Client is expected to continue the presentation to create a sequential presentation.

The chained-from MPD may be of type static or dynamic. The chained-to MPD may also be of type static and type dynamic.

MPD chaining can for example be used for pre-roll ads or creating a sequence of programs using multiple MPDs. In this case, the chained-from MPD may be of type static, whereas the chained-to MPD may be of type dynamic. In this case, the client is expected to join the dynamic MPD at the live edge, or if an anchor is presented as defined in C.4, at the indicated time in the anchor. The chained-from MPD may contain an `@xlink:href` Period for personalization of the ad, whereas the chained-to MPD is common.

MPD chaining is signalled by using an Essential or Supplementary Descriptor on MPD level with `@schemeIdUri` set to "urn:mpeg:dash:mpd-chaining:2016". Each MPD may contain at most one descriptor for MPD chaining. The `@value` of this descriptor shall be composed of the whitespace-separated parameters according to Table 44. If `@value` only contains the first parameter, no whitespace is needed.

Table 44 — Semantics of `value` for MPD chaining

Value parameters	Use	Description
<code>url</code>	M	<p>specifies the location of manifest to be played manifest (chained-to MPD) after this MPD.</p> <p>The client is expected, after playback of the entire MPD, to download the chained-to MPD from the location defined by this parameter and play chained-to MPD without any delays (similar to play back of a new period). In the case the <code>url</code> does not return a valid MPD or any HTTP error response, the client is expected to treat it as an invalid MPD.</p> <p>If the <code>url</code> has any anchor, it anchors the chained-to MPD according to C.4.</p>
<code>earliestTimeToResolve</code>	O	<p>recommends the earliest media presentation time in the chained-from MPD as a difference to the end of the Media Presentation in milliseconds to fetch the chained-to MPD.</p> <p>If not present, no recommendation is provided.</p>

5.11.3 Fallback Chaining

Fallback chaining refers to the case that a Media Presentation is played, but once an error condition occurs, a new chained Media Presentation may be played. Fallback chaining descriptor allows the author to provide alternative content the client can switch to in case of error condition preventing it from continuing normal playback.

An error condition triggering such processing is one of the following:

- Failure to of an MPD.
- Failure to download a media (sub)segment with a given value of earliest presentation time, after exhausting all possibilities of retrieving it (such as trying all BaseURLs and all Representations within an Adaptation Set), if the value of **MPD**@type is "static".
- Failure to retrieve a chained-to MPD.
- Unspecified error condition triggered via Presentation Termination event with value for fallback.

Fallback presentation descriptor provides a URL to an alternative MPD which should be played out once one of the aforementioned error conditions makes it impossible to continue playback of the current Media Presentation.

Fallback shall be signalled by a Supplementary Descriptor on MPD level with `@schemeIdUri` set to "urn:mpeg:dash:fallback:2016". Each MPD may contain at most one Fallback Presentation descriptor.

The `@value` of this descriptor shall be one URL or a whitespace-separated list of URLs of the "chained-to" MPD. If multiple URLs are provided, the content author expresses the preferences of using one of those by the order with the first one having the highest preference.

5.12 Producer Reference Time

5.12.1 General

The Producer Reference Time supplies media correlation between media timestamps and wall clock production time. This information permits the following, among others:

- provides media clients with information to enable consumption and production to proceed at equivalent rates, thus avoiding possible buffer overflow or underflow.
- enables measuring and potentially controlling the latency between the production of the media time and the payout.

The definition follows the Producer Reference Time ('prft') as defined in ISO/IEC 14496-12.

The information may be provided inband as part of the Segments in the ('prft'), in the MPD or both.

The semantics of the attributes and elements for producer reference time are provided in subclause 5.12.2, Table 45. The XML syntax of the Producer Reference Time is provided in subclause 5.12.3.

5.12.2 Semantics

Table 45 — Semantics of `ProducerReferenceTime` element

Element or Attribute Name	Use	Description
<code>ProducerReferenceTime</code>		specifies the producer reference time for the associated Representations.
<code>@id</code>	M	Identifies the Producer Reference time documented in the MPD.
<code>@inband</code>	OD default: false	signals if every segment or subsegment contains a 'prft' of the type below. If set to true, every segment or subsegment shall contain the 'prft' as defined in ISO/IEC 14496-12 and the flags set according to the <code>@type</code> attribute of this element.
<code>@type</code>	OD default: encoder	specifies the type of the Producer Reference Time from the following list: <ul style="list-style-type: none"> — <code>encoder</code> provides a reference when the media time was input to an encoder following the exact definition in ISO/IEC 14496-12, 'prft' for flags set to 0.

		<ul style="list-style-type: none"> — captured provides a reference when the media time was captured following the exact definition in ISO/IEC 14496-12, 'prft' for both flag 8 and flag 16 being set. — application provides a reference of the media time related to wall-clock time based on an application defined relation. In this case following ISO/IEC 14496-12 'prft' flag 16 shall be set and flag 8 shall be unset.
@applicationScheme	CM	<p>specifies the application scheme to which the time conforms if @type is set to application.</p> <p>If the @type is set to application, this attribute should be present.</p> <p>If the @type is set other than application, this attribute shall not be present.</p>
@wallClockTime	M	<p>specifies a wall-clock time in the following format</p> <ul style="list-style-type: none"> — if UTCTiming element is absent, the NTP format associated to @media as defined in ISO/IEC 14496-12 'prft' for ntp_timestamp — if UTCTiming element is present, the format is identical to the format as defined in the UTC Timing scheme. <p>NOTE If the MPD Generator extracts the information from the producer reference time box, the value needs to be converted to the correct scheme.</p>
@presentationTime	M	<p>specifies a presentation time in timescale of the Representation that relates to the value of the @wallClockTime.</p> <p>NOTE If the data is extracted from the producer reference time box, then this value is derived by the media_time field.</p>
UTCTiming	0 ... 1	<p>If present, then the wall-clock times provided in this context are synchronized with the timing anchor provided in this descriptor. The same UTC Timing descriptor shall also be present in the MPD.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.12.3 XML Syntax

```
<xs:simpleType name="ProducerReferenceTimeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="encoder"/>
    <xs:enumeration value="captured"/>
    <xs:enumeration value="application"/>
  </xs:restriction>
</xs:simpleType>
```

5.13 Leap seconds

5.13.1 Overview

Information on the occurrence of leap seconds and their relationship with **MPD@availabilityStartTime** can be expressed using the **LeapSecondInformation** element.

This information allows a client to perform time calculations accurately without an external source of information on the timing of past and future leap seconds. It also allows a service provider to express any correction for leap seconds that has already been made.

With this information, a client may perform timing calculations by assuming a constant day length of 86400 seconds if it first applies the relevant offset from the **LeapSecondInformation** element to the value of the **MPD@availabilityStartTime** attribute. The client can then correctly play a Media Presentation for which **MPD@type** is 'dynamic' when leap seconds have occurred since the time of **MPD@availabilityStartTime** and even when leap seconds occur during playback.

The offset specified in **@availabilityStartLeapOffset** applies if the current time is before **@nextLeapChangeTime**. **@nextAvailabilityStartLeapOffset** applies if the current time is equal to or after **@nextLeapChangeTime**, but its use may need to be delayed if the client's internal wall clock has not yet processed the leap second.

NOTE 1 If a particular client's wall clock is synchronized solely by means of a one-shot protocol, e.g. using the scheme `urn:mpeg:dash:utc:http-xsdate:2014` defined in subclause 5.8.5.7, the value of **@availabilityStartLeapOffset** remains appropriate for that client until its clock is next synchronized after the leap second has occurred.

NOTE 2 Leap seconds can occur at any of four possible times each year and can add or remove a second from the UTC timeline. Whilst leap seconds always occur immediately before midnight UTC, they can occur during the day in other time zones. Even within the Greenwich Mean Time (GMT) zone, the most common time for a leap second has been 23:59:60 on New Year's Eve, when many people may be watching live video streams. It is therefore important that leap seconds are handled correctly.

Some operating systems allow leap seconds to be handled by "smearing" the leap second over a longer period of time. Care should be taken to ensure that availability times in a published MPD are not affected by this.

The semantics of the attributes within the **LeapSecondInformation** element are provided in subclause 5.13.2, Table 46. The XML syntax of the **LeapSecondInformation** element is provided in subclause 5.13.3.

5.13.2 Semantics

Table 46 — Leap Second Information Semantics

Element or Attribute Name	Use	Description
LeapSecondInformation		specifies leap second information affecting MPD timing calculations
@availabilityStartLeapOffset	M	<p>specifies the number of seconds applying at the time of MPD publication that a client would need to subtract from MPD@availabilityStartTime in order to perform timing calculations without further consideration of leap seconds.</p> <p>If a leap second correction has already been applied, the LeapSecondInformation element should still be present but this attribute should be set to 0.</p>
@nextAvailabilityStartLeapOffset	0	<p>specifies the number of seconds that will apply from the time of the next leap second (indicated by the @nextLeapChangeTime) that a client would need to subtract from MPD@availabilityStartTime in order to perform timing calculations without further consideration of leap seconds.</p> <p>If the timing of the next leap second is unknown, this attribute shall be omitted.</p>
@nextLeapChangeTime	0	<p>specifies the UTC time at which a leap second will occur. Before this time, @availabilityStartLeapOffset applies. On or after this time, @nextAvailabilityStartLeapOffset applies for timing calculations made against a wall clock that has processed the leap second.</p> <p>If the timing of the next leap second is unknown, this attribute shall be omitted.</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

5.13.3 XML-Syntax

```

<xs:complexType name="LeapSecondInformationType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Leap Second Information
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="availabilityStartLeapOffset" type="xs:integer" use="required"/>
  <xs:attribute name="nextAvailabilityStartLeapOffset" type="xs:integer"/>
  <xs:attribute name="nextLeapChangeTime" type="xs:dateTime"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.13.4 Leap second information updates

Streams which continue to run continuously between leap second changes will need to update the MPD to change the **LeapSecondInformation** element. At a time after @nextLeapChangeTime, and preferably long enough to be able to expect all leap second aware clients to have resynchronized their clocks, the value of @nextAvailabilityStartLeapOffset can be moved into @availabilityStartLeapOffset, and the attributes @nextAvailabilityStartLeapOffset and @nextLeapChangeTime removed. Then, at least **MPD@minimumUpdatePeriod** before the next leap second change, the LeapSecondInformation element can be updated to include the new values for @nextAvailabilityStartLeapOffset and @nextLeapChangeTime. It is possible to make both of these changes at the same time if the new values are known when @availabilityStartLeapOffset is being updated.

NOTE Updating the MPD in this manner is normally be seamless for clients that process the LeapSecondInformation element and for those that do not. However, if the stream continues across many leap seconds, clients that do not process the **LeapSecondInformation** element have a cumulative error in their calculation of the live edge when they join the stream.

5.14 Content Popularity Rate

5.14.1 General

Content Popularity Rate is a method for signalling information on which content items amongst the different Preselections and Adaptation Sets are expected to be more popular than others. Example use cases for different content streams in a Media Presentation include:

- content streams from different viewpoints: in this case, an Adaptation Set containing recommended viewpoint is expected to be more popular than other Adaptation Sets.
- sub-picture streams in a 360 video: in this case, a set of Adaptation Sets containing the sub-picture streams that compose “recommended viewport” can be signalled as more popular than others.

Content Popularity Rate information is expected to be provided by the content provider based on the author’s intent, expected consumption rate by users, or actual viewing statistics that could have been measured by using DASH metrics, such as Play list as defined in Annex D.4.6.

If provided, the information can be used for data prefetching by clients or network elements, e.g. CDN servers. It can also be used for example to deliver the data that is considered most relevant on a preferred data link, for example on multicast or broadcast, whereas less popular data is only provided in unicast. Expected-to-be-popular data can also be prefetched in higher quality in order to ensure that such data is available in good quality.

To indicate the popularity of a Pre-selection or Adaptation Set, the **ContentPopularityRate** element may be present within that element. The **ContentPopularityRate** element contains one or more PR elements giving the rating for different parts of the content. The semantics of the **ContentPopularityRate** element are provided in 5.14.2 Table 47, the syntax is provided in 5.14.3.

5.14.2 Semantics

Table 47 — Semantics of `ContentPopularityRate` element

Element or Attribute Name	Use	Description
ContentPopularityRate		specifies the Content Popularity Rate information.
@source	M	<p>indicates the source of rating by this element. The value can be one of the following:</p> <ul style="list-style-type: none"> — <code>content</code> (when the value is specified by the content author or content provider). — <code>statistics</code> (when the value is specified based on viewing statistics). — <code>other</code>. <p>When “other” is used, <code>@source_description</code> should be provided.</p>
@source_description	0	UTF-8 string that provides a textual description of the source of rating.
PR	1..N	specifies start time and number of segments for a contiguous sequence of segments assigned identical Popularity Rate value.
@popularityRate	M	indicates the relative Popularity Rate of the segments of containing entity (i.e. the Preselection or Adaptation Set containing this ContentPopularityRate element), within the same Media Presentation. No unit. The value shall be in the range of 1 to 100. A greater value means higher likelihood of the series of (Sub)Segments being consumed/requested.
@start	0	<p>specifies the first Segment from which <code>@popularityRate</code> value of this PR element is relevant.</p> <p>If the addressing scheme for the containing entity is using Segment template with <code>\$Number\$</code>, then <code>@popularityRate</code> applies from the earliest presentation time of the Segment with the segment number specified by this attribute. For other addressing schemes, the value of <code>@popularityRate</code> applies from the Segment which contains the media sample whose presentation time specified by this attribute in unit of <code>@timescale</code> on Representation level.</p> <p>If not present, then it shall be assumed that <code>@popularityRate</code> applies from:</p> <ul style="list-style-type: none"> • the first Segment of the containing Period, for the first PR element

		<ul style="list-style-type: none"> the Segment after the last Segment of the previous PR element, for other PR elements.
@r	OD default: 0	specifies the number of segments after the segment indicated by @start which are included in this PR entry (i.e. the repeat count). A negative value indicates that the series continues until the segment before the first segment of the next PR element, or if there are no further PR elements until the end of the Period or the next MPD update.

Legend:

For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory.

For elements: <minOccurs>...<maxOccurs> (N=unbounded)

Elements are **bold**; attributes are non-bold and preceded with an @.

5.14.3 XML syntax

```

<xs:complexType name="ContentPopularityRateType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Content Popularity Rate
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="PR" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="popularityRate">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
              <xs:minInclusive value="1"/>
              <xs:maxInclusive value="100"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="start" type="xs:unsignedLong" use="optional"/>
        <xs:attribute name="r" type="xs:int" use="optional" default="0"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="source" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="content"/>
        <xs:enumeration value="statistics"/>
        <xs:enumeration value="other"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="source_description" type="xs:string"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

5.15 MPD patch framework

5.15.1 Overview

In the case of a live linear channel and dynamic MPD, MPD updates can happen frequently and create a considerable bandwidth and processing overhead. In the vast majority of cases, a small minority of elements in the MPD is added or removed. For example, in case of relatively frequent MPD requests and use of **SegmentTimeline**, only at most a few **S** elements are added per adaptation set.

MPD patch framework allows transmission of only changed parts of the MPD as opposed to the complete MPD and uses the XML patch framework defined in RFC 5261.

The MPD patch framework consists of the following main functionalities:

- 1) The location where the MPD patch document can be accessed and the information associated with this information. This is specified in subclause 5.2.3 in the MPD as the element **MPD.PatchLocation**. For details refer to subclause 5.15.2.
- 2) An MPD patch document that includes the instructions to patch an MPD. For details refer to subclause 5.15.3.
- 3) The processing model to generate an equivalent new MPD from an MPD in memory and the MPD patch document as documented in subclause 5.15.4.
- 4) Recommended practices and guidelines for the DASH client to make use of the MPD Patch functionality in subclause 5.15.5.

5.15.2 MPD Patch Location

An MPD may contain one or several **PatchLocation** elements to provide a location for an MPD patch document that may apply to the MPD including this element in order to generate an updated MPD.

The semantics of the MPD patch location element are provided in Table 48 and the XML syntax is provided in Table 49.

Table 48 — Semantics of **PatchLocation element in MPD**

Element or Attribute Name	Use	Description
PatchLocation		specifies a location at which the MPD patch is available. The referenced document shall conform to an MPD patch document as defined in subclause 5.15.3.
@ttl	0	specifies the time period duration (in seconds) starting from MPD@publishTime until the MPD patch document is at least available at the indicated location above. For details refer to the processing model in subclause 5.14.4. If not present, the value is unknown.
Key For attributes: M=mandatory, O=optional Elements are bold ; attributes are non-bold and preceded with an @.		

Table 49 — XML Syntax of PatchLocation element in MPD

```

<xs:complexType name="PatchLocationType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Patch Location Type
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attribute name="ttl" type="xs:double" use="optional"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

5.15.3 MPD patch document

5.15.3.1 General

The MPD Patch (MPP) is a document that contains required information to generate an equivalent new MPD from an MPD in hand and this information. The processing is provided in subclause 5.15.4. Typically, this processing is done by the DASH client, but the description is independent of the DASH client.

The MPP is an XML document that shall be formatted according to the XML schema provided in Annex B. Some context on the schema is provided in subclause 5.15.3.2.

The extension of the MPP XML schema (as provided in Annex B), in particular the addition of XML attributes or elements in the MPP namespace, is reserved to ISO/IEC.

The MPP shall be authored such that, after XML attributes or elements in the DASH namespace but not in the XML schema documented in Annex B are removed, the result is a valid XML document formatted according to that schema and that conforms to this document.

In addition, the MPP shall be authored such that, after XML attributes or elements in the other namespaces than the MPP namespace are removed, the result is a valid XML document formatted according to that schema and that conforms to this document.

Following XML rules, the MPP document shall contain exactly one **Patch** element as specified in subclause 5.15.3.2. The summary of the semantics of the attributes and elements within a **Patch** element are provided in Table 50, subclause 5.15.3.2. The XML syntax of the **Patch** element is provided in subclause 5.15.3.3 together with the schema. The **Patch** element may contain one or multiple **add**, **remove**, and **replace** elements which shall be as defined in IETF RFC 5261. The **add**, **remove**, and **replace** elements define operations to be applied to the MPD in hand that result in the creation of the fully updated MPD. The operations utilize XPath 1.0 selectors, restricted in complexity from their original W3C definition by IETF RFC 5261, to target elements and attributes within the MPD in hand. In addition, subclause 5.15.3.4 defines restrictions on the usage of the elements within selectors for targeting MPD elements.

The MIME type of the MPD document is defined in Annex C.

The encoding of the MPD shall be UTF-8 as defined in IETF RFC 3629. All data provided in extension namespaces shall be UTF-8 as defined in IETF RFC 3629. If binary data needs to be added, it shall be included in Base64 as described in IETF RFC 4648 within a UTF-8 encoded element with a proper name space or identifier, such that an XML parser knows how to process or ignore it.

The delivery of the MPP is outside the scope of this document. However, if the MPP is delivered over HTTP, then the MPP document may be transfer encoded for transport, as described in IETF RFC 7230.

Selected MPP examples are provided in Annex G.21.

5.15.3.2 Semantics

The MPD patch document contains a single **Patch** element, which contains multiple **add**, **remove**, and **replace** elements shall be as defined in IETF RFC 5261. The semantics of the **Patch** element are provided in Table 50 and the XML syntax is provided in subclause 5.15.3.3.

Table 50 — Semantics of Patch element in MPP

Element or Attribute Name	Use	Description
Patch	1	element that carries the patch data.
@mpdId	M	Specifies the identifier for the Media Presentation Description this patch applies to.
@publishTime	M	Specifies the wall-clock time when the MPD patch was generated and publish at the origin server.
@originalPublishTime	M	Specifies the wall-clock time used as a basis for generating information in the MPD patch. This patch only applies to the MPD with MPD @publishTime equal to this value, otherwise it is expected to be ignored by the DASH client
add	0..N	Specifies an <add> element as defined in RFC 5261, subclause 4.3. For restrictions, refer to subclause 5.15.3.4.
remove	0..N	Specifies an <remove> element as defined in RFC 5261, subclause 4.4. For restrictions, refer to subclause 5.15.3.4.
replace	0..N	Specifies an <replace> element as defined in RFC 5261, subclause 4.5. For restrictions, refer to subclause 5.15.3.4.
Key For attributes: M=mandatory, O=optional Elements are bold ; attributes are non-bold and preceded with an @.		

5.15.3.3 XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:patch="urn:ietf:params:xml:schema:patch-ops"
  xmlns="urn:mpeg:dash:schema:mpd-patch:2020" targetNamespace="urn:mpeg:dash:schema:mpd-patch:2020"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- Include patch operations from RFC5261 -->
  <xs:include schemaLocation="https://www.iana.org/assignments/xml-registry/schema/patch-ops.xsd"/>
  <xs:element name="Patch" type="PatchType"/>
  <!-- Patch -->
  <xs:complexType name="PatchType">
    <xs:choice minOccurs="1" maxOccurs="unbounded">
```

```

    <xs:element name="add" type="add"/>
    <xs:element name="remove" type="remove"/>
    <xs:element name="replace" type="replace"/>
    <xs:any namespace="##other" processContents="lax"/>
  </xs:choice>
  <xs:attribute name="mpdId" type="xs:string" use="required"/>
  <xs:attribute name="publishTime" type="xs:dateTime" use="required"/>
  <xs:attribute name="originalPublishTime" type="xs:dateTime" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
</xs:schema>

```

5.15.3.4 Element Addressing Restrictions

The following element addressing restrictions apply for DASH MPD elements:

- Elements with an @id attribute with uniqueness requirement among its siblings present shall only be identified by the value of this attribute. In particular, **Period**, **AdaptationSet**, **Representation**, and **SubRepresentation** shall be only addressed via their @id.
- **DescriptorType** elements with an @schemeIdUri attribute shall be identified by the value of this attribute, for example **SupplementalProperty**.
- **S** elements shall be addressed in one of the following ways:
 - @t or @n attribute of **S** elements if the said attribute is present in the **S** element.
 - by position.
- All other elements shall be addressed by position, e.g. `SegmentTemplate[1]`.

Examples:

```
/MPD/Period[@id="1"]/AdaptationSet[@id="1"]/SegmentTemplate[1]/Segment
Timeline[1]/S[@t="12345"]
```

```
/MPD/Period[@id="1"]/AdaptationSet[@id="1"]/SegmentTemplate[1]/Segment
Timeline[1]/S[42]@r
```

5.15.4 Processing Model

This processing model describes how to achieve an updated MPD from an in-memory MPD and the information in the MPD patch document.

The MPD derived through MPD patch application shall be identical in their canonical XML form as defined in W3C Canonical XML to the full MPD at the **MPD.Location**, if present, or the URL of the initial MPD with the same value of the **MPD@publishTime** attribute, prior to XLink resolution.

NOTE 1 An in-memory MPD is an MPD that is retained within either the temporary or persistent memory of a DASH client and is referenced by the DASH client to proceed with the playout of the media presentation.

To allow for DASH client implementation efficiency, the in-memory MPD need not be stored in full XML structure format, but it must be stored such that all DASH defined identifiers and element order of multiple events of the same type are preserved. See 5.15.3.3 for the XPath addressing modes permitted by this document.

The MPD patch document is guaranteed to be available between **MPD@publishTime** and **MPD@publishTime + PatchLocation@ttl**.

The patch operations shall be applied prior to Xlink dereferencing. For instance, if patch application results in a new remote Period, dereferencing shall occur after the patch.

NOTE 2 Patch application is not expected to affect periods previously resolved using XLink, unless the entity generating the patch is aware of the **Period@id** of the previously resolved period.

5.15.5 Recommended Client Operation

A DASH client is expected to carefully use the MPD patch information to avoid any mal-formed or wrong MPDs based on erroneous MPD Patch operation.

It is important to note that an MPD patch document is valid if and only if all of the following conditions are met given an MPD and an MPD patch:

- The **Patch@mpdId** value of the MPD patch is identical to the value of the **MPD@id** attribute of the in-memory MPD
- The value of the **Patch@originalPublishTime** is identical to the value of the **MPD@publishTime** attribute of the in-memory MPD
- The value of **Patch@publishTime** is greater than the value of the **MPD@publishTime** attribute of the in-memory MPD
- The patch contains a replace element modifying the value of **MPD@publishTime** to the value of **Patch@publishTime**

The DASH client is expected to check all of the above conditions.

If any of the above conditions does not hold,

- the patch application could produce a different MPD than the MPD with the same **MPD@id** and **MPD@publishTime**. This will lead to implementation-specific, fragile, and inconsistent operation of the client. In case of any doubt, the client is expected to request the full MPD, and ignore the patch.
- In case the request is made after the end of this window, the client should request the full MPD, rather than the patch.

If all the conditions hold, the DASH client is expected to use the MPD Patch to generate a new MPD at the client.

6 Segment formats

6.1 General

The Segment formats specify the syntax and semantics of the resources that are associated with HTTP-URLs identified by the MPD, or directly provided in data URLs. For example, an HTTP GET request to a resource identified in the MPD is responded with an HTTP response including an entity body that conforms to a segment format.

Different Segment types are defined in subclause 6.2.

This document focuses on Segment formats based on MPEG container formats. Specifically,

- in subclause 6.3, Segment formats are described for use with Media Segments based on the ISO Base Media File Format as defined in ISO/IEC 14496-12;
- in subclause 6.4, Segment formats are described for use with Media Segments based on the MPEG-2 Transport Stream as defined in the ISO/IEC 13818-1.

In both cases, the Segment formats are defined such that the Media Segment formats comply with the respective container formats.

Guidelines for adding other Segment formats are provided in Annex F.

6.2 Segment types

6.2.1 General

Four different Segment types are defined:

- Initialization Segments containing initialization information for accessing the Representation in subclause 6.2.2,
- Media Segments containing encoded media content components in subclause 6.2.3,
- Index Segments primarily containing indexing information for Media Segments in subclause 6.2.4,
- Bitstream Switching Segments containing essential data to switch to the Representation to which it is assigned in subclause 6.2.5.

6.2.2 Initialization Segment

The Initialization Segment contains initialization information for accessing the Representation. The Initialization Segment shall not contain any media data with an assigned presentation time.

NOTE The Initialization Segment is conceptually processed by the media engine in Figure 2 to initialize the media engines for enabling play-out of Media Segments of the containing Representation.

The Initialization Segment is media format specific and more details shall be defined for each media format that permits or requires the presence of an Initialization Segment.

6.2.3 Media Segment

6.2.3.1 General

A Media Segment contains and encapsulates media streams that are either described within this Media Segment or described by the Initialization Segment of this Representation or both.

In addition, a Media Segment

- 1) shall contain a number of complete access units.
- 2) should provide information on how to access the Media Presentation within this Segment, e.g. exact presentation time and an index. There is no requirement that a Media Segment starts with a SAP, but it is possible to signal in the MPD that all media streams in a Segments within a Representation start with a SAP.
- 3) shall contain only media streams that start with a SAP of type 1 or 2, if it is the first Media Segment in the Representation.
- 4) shall contain sufficient information to time-accurately present each contained media component in the Representation. The time-accuracy enables a client to seamlessly switch Representations and jointly present multiple Representations.
- 5) may be divided into Subsegments by a Segment Index as defined in subclause 6.2.3.2. In some media formats, the Segment Index may be contained in the Media Segment. In other formats, the Segment Index may be included in a dedicated Index Segment. For more details on Index Segments, refer to subclause 6.2.4.

- 6) shall specify all Media Presentation times relative to the start of the Period and compensated with the value of the `@presentationTimeOffset`. The presentation time in Media Segments shall be accurate to ensure accurate alignment of all Representations in one Period. For more details, refer to subclause 7.2.1.

The Media Segment is media format specific and more details are specified for individual media formats.

6.2.3.2 Subsegments and Segment Index

Media Segments may contain multiple Subsegments. Each Subsegment shall contain a number of complete access units. There may also be media-format-specific restrictions on Subsegment boundaries. If a Segment is divided into multiple Subsegments, this division is described by a compact Segment index, which provides the presentation time range in the Representation and corresponding byte range in the Segment occupied by each Subsegment for one or more media streams. Clients may download this index in advance and then issue requests for individual Subsegments.

NOTE Segment Index information is conceptually processed by the DASH access client in Figure 2 in order to access Subsegments by the use of HTTP partial GET requests.

In addition, the Segment Index provides timing and stream access information. This includes the earliest presentation time of access units in each Subsegment of an indexed media stream and the presentation time of the first SAP, if present.

If a Segment Index is present for at least one media stream, then for any media stream for which no Segment Index is present, referred to as non-indexed stream, the following applies:

- every access unit of the non-indexed streams shall be a SAP of type 1.
- for each Subsegment, every non-indexed stream shall contain exactly one access unit within the Subsegment with presentation time less than or equal to the earliest presentation time of the Subsegment.

When multiple media streams are indexed in a single index file, the corresponding Segment Index for different media streams should index the same number of Subsegments.

If no Segment Index is provided for a Media Segment, then the Media Segment constitutes one Subsegment.

The Segment Index may be included in the Media Segment, typically in the beginning of the file. Segment Index information may also be provided in separate Index Segments as defined in subclause 6.2.4. A Subsegment may itself be further subdivided using further Segment Indices. If a Subsegment only contains media data but no Segment Index, it is referred to as Media Subsegment.

The Segment Index may contain additional Subsegment indexing information for accessing different levels of Subsegments in a Media Subsegment. For more details, refer to subclause 6.2.3.3.

A generic mechanism for indexing of Media Segments is provided by the Segment Index ('`sidx`') box in ISO/IEC 14496-12. This indexing applies to all media formats defined in this document. In this case,

- the earliest presentation time of a Subsegment is documented in the `earliest_presentation_time` field.
- the byte range is documented by the `first_offset` field and the `reference_size` field. If two Segment Index boxes document the same byte range, then the value of their `reference_size` field shall be identical, and the values their `first_offset` field shall point to the same byte offset in the file.

6.2.3.3 Subsegment Index

Media Subsegments may be indexed further to enable accessing different levels of Subsegments in a Media Subsegment. This Subsegment Index may also be provided in separate Index Segments together with the Segment Index.

A generic syntax and semantic for Subsegment indexing is provided by the Subsegment Index ('ssix') in ISO/IEC 14496-12.

6.2.4 Index Segment

Index Segments contain information that is related to Media Segments and primarily contain indexing information for Media Segments. An Index Segment may provide information for one or more Media Segments.

The Index Segment may be media format specific and more details shall be defined for each media format that permits Index Segments.

6.2.5 Bitstream Switching Segment

A Bitstream Switching Segment contains data essential for switching to the Representation it is assigned to.

The Bitstream Switching Segment is media format specific and more details shall be defined for each media format that permits Bitstream Switching Segments.

6.2.6 Missing Content Segment

A Missing Content Segment is defined for the purpose to extend the Segment Timeline even if the data in the Segment is not present or is only partially present. This Segment contains no or a subset of media samples of the media timeline duration it is representing. If the Segment is not a decodable media segment, then the segment shall include a major brand 'miss'. The Segment may additionally contain an 'emsg', for example, to indicate an MPD validity expiration event.

In addition, for a Missing Content Segment, the following shall hold:

- a single Segment Index ('sidx') box shall be present and the values of the Segment Index shall describe accurate timing of the Segment Timeline, i.e.,
 - the `earliest_presentation_time` in the 'sidx' box shall be the value of **S**@t as signalled in the Segment Timeline in the MPD
 - the `Subsegment_duration` fields in the 'sidx' box shall be the value of **S**@d as signaled in the Segment Timeline in the in the MPD

By forcing these values, the DASH Client is able to properly extend the Segment Timeline even though no actual media data may be present.

6.3 Segment formats for ISO base media file format

6.3.1 General

This subclause defines Segment formats based on the ISO Base Media File Format as specified in ISO/IEC 14496-12. All Segment formats defined in subclause 6.3 shall contain one or more boxes in accordance with the box structure of the ISO base media file format ISO/IEC 14496-12.

Refinements on generic concepts are introduced in subclause 6.3.2. Segment formats are defined for Initialization Segments (subclause 6.3.3), Media Segments (subclause 6.3.4) and Self-Initializing Media Segments (subclause 6.3.5). Bitstream Switching Segments and Index Segments are not defined for this media format.

6.3.2 Preliminaries: Refinements of generic concepts

6.3.2.1 Subsegments

Media Subsegments for Media Segments based on the ISO base media file format are defined as a self-contained set of one or more consecutive movie fragments; such a set contains one or more movie fragment boxes with the corresponding media data ('mdat') box(es). A media data box containing data referenced by a movie fragment ('moof') box shall follow that movie fragment box and precede the next movie fragment box, if any, containing information about the same track.

For a Media Subsegment, the value of the `reference_type` field in the describing Segment Index ('sidx') box shall be set to 0.

6.3.2.2 Media stream access points

Different types of stream access points for the ISO base media file format are defined in ISO/IEC 14496-12.

6.3.2.3 Segment Index

If the Segment Index is provided, the Segment Index ('sidx') box in ISO/IEC 14496-12 shall be used. Exact definitions for the use of the Segment Index ('sidx') box with media formats based on the ISO base media file format are specified in ISO/IEC 14496-12.

6.3.2.4 Subsegment Index

If the Subsegment Index is provided, the Subsegment Index ('ssix') box in ISO/IEC 14496-12 shall be used. Exact definitions for the use of the Subsegment Index ('ssix') box for the use with media formats based on the ISO base media file format are specified in ISO/IEC 14496-12.

6.3.2.5 Resynchronization Point

A Resynchronization Point for the ISO BMFF is defined as the start of ISO BMFF Segment as defined in ISO/IEC 14496-12 with the restrictions in terms of both, cardinality and ordinality as defined in Table 51.

Table 51 — Resynchronization Point for the ISO BMFF - Cardinality and Ordinality

NL 0	Cardinality	Specification	Constraints	Description
styp	0/1	ISO/IEC 14496-12	DASH/CMAF constraints	Segment Type Signalling compatibility to CMAF Chunk
prft	0/1	ISO/IEC 14496-12	DASH/CMAF constraints	Producer Reference Time
emsg	*	ISO/IEC 23009-1	DASH/CMAF constraints	Event Message
free	*	ISO/IEC 14496-12	none	free box
skip	*	ISO/IEC 14496-12	none	skip box
moof	1	ISO/IEC 14496-12	DASH/CMAF constraints	Movie Fragment box and the boxes it contains
mdat	1	ISO/IEC 14496-12	DASH/CMAF constraints	Media Data container for media samples

Cardinality 0/1 means that at most one needs to be present, * means that any number or none of those are present.

NL refers to the nesting level, all boxes are on level 0.

For ISO BMFF based Resynchronization Point, the properties are defined as follows

- The index `Index` is defined as the byte offset of the first octet of the constrained ISO BMFF Segment.
- The earliest presentation time `Time` is defined as the smallest time of the combination of the decode time, the composition offset and the edit list, of any sample in this ISO BMFF Segment.
- The SAP type is defined according to subclause 4.5.2.
- If the '`styp`' is present and the box following the '`styp`' is one of the following '`prft`', '`emsg`', '`free`', '`skip`', '`moof`', and the cardinality and ordinality of the Resynchronization Point as provided in Table 51 is fulfilled then the marker is present.

6.3.3 Initialization Segment format

The Initialization Segment shall conform to the ISO base media file format.

The Initialization Segment shall contain an "`ftyp`" box, and a "`moov`" box. It shall not contain any "`moof`" boxes. It may contain other boxes, such as the "`pdin`" box. The tracks in the "`moov`" box shall contain no samples (i.e. the `entry_count` in the "`stts`", "`stsc`", and "`stco`" boxes shall be set to 0), and the "`moov`" box is thus small.

NOTE 1 This can reduce the start-up time significantly as the Initialization Segment needs to be downloaded before any Media Segment can be processed.

The "`mvex`" box shall be contained in the "`moov`" box to indicate that the client has to expect movie fragments. The "`mvex`" box also sets default values for the tracks and samples of the following movie fragments.

The Initialization Segment provides the client with the metadata that describes the encoding of the media content, specifically of the Representation. The media engine in the client uses the information in the "`moov`" box to identify the available media content components and their characteristics.

NOTE 2 It is expected that the media engine in the DASH Clients does not require any information in the MPD for successful decoding and presentation of the contained media streams.

6.3.4 Media Segment types

6.3.4.1 General

Media Segments can be of different types: Delivery Unit Media Segments, simple Media Segments, Random Access Media Segments, Switching Media Segments, Indexed Media Segments, Sub-Indexed Media Segments.

All Media Segments shall conform to the general definitions in subclause 6.3.4.2. Additional type-specific constraints are provided further below in subclause 6.3.4.

Further rules on Media Segments in combination with certain MPD attributes are provided in subclause 7.3.

Media Segments may conform to multiple types. Conformance can be expressed by adding the brand(s) to the '`styp`' box as a compatible brand and, if applicable, as the major brand.

Unless explicitly mentioned differently, the boxes referred in subclause 6.3.4 are specified in ISO/IEC 14496-12.

6.3.4.2 Delivery Unit Media Segment

A Media Segment conforming to the Media Segment Format is defined as follows:

- Each Media Segment shall contain one or more whole self-contained movie fragments. A whole, self-contained movie fragment is a movie fragment ('moof') box and a media data ('mdat') box that contains all the media samples that do not use external data references referenced by the track runs in the movie fragment box.
- Each 'moof' box shall contain at least one track fragment.
- The 'moof' boxes shall not use external data references, the flag 'default-base-is-moof' shall be set, and data-offset shall be used, i.e. 'base-data-offset-present' shall not be used. This combination of settings is referred to as movie-fragment relative addressing for media data.
- Absolute byte-offsets shall not be used for this media data. In a movie fragment, the duration by which each track extends should be as close to equal as practical. In particular, as movie fragments are accumulated, the track durations should remain close to each other and there should be no 'drift'.
- Each Media Segment may carry 'dums' in the Segment Type box ('styp') as a compatible brand. The conformance requirements of this brand are defined in this subclause.

6.3.4.3 Simple format type

A Media Segment conforming to the Simple Media Segment Format for DASH is defined as follows:

- It shall conform to the Delivery Unit Media Segment format as specified in subclause 6.3.4.2.
- Each 'traf' box shall contain a 'tfdt' box.

NOTE The track fragment adjustment box 'tfad' as defined in 3GPP TS26.244 can also be present. DASH Clients are discouraged to apply both the alignment established by the 'tfdt' and the time-shifting implied by the 'tfad', which would result in a double correction.

- Each Simple Media Segment may contain one or more 'sidx' boxes. If present, the first 'sidx' box shall be placed before any 'moof' box and the first Segment Index box shall document the entire Segment.
- For the purpose of determining overlapping and non-overlapping segments, redundant samples as defined in ISO/IEC 14496-12 shall be ignored. In other words, the earliest presentation time of any access unit in the stream shall be computed without taking redundant samples into account.
- Each Media Segment may contain a 'styp' box and if present shall carry 'msdh' as a compatible brand. The conformance requirement of this brand is defined in this subclause.

6.3.4.4 Indexed Media Segment

A Media Segment conforming to the Indexed Media Segment Format is defined as follows:

- Each Media Segment shall comply with the Delivery Unit Media Segment as defined in subclause 6.3.4.2 and in addition in each self-contained movie fragment, the movie fragment ('moof') box is immediately followed by its corresponding media data ('mdat').
- Each Media Segment shall contain one or more 'sidx' boxes. The first 'sidx' box shall be placed before any 'moof' box and shall document Subsegments that span the composition time of the entire Segment.

- Each Media Segment shall carry 'msix' as a compatible brand. The conformance requirements of this brand are defined in this subclause.

6.3.4.5 Sub-Indexed Media Segment

A Media Segment conforming to the Sub-Indexed Media Segment Format is defined as follows:

- It shall conform to the indexed Media Segment format as specified in subclause 6.3.4.3.
- The Subsegment Index box ('ssix') shall be present and shall follow immediately the 'sidx' box that documents the same Subsegment. This immediately preceding 'sidx' shall only index Media Subsegments.
- It shall carry 'sims' in the Segment Type box ('styp') as a compatible brand. The conformance requirements of this brand are defined in this subclause.

6.3.4.6 Random Access Media Segment

A Media Segment conforming to the Random Access Media Segment Format is defined as follows:

- It shall conform to the Simple format as specified in subclause 6.3.4.3.
- The first access unit in each movie fragment in a Random Access Media Segment shall correspond to the I_{sau} of a SAP of type 1, 2, or 3.
- The media segment shall carry sufficient information to access the media in the stream, e.g. all necessary encryption in combination with the Initialization Segment, if available.

6.3.5 Self-Initializing Media Segment formats

6.3.5.1 General format type

The Self-Initializing Media Segment is conformant with the ISO base media file format and defines the DASH Self-Initializing Media Segment 'dsms' brand.

The Self-Initializing Media Segment is conformant with the ISO base media file format.

NOTE Since one Representation only contains one self-initializing Media Segment, switching is expected to happen within the Segment, e.g. at a Subsegment that contains a SAP.

6.3.5.2 Indexed self-initializing Media Segment

The Indexed Self-Initializing Media Segment conforms to the concatenation of an Initialization Segment and a single Indexed Media Segment without the 'styp' box preceding the Media Segment and shall carry 'dash' as a compatible brand.

The format of the Indexed self-initializing Media Segment is a conforming ISO base media file format file and defines the 'dash' brand.

6.4 Segment formats for MPEG-2 transport streams

6.4.1 General

This subclause introduces Segment formats that are suitable to be used if Media Segments are valid MPEG-2 TS, conforming to ISO/IEC 13818-1.

NOTE It is possible to encapsulate MPEG-2 TS formatted media within an ISO base media file format. This mode of operation is not discussed in this subclause. If MPEG-2 TS formatted media is encapsulated in an ISO base media file format, then the rules as defined in subclause 6.3 apply.

Refinements on generic concepts are introduced in subclause 6.4.2. Segment formats are defined for Initialization Segments (see subclause 6.4.3), Media Segments (see 6.4.4), Bitstream Switching Segments (see subclause 6.4.5) and Index Segments (see subclause 6.4.6). MPEG-2 TS specific box structures are defined in subclause 6.4.7.

6.4.2 Preliminaries: Refinements of generic concepts

6.4.2.1 Subsegment

In the context of MPEG-2 TS based delivery formats, a Subsegment is defined as an indexed set of access units consecutive in decode order. A subsegment shall contain complete access units for the indexed media stream (i.e. stream for which `reference_ID` equals PID); however, it may contain incomplete PES packets from other media streams.

These access units are encapsulated in one or more PES packets. Each PES packet is encapsulated into one or more TS packets with the same PID value.

6.4.2.2 Media stream access points

For the case of MPEG-2 TS, a media stream is equivalent to an Elementary Stream as defined in ISO/IEC 13818-1.

Different types of stream access points are defined in ISO/IEC 14496-12. The same type definitions shall apply for the MPEG-2 TS. More specifically, in the case of MPEG-2 TS, a SAP corresponds to an Elementary Stream Random Access Point, as defined in ISO/IEC 13818-1. Consequently, I_{SAU} is the position of the first (sync) byte of a TS packet with PID assigned to this Elementary Stream. This TS packet contains the first byte of a PES packet, which, in turn, contains the Elementary Stream Access Point. PES packet starting at I_{SAU} shall contain only an integral number of access units and shall contain a PTS.

NOTE 1 I_{SAU} generally corresponds to the start of a TS packet with PID value for one Elementary Stream, the `payload_unit_start_indicator` field set to '1', `adaptation_field_control` set to '11', and the `random_access_indicator` field in the Adaptation Field is set to '1'. For SAP types 1-3, the `random_access_indicator` field in the Adaptation Field is commonly set to '1' (this is the case unless no PES payload bytes are found within the packet payload).

NOTE 2 Following the definitions in this subclause, the first packet of the PCR PID is present at or prior to the TS packet at smallest I_{SAP} . If PCRs are carried on a media PID, the first packet of this PID is the first packet following the initialization data, and carries a PCR. In order to avoid changing the underlying content, the implementer is able to choose adding a packet carrying only adaptation field with a PCR, but no payload. This packet is placed prior to the smallest I_{SAU} of any stream in this Representation.

NOTE 3 If Index Segment is provided, and the 'pcrb' box is present, PCR can be inferred from this box.

6.4.2.3 Segment Index

If the Segment Index is provided, the Segment Index ('`sidx`') box in ISO/IEC 14496-12 shall be used for Segment Indexing. In addition to these definitions, the following conditions shall be met for a Segment Index used to describe MPEG-2 TS based Media Segment:

- `reference_ID` field of '`sidx`' box shall be the PID value of the indexed stream.
- All media offsets within '`sidx`' boxes shall be to the first (sync) byte of a TS packet.

NOTE Times within '`sidx`' boxes are expressed in units of the `timescale` field, rather than in 90 kHz clock ticks.

6.4.2.4 Subsegment Index

If the Subsegment Index is provided, the Subsegment Index ('ssix') box in ISO/IEC 14496-12 shall be used for indexing byte ranges within a subsegment. In addition to these definitions, the following conditions shall be met for a Segment Index used to describe MPEG-2 TS based Media Segment.

- All media offsets within 'ssix' boxes shall be to the first (sync) byte of a TS packet.

6.4.3 Initialization Segment types and formats

6.4.3.1 Initialization information

Initialization information is any information necessary to enable the media engine to start decoding the payload of any TS packet belonging to any media stream within a (Sub)Segment.

Untimed initialization information includes PAT, CAT, PMT, EMM, and any other PSI information possibly included by the Media Presentation author. Any additional information that does not alter the Media Presentation timeline is allowed.

Time-varying initialization information is information that is required for the successful start of playout but is different for at least two Subsegments or Segments within a Representation.

Mandatory initialization information summarizes information that shall be present prior to any media data to enable decoding and presentation. As a consequence, mandatory initialization information includes at least the following information, in this order:

- PAT (untimed, unless changes within the Representation);
- PMT (untimed, unless changes within the Representation);
- PCR (time-varying).

If MPEG-2 Conditional Access is used, ECM is considered mandatory untimed initialization information if it does not change for the whole duration of the Period; otherwise it is considered mandatory time-varying initialization information.

6.4.3.2 Initialization Segment

An Initialization Segment shall be a valid MPEG-2 TS, conforming to ISO/IEC 13818-1.

The concatenation of an Initialization Segment with any Media Segment shall have the same presentation duration as the original Media Segment.

The Initialization Segment shall contain mandatory untimed initialization information as defined in subclause 6.4.3.1. Time-varying initialization information shall not be present in the Initialization Segment, i.e.

- PCR-bearing packets shall not be present in the Initialization Segment;
- ECM may be present as long as it does not change within the entire Representation;
- Any PSI table may be present as long as it does not change within the entire Representation.

The Initialization Segment shall contain only complete sections.

Initialization Segment may or may not be present. If it is not present for a given Representation, all Media Segments belonging to this Representation shall be self-initializing. Also, if an Initialization Segment is used, not all initialization information needs to reside in the Initialization Segment, only presence of complete initialization information in the concatenation of Initialization Segment and Media Segment is required.

6.4.4 Media Segment types and formats

6.4.4.1 General

All Media Segments shall conform to the basic Media Segment in subclause 6.4.4.2.

Further rules on Media Segments in combination with certain MPD attributes are provided in subclause 7.4.

6.4.4.2 Basic Media Segment

A Media Segment shall be a valid MPEG-2 TS, conforming to ISO/IEC 13818-1.

As a consequence of the requirement in subclause 5.3.5.1, the concatenation of consecutive Media Segments of the same Representation shall also yield a valid MPEG-2 TS conforming to ISO/IEC 13818-1.

In addition, the following conditions shall be met:

- Media Segments shall contain complete MPEG-2 TS packets,
- Media Segments shall contain exactly one program,
- All time-varying initialization information shall be present between I_{SAP} and I_{SAU} and/or in the Index Segment, if present,
- No Media Segment shall depend on initialization information appearing in any preceding Media Segment.

Media Segments should contain only complete PES packets and sections. Each PES packet should be comprised of one or more complete access units in each packet. Media Segments should contain only complete access units.

6.4.4.3 Content Protection

All information necessary for decrypting, or locating information required to decrypt, the encrypted TS packets in a (Sub)Segment shall be present before the encrypted packet(s) to which they apply, either in the same (Sub)Segment, and/or in the Initialization Segment (if used). As an example, this requires the presence of the ECM necessary for decrypting the first encrypted packet of the (Sub)Segment is within the (Sub)Segment before such a packet. A Subsegment may not have an ECM preceding the first encrypted packet if the location of this ECM can be determined using an Index Segment.

NOTE Sub-Representations can be arranged such that information such as ECM is included in all Sub-Representations that need them, for example by assigning the ECM an individual level and add dependency on all relevant Sub-Representations on this level.

6.4.4.4 Self-initializing Media Segment

A Self-initializing Media Segment conforms to the basic Media Segment as defined in subclause 6.4.4.2 and in addition shall contain at least all mandatory untimed and timed initialization information as defined in subclause 6.4.3.1.

All required initialization information as defined in subclause 6.4.3.1 should be present prior to any media data.

6.4.5 Bitstream Switching Segment

A Bitstream Switching Segment shall be a valid MPEG-2 TS, conforming to ISO/IEC 13818-1.

A Bitstream Switching Segment, when concatenated with any Media Segment, shall not alter the Media Presentation timeline for the corresponding Media Segment.

If initialization information is carried within a Bitstream Switching Segment, it shall be identical to the one in the Initialization Segment, if present, of the Representation.

NOTE Authors are encouraged to use Bitstream Switching Segments when there is a reasonable expectation of non-conforming behaviour (such as continuity counter errors, etc.) at the concatenation point of two consecutive Media Segments from different Representation, lack of correct initialization information (two Representations with different initialization information).

6.4.6 Index Segment

6.4.6.1 General

Index Segments consist of a sequence of ISO BMFF-box-structures.

Index Segments may either be associated to a single Media Segment as specified in subclause 6.4.6.2 or may be associated to all Media Segments in one Representation as specified in subclause 6.4.6.3. An Index Segment may also contain a Subsegment Index as specified in subclause 6.4.6.4 and any other boxes defined in subclause 6.4.7.

It is recommended that Index Segments be at least provided for one media stream.

NOTE 1 Although the Media Segments are MPEG-2 TS based, Index Segments are reusing ISO BMFF-box-structures. This allows the DASH access client in the model of Figure 2 to be universal and independent of the Media Format.

NOTE 2 Index Segments are not valid ISO BMFF files, and complete implementation of ISO BMFF is not necessary to utilize indexes in a MPEG-2 TS based client. A partial implementation would suffice, since only few ISO BMFF boxes, such as 'styp', 'sidx', and 'ssix', are required in order to parse an MPEG-2 TS Index Segment.

Other box types may be present in an MPEG-2 TS Index Segment, but, if present, they shall not contain information required to interpret the 'styp', 'sidx' or 'ssix' boxes.

6.4.6.2 Single Index Segment

A Single Index Segment indexes exactly one Media Segment and is defined as follows:

- Each Single Index Segment shall begin with a 'styp' box, and the brand 'sixx' shall be present in the 'styp' box. The conformance requirement of the brand 'sixx' is defined in this subclause.
- Each Single Index Segment shall contain one or more 'sidx' boxes which index one Media Segment.
- A Single Index Segment may contain one or multiple 'ssix' boxes. If present, the 'ssix' shall follow the 'sidx' box that documents the same Subsegment without any other 'sidx' preceding the 'ssix'.
- A Single Index Segment may contain one or multiple 'pcrb' boxes as defined in subclause 6.4.7.2. If present, 'pcrb' shall follow the 'sidx' box that documents the same Subsegments, i.e. a 'pcrb' box provides PCR information for every subsegment indexed in the last 'sidx' box.

6.4.6.3 Representation Index Segment

A Representation Index Segment indexes all Media Segments of one Representation and is defined as follows:

- Each Representation Index Segment shall begin with an 'styp' box, and the brand 'rix' shall be present in the 'styp' box. The conformance requirement of the brand 'rix' is defined by this subclause;

- Each Media Segment is indexed by one or more Segment Index box(es); the boxes for a given Media Segment are contiguous;
- Each Segment Index box may be followed by an 'ssix' and/or 'pcrb' box;
- The Segment Index for each Media Segments is concatenated in order, preceded by a single Segment Index box that indexes the Index Segment. This initial Segment Index box shall have one entry in its loop for each Media Segment, and each entry refers to the Segment Index information for a single Media Segment.

The structure of a Representation Index Segment is shown in Figure 6. This figure illustrates a case where a Representation Index Segment is provided and the Subsegment Index is used in order to enable efficient trick mode operation. The figure shows four consecutive Subsegments, S0, S1, S2, and S3, each indexed by an 'sidx' box, and two temporal layers within a video stream, I frames (L0) and P frames (L1), indexed by an 'ssix' box.

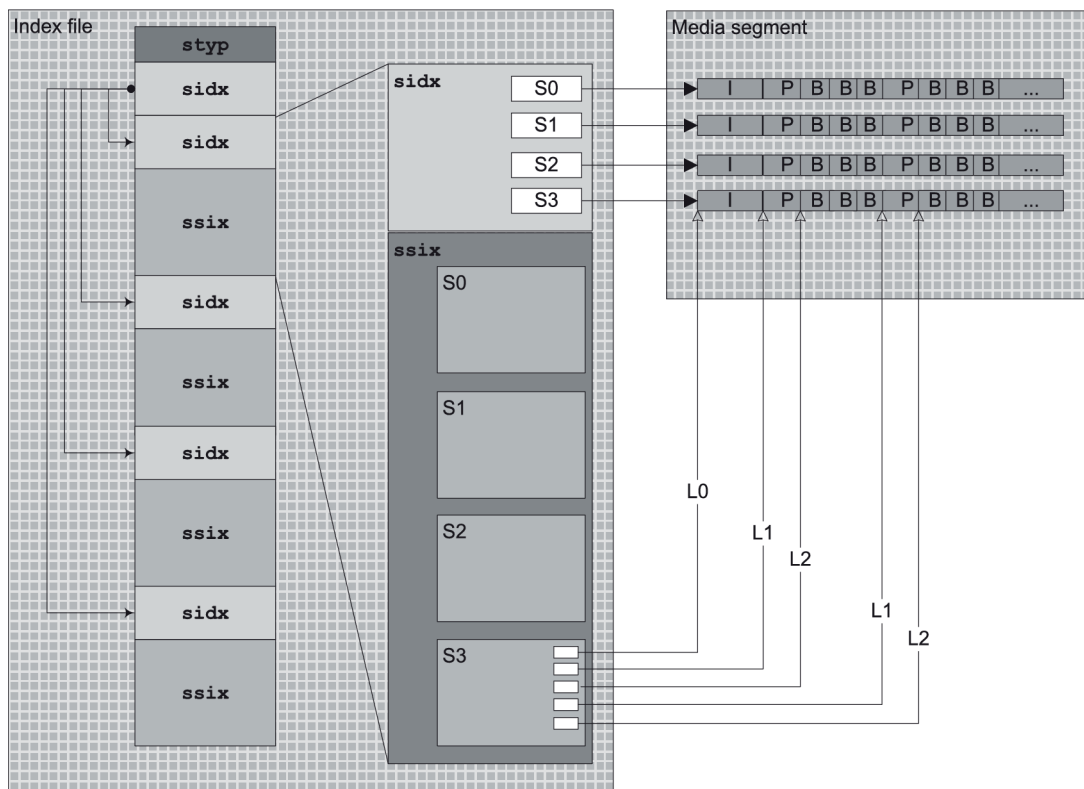


Figure 6 — Structure of Representation Segment Index

6.4.6.4 Subsegment Index Segment

A Subsegment Index Segment shall conform to an Index Segment and shall also include a Subsegment Index. A Subsegment Index Segment is defined as follows:

- It shall be either a Single Index Segment or a Representation Index Segment.
- The Subsegment Index box ('ssix') shall be present and shall follow immediately the 'sidx' box that documents the same Subsegment. The value of the `reference_type` field shall be equal to 0

for this Subsegment in this immediately preceding Segment Index ('sidx') box. If the 'pcrb' box is present, it shall follow 'ssix'.

- It shall carry 'ssss' in the Segment Type box ('styp') as a compatible brand. The conformance requirement of this brand is defined in this subclause.

6.4.7 Boxes used with MPEG-2 TS Index Segments

6.4.7.1 General

Index Segments may contain additional auxiliary information contained in boxes conforming to the ISO base media file format boxes. Boxes exclusively relevant for the MPEG-2 TS Media Segments are documented in subclause 6.4.7.

6.4.7.2 MPEG-2 TS PCR information box

6.4.7.2.1 Definition

Box Type: 'pcrb'

Container: File

Mandatory: No

Quantity: Zero or one

Signals the PCR information for MPEG-2 TS.

6.4.7.2.2 Syntax

```
aligned(8) class MPEG2TSPCRInfoBox extends Box('pcrb', 0) {
    unsigned int(32)    subsegment_count;
    for( i=1; i <= subsegment_count; i++){
        unsigned int(42)    pcr;
        unsigned int(6)     pad = 0;
    }
}
```

6.4.7.2.3 Semantics

- `subsegment_count` is a positive integer specifying the number of Subsegments for which partial Subsegment information is specified in this box. `subsegment_count` shall be equal to `reference_count` in the last Segment Index box.
- `pcr` for each iteration of the loop indicates the MPEG-2 TS PCR corresponding to the first (sync) byte of the first MPEG-2 TS packet in the media Subsegment corresponding to the current iteration. If this TS packet carries a PCR, its value will be different from the one specified in this field, since ISO/IEC 13818-1 defines PCR as relative to the byte containing the last bit of the `program_clock_reference_base` field.

7 Combined semantics of MPD and Segment formats

7.1 Overview

An MPD and the referenced Segments comprise a Media Presentation. The formats for these two key components of a DASH-compatible Media Presentation are defined in Clauses 5 and 5.15. In this clause, Media Presentation authoring rules are provided on how the MPD and different Segment formats may be combined to establish a complete Media Presentation.

Specifically, aspects are addressed that deal with the Segment, that have special alignment with the Segments of other Representations to enable and simplify seamless switching and joint presentation.

General Media Presentation authoring rules are provided in subclause 7.2 and specific ones for each media format are provided in the remainder of Clause 7. Specifically, rules when using the ISO base media file format are provided in subclause 7.3 and the rules when using the MPEG-2 TS are provided in subclause 7.4. Guidelines for other formats are provided in Annex F.

NOTE Representation metadata present in the MPD can also be repeated in the media streams, e.g. in an Initialization Segment or a Media Segment. The Media Presentation is expected to be provided such that no mismatch between these two values occurs. If it does, the value in the media stream itself takes precedence over values expressed in the MPD, especially when used in the media decoding process. In addition, it is important to consider that metadata in the MPD is primarily used for selection of Representations by the DASH Client, whereas data in the media stream is used by the media decoder in order to establish the decoding and rendering process. Metadata in the MPD is preferably added if the Media Presentation author expects that the DASH Client can make use of this information in the selection process.

7.2 General

7.2.1 Media Presentation timeline

One of the key features in DASH is that encoded versions of different media components share a common timeline. The presentation time of access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components.

The presentation times within each Period are relative to the *PeriodStart* time of the Period minus the value of the `@presentationTimeOffset`, T_0 , of the containing Representation. This means for an access unit with a presentation time T_P signalled in the media stream, the Media Presentation time relative to the *PeriodStart* is $T_M = T_P - T_0$.

A Representation should cover all presentation times T_P that are greater than and equal to the value of the `@presentationTimeOffset`, T_0 . If smaller values are present, i.e. the value of the `@eptDelta` is negative, then presentation of the Media Segment is expected to only take place for presentation times greater than or equal to T_0 .

In case the `@duration` attribute is used for the signalling of the duration of Segments, then the MPD start times as defined in subclause 5.3.9.5.3 should provide an approximation of the Media Presentation time T_M within the Period. Specifically, the difference between MPD start time and presentation time T_P shall not exceed 50% of value of `@duration` divided by the value of the `@timescale` attribute. In case the Segment Timeline is used for the signalling of the duration of Segments, then the MPD start times as defined in subclause 5.3.9.5.3 shall provide exactly the Media Presentation time T_M within the Period, i.e. the MPD start time is the earliest presentation time of the Segment.

At the start of a new Period, the playout procedure of the media content components may need to be adjusted at the end of the preceding Period to match the *PeriodStart* time of the new Period as there may be small overlaps or gaps with a Representation at the end of the preceding Period. Overlaps (respectively gaps) may result from Media Segments with actual presentation duration of the media stream longer (respectively shorter) than indicated by the Period duration.

There may be cases where the Media Presentation author observes issues in generating media, especially for live services. This can be because the input signal to the encoder is not available or the encoder is down. Generally, the Media Presentation author should address these issues by providing redundant architectures or by the use of specific outage or blackout signals. Also, if the format in use permits, empty Segments or zero duration Segments may be used.

However, if such remedies are not available, the Media Presentation author may signal gaps in the timeline. If the Segment Timeline is in use as defined in subclause 5.3.9.6, the gaps in the timeline may be explicitly signalled. Alternatively, gaps in the segment timeline may be signalled by leaving gaps.

If no such means are available, early termination Periods may be used as defined in subclause 5.3.2. In this case, the gaps at the end of the Period may be longer than indicated above. A client is expected to overcome such outages and continue the Media Presentation with the availability of a new Period.

For the case when **MPD@type** is "dynamic" and the attribute **MPD@suggestedPresentationDelay** is present, then the sum of value of the **MPD@availabilityStartTime**, the *PeriodStart* value, the presentation time within the Period of an access unit, T_M , and the value of the attribute **MPD@suggestedPresentationDelay** provides a mapping of the presentation time of each access unit to the wall-clock time, for example to express synchronization with a content internal time or for other reasons to enable synchronization of presentation to the wall-clock.

Further media format specific definitions of presentation time may be defined.

7.2.2 Segment Index

If a Segment Index is present in a Media Segment of one Representation within an Adaptation Set, then the following shall hold:

- the order of Segment Index boxes for multiple media streams induces an ordering on the media content components equal to the order in which a Segment Index box for a media stream for each component first appears. This ordering shall be the same for all Segments of all Representations of an Adaptation Set. As a consequence, if there is a Segment Index for a media content component in one Segment, there shall be a Segment Index for that media component in all Segments in this Adaptation Set.
- non-indexed media streams in all Representations of an Adaptation Set shall have the same access unit duration.

7.2.3 Segment alignment

The requirements stated in subclause 5.3.3.2 shall apply.

7.2.4 Subsegment alignment

The requirements stated in subclause 5.3.3.2 shall apply.

7.3 Media Presentation based on the ISO base media file format

7.3.1 General

The Media Presentation as introduced in 5 and 5.15 is instantiated in this subclause using the ISO base media file format as defined in ISO/IEC 14496-12 as Segment formats.

An ISO BMFF-based DASH Media Presentation is described by an MPD as specified in subclause 5.1. The MIME type of the MPD shall be as defined in Annex C.

The general rules defined in subclause 7.2 shall apply.

The **@mimeType** attribute of each Representation shall be provided according to IETF RFC 4337. Additional parameters may be added according to IETF RFC 6381.

If present, the **@SegmentProfiles** shall provide a whitespace-separated list of the individual Segment profile identifiers.

The following Segment types and formats may be used:

- Initialization Segments complying with formats as defined in subclause 6.3.3.
- Media Segments complying with formats as defined in subclause 6.3.4.2.
- Self-Initializing Media Segments complying with formats as defined in subclause 6.3.5.

For ISO BMFF-based Media Presentation, the following applies:

- 1) In all cases for which a Representation contains more than one Media Segment, the following applies:
 - i) The Initialization Segment as defined in subclause 6.3.3 shall be present.
 - ii) Media Segments shall not be self-initializing. The Media Segment format is defined in subclause 6.3.4.
 - iii) If the Media Segment is the last Media Segment in the Representation, this Media Segment may carry the 'lmsg' compatibility brand. If the Media Segment is not the last Media Segment in the Representation, the 'lmsg' compatibility brand shall not be present. The 'lmsg' type is defined in this subclause.
- 2) In case a Representation contains only a single Media Segment, then one of the following two options are valid.
 - One Initialization Segment as defined in subclause 6.3.3 and one Media Segment as defined in subclause 6.3.4 that is not self-initializing.
 - One Self-Initializing Media Segment as defined in subclause 6.3.5.

Index Segments may be present.

The content authoring rules for the Media Segments in combination with certain MPD attributes for ISO BMFF-based DASH are provided in subclause 7.3.2.

In case Sub-Representations are used, the rules in subclause 7.3.4 shall apply.

7.3.2 Media presentation timeline

The presentation time T_P internal in the media that maps the media to the Media Presentation timeline shall be relative to the movie timeline, i.e. they are composition times after the application of any edit list for the track using 'sid_x', as defined in ISO/IEC 14496-12.

It is recommended that the @timescale attribute in the MPD matches the timescale field in the Media Header Box of a present track. If the Segment Index ('sid_x') box is present, then it is further recommended that the track for which the Segment Index ('sid_x') box appears first in the Media Segment is the track defining the value of the @timescale attribute.

7.3.3 Authoring Rules for specific MPD attributes

7.3.3.1 Segments starting with media stream access points

No additional requirements beyond those stated in subclauses 5.3.3.2 and 6.3.2.2 are defined.

7.3.3.2 Bitstream switching

If the @bitstreamSwitching is set to 'true' for a set of Representations within an Adaptation Set, the conditions stated in subclause 5.3.3.2 shall be satisfied and the Bitstream Switching Segment shall not be present.

As a consequence of `@bitstreamSwitching` being set to 'true', at least the following conditions are satisfied:

- The track IDs for the same media content component are identical for each Representation in each Adaptation Set.
- The conditions required for setting the `@segmentAlignment` attribute to 'true' for the Adaptation Set are fulfilled.
- The conditions required for setting (i) the `@startWithSAP` attribute to 2 for the Adaptation Set, or (ii) the conditions required for all Representations within the Adaptation Set to share the same value of `@mediaStreamStructureId` and setting the `@startWithSAP` attribute to 3 for the Adaptation Set, are fulfilled.

7.3.4 Sub-Representations

If a **SubRepresentation** element is present in a Representation in the MPD and the attribute **SubRepresentation@level** is present, then the Media Segments in this Representation shall conform to a Sub-Indexed Media Segment as defined in subclause 6.3.4.4. The Initialization Segment shall contain the Level Assignment ('level') box.

The attribute `@level` specifies the level to which the described Sub-Representation is associated to in the Subsegment Index. The information in Representation, Sub-Representation and in the Level Assignment ('level') box contains information on the assignment of media data to levels.

Media data should be ordered such that each level provides an enhancement compared to the lower levels.

7.3.5 Segment Timeline without Segment Index

If the Segment Timeline is in use and the `$Time$` templating is applied and no Segment Index ('sidx') box is present in the Media Segment, then:

- a single track shall be present in the Media Segment;
- a single movie fragment header shall be present in the Media Segment;
- the `baseMediaDecodeTime` in the 'tfdt' of the first movie fragment shall be the earliest presentation time of the Segment and may be used for generating the URL for this segment;
- the sum of all `sample_duration` of track run boxes ('trun') of the only track fragment box shall be the presentation duration of the Segment and may therefore be used to derive the address of the next Media Segment from the actual Media Segment without requiring an updated MPD.

7.4 Media Presentation based on MPEG-2 TS

7.4.1 General

In this subclause, a Media Presentation is instantiated based on Media Segment Formats using the MPEG-2 TS as defined in ISO/IEC 13818-1. A MPEG-2 TS-based DASH Media Presentation is described by an MPD as specified in subclause 5.2. The MIME type of the MPD shall be as defined in Annex C.

The general rules defined in subclause 7.2 shall apply.

The `@mimeType` attribute of each Representation shall be "video/mp2t".

The following Segment types and formats may be used:

- Initialization Segments complying with formats as defined in subclause 6.4.3,
- Media Segments complying with formats as defined in subclause 6.4.4,
- Bitstream Switching Segments complying with formats as defined in subclause 6.4.5,
- Index Segments complying with formats as defined in subclause 6.4.6.

The `@segmentProfiles` attribute may be absent. If present, it is expected to be ignored.

If the Segment Timeline is in use and the `$Time$` templating is applied, the Segment Index shall be present.

7.4.2 Media presentation timeline

The presentation time T_P internal in the media that maps the media to the Media Presentation timeline shall be the one defined by the PTS in the MPEG-2 TS.

More specifically, for one Representation, let $PTS(i)$ be the PTS of the i^{th} access unit in the media stream. Furthermore, let $PTS_A(i)$ be $PTS(i)$ adjusted for 33-bit rollovers, i.e. calculated as if PTS had an infinite amount of bits.

T_P calculation is based on differences between $PTS(i)$ and $PTS(0)$, and therefore $T_P(i) = [PTS_A(i) - PTS_0] * S / 90\ 000$ with PTS_0 typically $PTS(0)$. With appropriate scaling, PTS_0 can be derived from the value of `@presentationTimeOffset` attribute.

NOTE If Index Segment is used, S is provided by in the `timescale` field of the 'sidr' box.

If a media stream contains a discontinuity, the $PTS_A(i)$ calculation assumes relative timing is maintained. Therefore, $PTS_A(i)$ is adjusted by the difference between the value of PCR of the first PCR-bearing packet after the discontinuity and its interpolated PCR value (calculated using the pre-discontinuity PCR rate).

In case of discontinuities, it is recommended to add a new Period to reset the value of `@presentationTimeOffset`.

It is recommended that the `@timescale` attribute in the MPD match the clock frequency S of the elementary streams. If the Segment Index ('sidr') box is present, then it is further recommended that the media stream for which the Segment Index ('sidr') box that appears first in the Index Segment is the elementary stream defining the value of the `@timescale` attribute.

7.4.3 Authoring rules for specific MPD attributes

7.4.3.1 Segments starting with Media stream access points

No additional requirements beyond those stated in subclauses 5.3.3.2 and 6.4.2.2 are defined.

7.4.3.2 Segment alignment

If the `@segmentAlignment` attribute is set to 'true', the requirements stated in subclauses 5.3.2 and 5.3.3.2 shall be met. In addition, the Media Segment shall contain only complete PES packets and sections and only complete access units for each PID, and the first PES packet shall contain a PTS timestamp.

7.4.3.3 Subsegment alignment

If the `@subsegmentAlignment` flag is not set to 'false', the semantics as defined in subclause 5.3.3.2 shall apply. In particular, for an MPEG-2 TS-based Media Presentation, a Subsegment shall contain

only complete PES packets and sections for each PID, and the first PES packet from each elementary stream shall contain a PTS.

7.4.3.4 Bitstream switching

If `@bitstreamSwitching` flag is set to 'true' for a set of Representations within an Adaptation Set, then the conditions stated in subclause 5.3.3.2 shall be satisfied. In addition, the conditions in subclause 5.3.3.2 shall not only hold for the entire sequence from $i=1,...,M$, but for any consecutive sequence of segments with any start index $i_s=1,...,M$ and any end index $i_e=i_s,...,M$.

If `@bitstreamSwitching` flag is set to 'true', the Bitstream Switching Segment may be present, indicated by **BitstreamSwitching** in the Segment Information. In this case, for any two Representations, X and Y, within the same Adaptation Set, concatenation of Media Segment i of X, Bitstream Switching Segment of Representation Y, and Media Segment $i+1$ of Representation Y shall be a MPEG-2 TS conforming to ISO/IEC 13818-1.

As a consequence of the conformance rule as stated in 5.3.3.2, at least the following conditions are satisfied if `@bitstreamSwitching` flag is set to 'true':

- The conditions required for setting the `@startWithSAP` attribute to 2 for the Adaptation Set, or the conditions required for all Representations within the Adaptation Set sharing the same value of `@mediaStreamStructureId` and setting the `@startWithSAP` attribute of the Adaptation Set 3, are fulfilled.
- The conditions required for setting the `@segmentAlignment` attribute set to 'true' for the Adaptation Set are fulfilled.
- PCR shall be present in the Segment prior to the first byte of a TS packet payload containing media data, and not inferred from the 'pcrb' box.

7.4.4 Sub-Representations

If a **SubRepresentation** element is present in a Representation in the MPD and the **SubRepresentation@level** is present, then an Index Segment shall be present and shall conform to the format defined in 6.4.6.4.

The Subsegment Index box shall contain at least one entry for the value of **SubRepresentation@level** and for each value provided in the **SubRepresentation@dependencyLevel**. The remaining attributes of the **SubRepresentation** element should provide sufficient information such that the data contained in the Sub-Representation can be differentiated from the containing Representation as for the MPEG-2 TS no inband assignment of levels is provided.

If Subsegment Index is used for extraction of temporal subsequences, PCR should precede the first bytes of media within the range indicated in the Subsegment index. Also, encryption keys (if used) should not change within the duration of a Subsegment.

8 Profiles

8.1 Definition

Profiles of DASH are defined so as to enable interoperability and the signalling of the use of features.

A profile imposes a set of specific restrictions. Those restrictions are typically on features of the Media Presentation Description (MPD) document and on Segment formats, but may also be on content delivered within Segments, such as on media content types, media format(s), codec(s), and protection formats, or

on quantitative measures such as bit-rates, Segment durations and sizes, as well as horizontal and vertical visual presentation size. Profiles defined in this document define restrictions on features of this document and on Segment formats only (e.g. not codec types). Externally defined profiles may additionally impose restrictions on other aspects.

NOTE 1 A profile can also be understood as permission for DASH Clients that only implement the features required by the profile to process the Media Presentation (MPD document and Segments). However, as DASH Client operation is not specified normatively in this document, it is also unspecified how a DASH Client conforms to a particular profile. Hence, profiles merely specify restrictions on MPD and Segments rather than DASH Client behaviour.

A profile has an identifier, which is a URI. The identifier of a profile shall not contain any comma. The profiles with which an MPD complies are indicated in the **MPD@profiles** attribute as a comma-separated list of profile identifiers. Profile identifiers defined in this document are URNs and shall conform to IETF RFC 8141. Externally defined profiles may use profile identifiers that are URNs or URLs. When a URL is used, it should also contain a month-date in the form *mm/yyyy*; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date, to avoid problems when domain names change ownership.

An MPD is conforming when it satisfies the following:

- 1) The MPD is valid in terms of the schema defined in Annex B.
- 2) The MPD conforms to the requirements defined in this document.
- 3) The MPD conforms to each of the profiles indicated in the **MPD@profiles** attribute as specified below.

When *ProfA* is included in the **MPD@profiles** attribute, the MPD is modified into a profile-specific MPD for profile conformance checking using the following ordered steps:

- 1) The **MPD@profiles** attribute of the profile-specific MPD contains only *ProfA*.
- 2) An **AdaptationSet** element for which **@profiles** does not or is not inferred to include *ProfA* is removed from the profile-specific MPD.
- 3) A **Representation** element for which **@profiles** does not or is not inferred to include *ProfA* is removed from the profile-specific MPD.
- 4) All elements or attributes that are either (i) in this document and explicitly excluded by *ProfA*, or (ii) in an extension namespace and not explicitly included by *ProfA*, are removed from the profile-specific MPD.
- 5) All elements and attributes that “may be ignored” according to the specification of *ProfA* are removed from the profile-specific MPD.

An MPD is conforming to profile *ProfA* when it satisfies the following:

- 1) *ProfA* is included in the **MPD@profiles** attribute.
- 2) The profile-specific MPD for *ProfA* is valid in terms of the schema defined in Annex B.
- 3) The profile-specific MPD for *ProfA* conforms to the normative semantics defined in this document.
- 4) The profile-specific MPD for *ProfA* conforms to the restrictions specified for *ProfA*.

A Media Presentation is conforming to profile *ProfA* when it satisfies the following:

- 1) The MPD of the Media Presentation is conforming to profile *ProfA* as specified above.
- 2) There is at least one Representation in each Period in the profile-specific MPD for *ProfA*.
- 3) The Segments of the Representations of the profile-specific MPD for *ProfA* conform to the restrictions specified for *ProfA*.

NOTE 2 In other words, each MPD contains at least one Representation in each Period, which fulfils the requirements of a profile listed in **MPD@profiles**. There can be stricter rules on the occurrence of Representations in the specified profiles. For example, it can be required that there is at least one Representation for each media type that contains or is inferred to have the profile identifier of a specific profile.

This document defines several profiles.

External organizations or individuals may define restrictions, permissions and extensions by using this profile mechanism. It is recommended that such external definitions be not referred to as profiles, but as *Interoperability Points*. Such an interoperability points may be signalled in the @profiles parameter once a URI is defined. The owner of the URI is responsible to provide sufficient semantics on the restrictions and permission of this interoperability point.

Several profiles are defined relying on the ISO base media FF as Segment formats.

Several other profiles are defined for MPEG-2 TS based Media Segment formats.

All profiles are a subset of the full profile is defined in subclause 8.2.

8.2 Full profile

8.2.1 General

The full profile includes all features and Segment Types defined in this document.

The full profile is identified by the URN "urn:mpeg:dash:profile:full:2011".

8.2.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD as defined in subclause 7.3 or 7.4 shall apply.
- The elements and attributes listed in subclause 5.2.3.2 may be ignored.

8.2.3 Segment format constraints

Representations and Segment formats shall conform to the following constraints:

- Representations shall comply either with the formats defined in subclause 7.3, referring to the Segment formats in subclause 6.3, or to the formats defined in subclause 7.4, referring to the Segment formats in subclause 6.4.

8.3 ISO Base media file format On Demand profile

8.3.1 General

This profile is intended to provide basic support for On-Demand content. The primary constraints imposed by this profile are the requirement that each Representation is provided as a single Segment, that Subsegments are aligned across Representations within an Adaptation Set and that Subsegments

begin with Stream Access Points. This permits scalable and efficient use of HTTP servers and simplifies seamless switching.

The On-Demand profile is identified by the URN "urn:mpeg:dash:profile:isoff-on-demand:2011".

8.3.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and the segments as defined in subclause 7.3 shall apply.
- Representations not inferred to have @profiles equal to the profile identifier as defined in subclause 8.3.1 may be ignored.

NOTE A condition to comply with the restrictions defined in subclause 7.3 is that the @mimeType equals video/mp4, audio/mp4, or application/mp4.

- The elements and attributes listed in subclause 5.2.3.2 may be ignored.
- **MPD**@type shall be "static".
- The **Subset** element may be ignored.
- Neither the **Period.SegmentList** element nor the **Period.SegmentTemplate** element shall be present.
- For Adaptation Sets conforming to this profile:
 - if either the **AdaptationSet.SegmentList** or the **AdaptationSet.SegmentTemplate** element is present in an **AdaptationSet** element then this **AdaptationSet** element may be ignored;
 - if either the **Representation.SegmentList** or the **Representation.SegmentTemplate** element is present in a **Representation** element then this **Representation** element may be ignored;
 - if the **Representation** element does not contain a **BaseURL** element then this **Representation** element may be ignored;
 - **AdaptationSet** elements with **AdaptationSet**@subsegmentAlignment not present or set to 'false' may be ignored;
 - **Representation** elements with a @subsegmentStartsWithSAP value absent, zero or greater than 3 may be ignored;
 - **Representation** elements with @subsegmentStartsWithSAP value equal to 3 may be ignored if both the following conditions hold:
 - the containing Adaptation Set contains more than one Representation, and
 - no other Representation has the same value for @mediaStreamStructureId.

- Elements using the `@xlink:href` attribute may be ignored from the MPD. The Representations conforming to this profile are those not accessed through an Adaptation Set that uses an `@xlink:href`.

8.3.3 Segment format constraints

For Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints shall be met:

- Representations shall comply with the formats defined in subclause 7.3, referring to the Segment formats in subclause 6.3.
- Each Representation shall have one Segment that complies with the Self-Initializing Media Segment as defined in subclause 6.3.5.2.
- All Segment Index ('`sidx`') and Subsegment Index ('`ssix`') boxes shall be placed before any Movie Fragment ('`moof`') boxes.
- Index Segments shall not be present. However, a **RepresentationIndex** element or a `@indexRange` attribute may be present to signal the byte range for Segment Index within a Media Segment.

8.4 ISO Base media file format live profile

8.4.1 General

This profile is optimized for live encoding and may achieve latency of a few seconds by encoding and immediate delivery of short Segments consisting of one or more movie fragments of ISO file format, typically with relatively short duration. Each movie fragment may be requested as soon as available using a template generated URL, so it is not normally necessary to request an MPD update prior to each Segment request. Segments are constrained so that accessing Representations at Segment boundaries is enabled and seamless switching within one Adaptation Set may be performed by first processing (i.e. downloading, decoding and presenting) the come-from Representations and then processing the go-to Representation. Although the profile is optimized for live services, the **MPD**`@type` attribute may be set to '`static`' to distribute non-live content, for example in case a live Media Presentation is terminated, but kept available as On-Demand service.

The ISO Live profile is identified by the URN "`urn:mpeg:dash:profile:isoff-live:2011`".

8.4.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and segments as defined in subclause 7.3 shall apply.
- The elements and attributes listed in subclause 5.2.3.2 may be ignored.
- Representations not inferred to have `@profiles` equal to the profile identifier as defined in subclause 8.4.1 may be ignored.
- In addition, **Representation** elements contained in an **AdaptationSet** element complying to this profile shall have the following constraints:
 - **Representation** elements with `@startWithSAP` value (either supplied directly or inherited from the containing **AdaptationSet**) equal to 3 may be ignored if both the following conditions hold:

- the containing Adaptation Set contains more than one Representation, and
- no other Representation has the same value for @mediaStreamStructureId.
- The **SegmentTemplate** element shall be present on at least one of the three levels, the Period level containing the Representation, the Adaptation Set containing the Representation, or on Representation level itself.
- **Representation** elements with a @startWithSAP value (either supplied directly or inherited from the containing) absent, zero or greater than 3 may be ignored.
- **AdaptationSet** elements with a @segmentAlignment value 'false' or absent may be ignored.
- **Representation** elements with @startWithSAP value (either supplied directly or inherited from the containing Adaptation Set) equal to 3 may be ignored if both of the following conditions hold:
 - the containing Adaptation Set contains more than one Representation, and
 - no other Representation has the same value for @mediaStreamStructureId.
- **Subset** elements may be ignored.
- Elements using the @xlink:href attribute may be ignored from the MPD. The Representations conforming to this profile are those not accessed through an Adaptation Set that uses an @xlink:href.
- When the MPD is updated, the value of **MPD**@availabilityStartTime shall be the same in the original and the updated MPD.

8.4.3 Segment format constraints

For Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints shall be met:

- Representations shall comply with the formats defined in subclause 7.3, referring to the Segment formats in subclause 6.3.
- Each Representation shall have one Initialization Segment and at least one Media Segment.
- Media Segments containing multiple Media Components shall comply with the formats defined in subclause 6.3.4.3, i.e. the brand 'msix'.
- In Media Segments, all Segment Index ('sidx') and Subsegment Index ('ssix') boxes shall be placed before any Movie Fragment ('moof') boxes.
- Index Segments shall not be present.

8.5 ISO Base media file format main profile

8.5.1 General

This profile is identified by the URN "urn:mpeg:dash:profile:isoff-main:2011".

8.5.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and segments as defined in subclause 7.3 shall apply.
- The elements and attributes listed in subclause 5.2.3.2 may be ignored.
- Representations not inferred to have @profiles equal to the profile identifier as defined in subclause 8.5.1 may be ignored.
- The **Subset** element may be ignored.
- Elements using the @xlink:href attribute may be ignored from the MPD. The Representations conforming to this profile are those not accessed through an Adaptation Set that uses an @xlink:href.
- For Adaptation Sets conforming to this profile:
 - **Representation** elements with a @startWithSAP value greater than 3 or contained in an **AdaptationSet** element with @subsegmentStartsWithSAP value greater than 3 may be ignored.
 - If **MPD**@type is 'dynamic',
 - **AdaptationSet** elements with a @segmentAlignment value 'false' or absent may be ignored;
 - **Representation** elements with a @startWithSAP value (either supplied directly or inherited from the containing **AdaptationSet**) absent or zero may be ignored.
 - **Representation** elements with @startWithSAP value (either supplied directly or inherited from the containing **AdaptationSet**) equal to 3 may be ignored if both the following conditions hold:
 - the containing Adaptation Set contains more than one Representation, and
 - no other Representation has the same value for @mediaStreamStructureId.

8.5.3 Segment format constraints

For Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints shall be met:

- Representations shall comply with the formats defined in subclause 7.3, referring to the Segment formats in subclause 6.3.
- At least one SAP of type 1 to 3, inclusive, shall be present for each track in each Subsegment.
- In Media Segments, all Segment Index ('sidx') and Subsegment Index ('ssix') boxes shall be placed before any Movie Fragment ('moof') boxes.

- Each Media Segment of the Representations not having `@startWithSAP` present or having `@startWithSAP` value 0 or greater than 3 shall comply with the formats defined in subclause 6.3.4.3, i.e. the brand 'msix'.

8.6 MPEG-2 TS main profile

8.6.1 General

This profile imposes little constraint on the Media Segment format for MPEG-2 Transport Stream content. This profile is identified by the URN "urn:mpeg:dash:profile:mp2t-main:2011".

8.6.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD as defined in subclause 7.4 shall apply.
- The elements and attributes listed in subclause 5.2.3.2 may be ignored.
- Representations not complying with the restrictions defined in subclause 7.4 or not inferred to have `@profiles` equal to the profile identifier as defined in subclause 8.6.1 may be ignored.
- Representations not in group 0 may be ignored.
- **Subset** may be ignored.
- Representations containing the **SegmentTimeline** element may be ignored.
- It shall be possible to present a presentation conforming to this profile without resolving `@xlink:href` in **AdaptationSet** or **SegmentList** elements. Any initial **Period** elements using `@xlink:href` may be ignored, and the first non-excluded Period shall have an explicit `@start` attribute. After the first non-excluded Period, there shall be no Period using `@xlink:href`.
- When the MPD is updated, the value of **MPD**`@availabilityStartTime` shall be the same in the original and the updated MPD.

8.6.3 Segment format constraints

For Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints shall be met:

- Representations shall comply with the formats defined in subclause 7.4, referring to the Segment formats in subclause 6.4.

8.6.4 Comments and recommendations

The following may be used, if desired:

- Representations not complying with the restrictions defined in subclause 7.4 may still be present, but the presentation should be presentable if they are ignored;
- Both **SegmentTemplate** or **SegmentList** elements may be used; the normal case is the use of **SegmentList** elements; however, clients should be capable of handling **SegmentTemplate** elements.

For Representations conforming to this profile:

- Index Segments should be supplied.
- **AdaptationSet** elements containing Representations conforming to this profile should not set the value of the `@segmentAlignment` attribute (either supplied directly or inherited from the containing **MPD**) to `'false'`.
- Representations conforming to this profile should set the value of the `@startWithSAP` to 1 or 2. `@startWithSAP` may be set to 3 if `@mediaStreamStructureId` is identical across Representations.

8.7 MPEG-2 TS simple profile

8.7.1 General

This profile is a subset of MPEG-2 TS main profile as defined in subclause 8.6. It poses more restrictions on content encoding and multiplexing in order to allow simple implementation of seamless switching. This is achieved by guaranteeing that a media engine conforming to ISO/IEC 13818-1 can play any bitstream generated by concatenation of consecutive segments from any Representation within the same Adaptation Set.

This profile is identified by the URN `"urn:mpeg:dash:profile:mp2t-simple:2011"`.

8.7.2 Media Presentation Description constraints

The Media Presentation Description shall conform to the following constraints:

- All MPD constraints of MPEG-2 TS Main Profile as defined in subclause 8.6.2 shall be obeyed.
- The elements and attributes listed in subclause 5.2.3.2 may be ignored.
- Representations not complying with the restrictions defined in subclause 7.4 or not inferred to have `@profiles` equal to the profile identifier as defined in subclause 8.7.1 may be ignored.
- If an Index Segment is provided, any Adaptation Set with `@subsegmentAlignment` set to `'false'` may be ignored.
- Any Adaptation Set which contains more than one Representation and has `@bitstreamSwitching` not set to `'true'` may be ignored.
- When the MPD is updated, the value of **MPD**`@availabilityStartTime` shall be the same in the original and the updated MPD.

8.7.3 Segment format constraints

For Representations and Segments referred to by the Representations in the profile-specific MPD for this profile, the following constraints shall be met:

- Representations shall comply with the formats defined in subclause 7.4, referring to the Segment formats in subclause 6.4.
- All Media Segment constraints of MPEG-2 TS main profile as defined in subclause 8.6.3 shall be obeyed.
- PSI information, including versions, shall be identical within all Representations contained in an AdaptationSet.

- If MPEG-2 Conditional Access framework is used, the same ECM shall be valid for the whole Subsegment, or for the whole Segment if Index Segment is not present.
- For an Index Segment, any single Segment Index ('sidx') box may either reference media, or other 'sidx', but the same 'sidx' box may not reference both.

8.7.4 Recommendations

For Representations conforming to this profile, it is recommended that:

- Index Segments be supplied,
- **SegmentTemplate** elements be used.

8.8 ISO Base media file format extended live profile

8.8.1 General

This profile is largely an extension of ISO BMFF Live profile as described in subclause 8.4 of this document. The main extensions are non-exclusion of remote elements and features introduced in the second edition of this document, such as events.

This profile also imposes additional restrictions on MPD and Segment format in order to simplify implementations.

The ISO-Base media file format extended live profile is identified by the following URN: "urn:mpeg:dash:profile:isoff-ext-live:2014".

8.8.2 Media Presentation Description constraints

8.8.2.1 General

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and the Segments as defined in subclause 7.3 shall apply.
- Periods which do not conform to the constraints in subclause 8.8.2.2 may not be presented.
- Representations not inferred to have @profiles equal to the profile identifier as defined in subclause 8.8.1 may be ignored.

8.8.2.2 Constraints on Period elements

- The **Subset** element may be ignored.
- The **Period.SegmentList** element shall not be present.
- If a Period contains multiple Adaptation Sets with @contentType="video" then at least one Adaptation Set shall contain a Role element with @schemeIdUri="urn:mpeg:dash:role:2011" and @value="main" and each Adaptation Set containing such a **Role** element shall provide perceptually equivalent media streams.
- **AdaptationSet** elements that do not conform to subclause 8.8.2.3 may be ignored.

8.8.2.3 Constraints on AdaptationSet elements

- **AdaptationSet** element can be ignored unless **AdaptationSet.SegmentTemplate** is present and/or for each Representation within this Adaptation Set **Representation.SegmentTemplate** element is present;
- **AdaptationSet** element that contains more than one Representation can be ignored unless all of the following hold:
 - **AdaptationSet@SegmentAlignment** is present and has value of 'true';
 - **AdaptationSet@startsWithSAP** is present and has value of 1 or 2;
- **Representation** elements that do not conform to subclause 8.8.2.4 may be ignored.

8.8.2.4 Constraints on Representation elements

- Representations with value of the @mimeType attribute other than video/mp4, audio/mp4, application/mp4, or text/mp4 may be ignored. Additional profile or codec specific parameters may be added to the value of the MIME type attribute.
- If **Representation.InbandEventStream** or **SubRepresentation.InbandEventStream** are present, this Representation can be ignored.

8.8.3 Segment format constraints

Representations and Segments complying to this profile shall meet the following constraints:

- Representations shall comply with the formats defined in subclause 7.3.
- In Media Segments, all Segment Index ('sidx') and Subsegment Index ('ssix') boxes, if present, shall be placed before any Movie Fragment ('moof') boxes.
- Index Segments shall not be present.

8.8.4 Inband Events

If an **AdaptationSet** element inferred to have this profile within contains an **InbandEventStream** element, and **InbandEventStream@schemeIdUri** has value "urn:mpeg:dash:event:2012", all representations within this adaptation set shall contain aligned inband events.

NOTE 1 MPD validity expiration inband events (see subclause 5.10.4.2) are essential for correct presentation of content formatted for the ISO BMFF Extended Live profile.

NOTE 2 The author can assume that, for each value of **MPD@publishTime** they announce using MPD validity expiration event(s), the DASH Client receives and processes at least one Event Message ('emsg') box with this value in course of normal playback of this Period.

8.9 ISO Base media file format extended On Demand profile

8.9.1 General

This profile is largely an extension of ISO BMFF On Demand profile as described in subclause 8.3. The main extensions are non-exclusion of remote elements and features introduced in the second edition of this document.

This profile also imposes additional restrictions on MPD and Segment format in order to simplify implementations.

The ISO-Base media file format extended On Demand profile is identified by the following URN: "urn:mpeg:dash:profile:isoff-ext-on-demand:2014".

8.9.2 Media Presentation Description constraints

8.9.2.1 General

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and the Segments as defined in subclause 7.3 shall apply.
- Periods which do not conform to the constraints in subclause 8.9.2.2 may not be presented.
- Representations not inferred to have @profiles equal to the profile identifier as defined in subclause 8.9.1 may be ignored.
- **MPD**@type shall be "static".

8.9.2.2 Constraints on Period elements

- The **Subset** element may be ignored.
- Neither the **Period.SegmentList** element nor the **Period.SegmentTemplate** element shall be present.
- If a Period contains multiple Adaptation Sets with @contentType="video" then at least one Adaptation Set shall contain a Role element with @schemeIdUri="urn:mpeg:dash:role:2011" and @value="main" and each Adaptation Set containing such a **Role** element shall provide perceptually equivalent media streams.
- **AdaptationSet** elements that do not conform to subclause 8.9.2.3 may be ignored.

8.9.2.3 Constraints on AdaptationSet elements

- **AdaptationSet** element can be ignored unless for each Representation within this Adaptation Set **Representation.BaseURL** is present.
- If either the **AdaptationSet.SegmentList** or the **AdaptationSet.SegmentTemplate** element is present in an **AdaptationSet** element then this **AdaptationSet** element may be ignored.
- If an **AdaptationSet** element contains more than one **Representation** element, then this **AdaptationSet** element can be ignored unless **AdaptationSet@subsegmentAlignment** is present and has value of 'true'; and **AdaptationSet@subsegmentStartsWithSAP** is present and has value of 1 or 2.
- **Representation** elements that do not conform to subclause 8.9.2.4 may be ignored.

8.9.2.4 Constraints on Representation elements

- Representations with value of the @mimeType attribute other than video/mp4, audio/mp4, application/mp4, or text/mp4 may be ignored. Additional profile or codec specific parameters may be added to the value of the MIME type attribute.
- If either the **Representation.SegmentList** or the **Representation.SegmentTemplate** element is present in a **Representation** element then this **Representation** element may be ignored.
- If the **Representation** element does not contain a **BaseURL** element then this **Representation** element may be ignored.
- If Representation consists of a single Segment that complies with Indexed Media Segment or Indexed Self-Initializing Media Segment, this **Representation** element can be ignored unless **SegmentBase@indexRange** is present.

8.9.3 Segment format constraints

Representations and Segments complying to this profile shall meet the following constraints:

- Representations shall comply with the formats defined in subclause 7.3, referring to the Segment formats in subclause 6.3.
- Each Representation shall have one Segment that either (i) complies with the Indexed Self-Initializing Media Segment as defined in subclause 6.3.5.2 or (ii) complies with the Self-Initializing Media Segment as defined in subclause 6.3.5.1 and the Index Segment is present.
- For Indexed Self-Initializing Media Segments all Segment Index ('sidx') and Subsegment Index ('ssix') boxes shall be placed before any Movie Fragment ('moof') boxes.
- Event Message ('emsg') boxes shall not be present.

8.10 ISO Base media file format common profile

8.10.1 General

This profile is a restricted combination of both extended profiles described in 8.8 and 8.9. Use of this profile implies that one can have a mix of two profiles in a single MPD, but not within the same Period.

The ISO-Base media file format common profile is identified by the following URN: "urn:mpeg:dash:profile:isoff-common:2014".

8.10.2 Media Presentation Description constraints

8.10.2.1 General

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD and the Segments as defined in subclause 7.3 shall apply.
- Periods which do not conform to the constraints in subclause 8.10.2.2 may not be presented.
- **MPD@profiles** shall contain "urn:mpeg:dash:profile:isoff-common:2014".

8.10.2.2 Constraints on Period elements

- Each Period shall conform either to constraints in subclause 8.8.2.2 or to constraints in subclause 8.9.2.2.

8.10.3 Segment format constraints

Segments referred from Periods complying to constraints in subclause 8.8.2.2 shall conform to subclause 8.8.3.

Segments referred from Periods complying to constraints in subclause 8.9.2.2 shall conform to subclause 8.9.3.

8.11 ISO Base media file format broadcast TV profile

8.11.1 General

This profile provides a restricted profile primarily for distributing broadcast TV over broadcast and broadband services, including service offerings for combined unicast and broadcast services. The profile is based on ISO BMFF. In order to enable those advanced use cases, this profile introduces the main restrictions that follow compared to the extended live profile:

- Use a single `@timescale` for all Representations in one Adaptation Set.
- Use Segment Timeline for signalling of segment durations:
 - The timing of the segments in the MPD is accurate.
 - The Segment Timeline may be on Representation level to allow different segment durations in different Representations. However, it may be defaulted on Adaptation Set level.
 - The Segment Timeline may use open ended `@r (-1)` or closed `@r (>=0)`.
 - The Segment Timeline may use Segment sequences and Hierarchical Templating.
- Each Representation shall provide at least one **RandomAccess** element.
- If an Adaptation contains more than one Representation, then at least one **Switching** element shall be present.
- Segment alignment and start with SAP signalling may be used for backward compatible deployments but should generally not be used.
- Data URLs as defined in IETF RFC 2397 may be used for Initialization Segments.

The ISO-Base media file format broadcast TV profile is identified by the following URN: "urn:mpeg:dash:profile:isoff-broadcast:2015".

8.11.2 Media Presentation Description constraints

8.11.2.1 General

The Media Presentation Description shall conform to the following constraints:

- The rules for the MPD as defined in subclause 7.3 shall apply.
- The rules for the Segments as defined in subclause 7.3.5 shall apply.

- Periods which do not conform to the constraints in subclause 8.11.2.2 may not be presented
- Representations not inferred to have `@profiles` equal to the profile identifier as defined in subclause 8.11.1 may be ignored

8.11.2.2 Constraints on Period elements

- The **Subset** element may be ignored.
- The **Period.SegmentList** element shall not be present.
- **AdaptationSet** elements that do not conform to subclause 8.11.2.3 may be ignored.

8.11.2.3 Constraints on AdaptationSet elements

- **AdaptationSet** element may be ignored unless **AdaptationSet.SegmentTemplate** is present and/or for each Representation within this Adaptation Set **Representation.SegmentTemplate** element is present;
- **AdaptationSet** element may be ignored unless **AdaptationSet.RandomAccess** is present and/or for each Representation within this Adaptation Set **Representation.RandomAccess** element is present;
- **AdaptationSet** element that contains more than one Representation may be ignored unless **AdaptationSet.Switching** is present and/or for each Representation within this Adaptation Set **Representation.Switching** element is present and all the **SegmentTemplate** elements conform to subclause 8.11.2.5;
- **InBandEventStream** shall only be used on Adaptation Set level;
- **Representation** elements that do not conform to subclause 8.11.2.4 may be ignored.

8.11.2.4 Constraints on Representation elements

- Representations with value of the `@mimeType` attribute other than `video/mp4`, `audio/mp4`, `application/mp4`, or `text/mp4` may be ignored. Additional profile or codec specific parameters may be added to the value of the MIME type attribute.
- **Representation** elements may be ignored if **Representation.RandomAccess** element is not present and also no **AdaptationSet.RandomAccess** element is present.
- **InBandEventStream** shall not be present on Representation level.
- Segment Timeline shall be used for signalling of segment durations and the following restrictions shall apply:
 - The timing of the segments in the MPD shall be accurate.
 - The Segment Timeline may be open ended `@r (-1)` or closed `@r (>=0)`.
 - The Segment Timeline may contain Segment Sequences as defined in subclause 5.3.9.6.4 and Hierarchical Templating as defined in subclause 5.3.9.6.5.

- The Segment Timeline may be on Representation level to allow different segment durations in different Representations. However, it may be defaulted on Adaptation Set level.

8.11.2.5 Constraints on SegmentTemplate elements

- @initialization attribute may include data URLs as defined in IETF RFC 2397.

8.11.3 Segment format constraints

Representations and Segments complying with this profile shall meet the following constraints:

- Representations shall comply with the formats defined in subclause 7.3.5.
- If Segment Sequences as defined in subclause 5.3.9.6.4 and Hierarchical Templating as defined in subclause 5.3.9.6.5 are used, then the first Segment of a Segment Sequence shall not carry 'dums' brand in the Segment Type box ('styp') as major brand and all other Segments of the Segment Sequence shall carry 'dums' brand in the Segment Type box ('styp') as major brand.

8.11.4 MPD Updates and Inband Event Streams

In order for a DASH Client to operate without frequent MPD requests and use the information contained in Inband Event Streams, the content authoring needs to obey certain rules.

In case of **MPD@type="dynamic"** and the MPD indicates that one or several Representation(s) contain an inband event stream in order to signal MPD validity expirations, then the following applies:

- The **MPD@publishTime** shall be present.
- The **MPD@minimumUpdatePeriod** should be set to a small number, preferably 0.
- For each newly published MPD that includes changes that are not restricted to any of the following (e.g. a new Period):
 - The value of the **MPD@minimumUpdatePeriod** is changed,
 - The value of a **SegmentTimeline.S@r** has changed,
 - A new **SegmentTimeline.S** element is added, and
 - Any information that is no longer in the availability time window.

The following shall be done:

- a new MPD shall be published with a new publish time **MPD@publishTime**;
- an 'emsg' box shall be added to each segment of each Representation that contains an **InbandEventStream** element with
 - `scheme_id_uri="urn:mpeg:dash:event:2012"`,
 - @value either set to 1 or set to 3,
 - the value of the **MPD@publishTime** of the previous MPD as the `message_data`.

8.12 DASH profile for CMAF content

8.12.1 General

ISO/IEC 23000-19 defines the common media application format for segmented media. The CMAF addressable media objects, also known as resources, may be delivered in a DASH Media Presentation.

Common Media Application Format (CMAF) specifies content formats independently of any manifest format and defines the structural format of CMAF Tracks and relationships between these tracks. CMAF is built on the ISO Base Media File Format (BMFF) as defined in ISO/IEC 14496-12 and its Segments follow the ISO BMFF Segment constraints as defined in subclause 6.2.

In order to distribute CMAF content in DASH, two profiles are defined in the following, providing a prescriptive mapping of CMAF structures and timelines to DASH structures and timelines. Applying this mapping enables content to conform to both CMAF constraints and DASH requirements as well as to distribute CMAF generated content with DASH.

Two profiles are defined:

- The DASH core profile for CMAF Content is defined in subclause 8.12.5. This profile addresses a restrictive mapping of CMAF Presentation timelines to DASH Media Presentations.
- The DASH extended profile for CMAF Content is defined in subclause 8.12.6. This profile is more permissive on timeline mapping and addresses cases for which splicing of content from different sources need to happen. In this case it is not always practical and feasible to map a CMAF Presentation exactly on a DASH Period.

In order to introduce the CMAF Mapping to DASH, a CMAF content model introduced in subclause 8.12.2 and high-level mapping considerations are provided in subclause 8.12.3. Subclause 8.12.4 provides common constraints for the DASH profile for media content. Finally, some conformance checking considerations are provided in subclause 8.12.6.

Example MPDs for this profile are provided in Annex G.

While the mappings defined by these profiles may not be the only possible mapping of CMAF content to DASH, it is intended to enable the distribution of CMAF generated content in DASH in a consistent manner.

The profiles may also be used to provide a simple and well-defined manifest format for CMAF content, for example for the exchange of CMAF. Such an MPD may also be used to describe the relationship of stored CMAF Resources.

8.12.2 CMAF content model for DASH profile definition

8.12.2.1 Overview

This subclause introduces the main features of CMAF and defines the relevant terms in order to map CMAF content to DASH Segments and provide the relevant MPD information.

The CMAF content model is shown in Figure 7 with a very principle mapping to DASH.

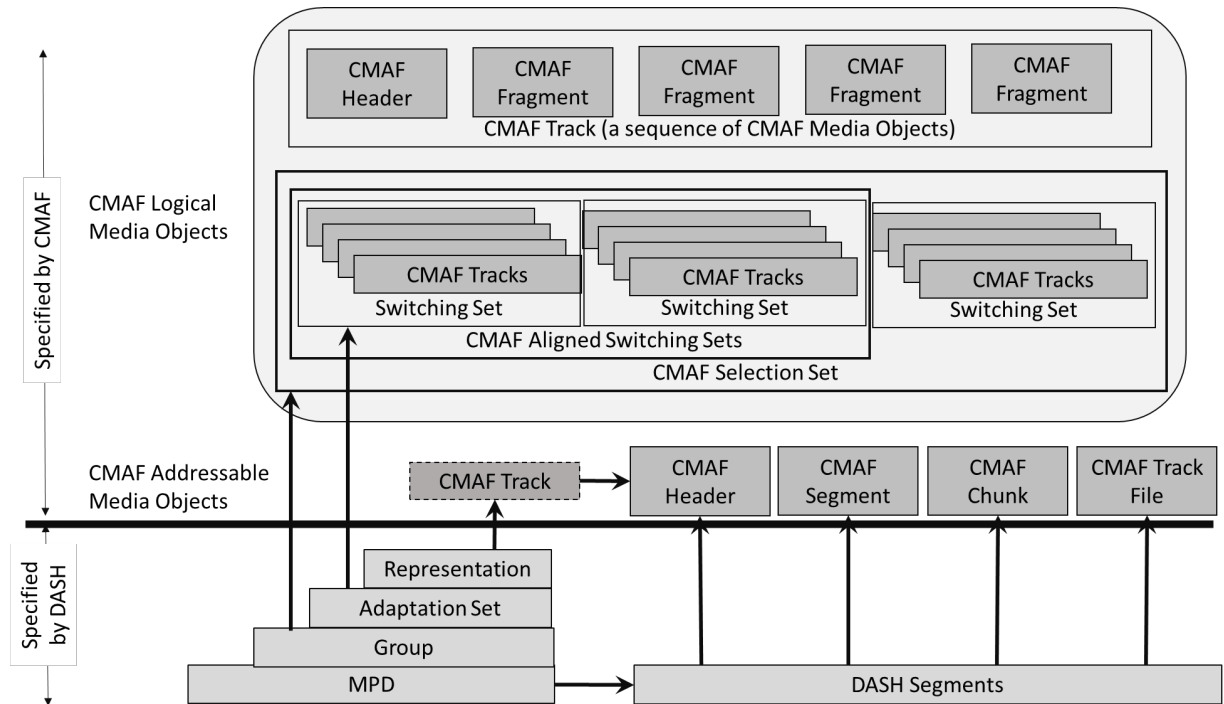


Figure 7 — CMAF Content Model

8.12.2.2 CMAF Addressable Objects

CMAF, as defined in ISO/IEC 23000-19, provides the following addressable objects:

- CMAF Track File
- CMAF Chunk
- CMAF Segment

For details, please refer to the CMAF specification in ISO/IEC 23000-19.

CMAF Fragments and CMAF Chunks may be embedded in DASH Segments. In the context of this standard, DASH Segments are the addressable units, i.e. units with an assigned URL.

Note that in practical applications,

- CMAF Track Files are analogous to DASH Self-Initializing Media Segments
- CMAF Headers are analogous to DASH Initialization Segments
- CMAF Segments (consisting of one or more CMAF Fragments or CMAF Chunks) are analogous to DASH Media Segments.

8.12.2.3 CMAF Track Data Model

In the context of this document, the following definitions are relevant for CMAF Tracks as defined in ISO/IEC 23000-19. For each CMAF Track k ($k=1, \dots, K$) in a CMAF Switching Set, the following features are defined:

- CMAF Header $CH[k]$, $k=1, \dots, K$
- CMAF Fragments $CF[k, i]$, $i = 1, 2, 3, \dots, N$, with

- The position in the CMAF track i
- The earliest presentation time: $t_f[k, i]$
- The CMAF Fragment duration: $df[k, i] = t_f[k, i+1] - t_f[k, i]$
- Optionally, wall-clock time assigned to the earliest presentation time of CMAF Fragment: $t_{wc}[k, i]$
- One or possibly more CMAF Chunks $CC[k, i, j]$, $j = 1, 2, 3, \dots, C[i]$
 - Number of chunks in a fragment $C[i]$
 - Position in the fragment j
 - Earliest decode time $t_c[k, i, j]$
 - Chunk duration in decode times $dc[k, i, j]$
- The earliest presentation time of the first fragment, i.e. $t_f[k, i=1]$
- The duration of the CMAF Track is defined as $dt[k]$
- The CMAF Track has an assigned media profile, which includes:
 - CMAF media profile brand
 - Suitable MIME Type string providing
 - Media type
 - Codecs parameter
 - Profiles parameter
- The CMAF Track has samples $sample[k, s]$ with $s=1, \dots, S$, each with presentation time $T[k, s]$.

8.12.2.4 CMAF Switching Set Model for this Specification

From the constraints on a CMAF Switching Set as defined in ISO/IEC 23000-19, the following holds:

- All CMAF Tracks in a Switching Set conform to one media profile.
- One CMAF header is identified to be the "CMAF Principal Header" which has the highest decoding requirements and as such can initialize a media pipeline for decoding any other track in the Switching Set. This header referred as CMAF Principal Header CH*.
- The CMAF Header for each CMAF Track in a Switching Set is defined such that appending it to the track buffer does not result in a reinitialization of the decoding and rendering pipeline.
- Each CMAF Track in a Switching Set has the same number of CMAF Fragments.

- The earliest decoding times of the CMAF Fragments at a position i in different CMAF Tracks of a CMAF Switching Set are identical.
- The earliest presentation times of the CMAF Fragments at a position i in different CMAF Tracks of a CMAF Switching Set are identical, ie the value of $tf[k, i]$ is the same for all k in the switching set.
- The fragment duration of the CMAF Fragments at a position i in different CMAF Tracks of a CMAF Switching Set are identical, ie the value of $df[k, i]$ is the same for all k in the switching set.
- The duration of each CMAF track $dt[k]$ is identical and defines the duration of the CMAF Switching Set dt .

Note that the equalities above only hold for CMAF Fragments, not necessarily for CMAF Chunks.

8.12.2.5 CMAF presentation timing model

There are multiple timelines involved in the authoring, playout, and rendering CMAF tracks within a presentation:

- Each track is a sequence of timed samples $sample[k, s]$ with $s=1, \dots, S$, each with presentation time $T[k, s]$. Each sample has a decode time and duration and may also have a composition (display) time offset. Edit lists may be used to override the implicit direct mapping of the media timeline, into the timeline of the overall movie. The movie timeline is used to synchronize CMAF Tracks in a CMAF presentation and also serves as the synchronization source for playback – in a DASH player, or in an HTML 5 media element and media source.
- Movie timelines are aligned between all CMAF Tracks in a CMAF presentation, but in contrast to ISO BMFF files the isn't required to be at a presentation time 0.
- In addition, each CMAF Track may have assigned an anchor wall-clock time (e.g. UTC time). The wall clock time may be used to relate a presentation time of the track to wall-clock time, for example expressing the time when the corresponding sample was captured, encoded, or packaged.

8.12.3 CMAF to DASH Mapping Principles

8.12.3.1 General

This subclause documents some high-level considerations on how to map and deliver CMAF Presentations in DASH. It addresses both time mapping and structural mapping.

8.12.3.2 Timing Model Mapping

The CMAF mapping principles and timing model mapping for DASH and ISO BMFF is explained in ISO/IEC 23000-19. References are provided in the following and their mapping to DASH is explained.

In the context of this profile, the CMAF presentation time equal to zero as defined in ISO/IEC 23000-19 is mapped to the start of a DASH Period. As a consequence of this,

- the start of a Period is also aligned with the earliest video media sample presentation time.
- Each CMAF Track is mapped to a DASH Representation r .
- The presentation time offset in DASH is identical to the earliest video media sample presentation time. The earliest video media sample presentation time is derived as the movie time of the track according to regular file format principles taking into account the baseMediaDecodeTime and, if

present, the composition offset, and the track edit list (in the track header). Note that the earliest presentation time of a CMAF Fragment may not be the presentation time of the first sample in decode time of the CMAF Fragment.

- The presentation time offset is mapped to a DASH MPD by the value of the `@presentationTimeOffset`.
- The mapping of non-video tracks to DASH Periods is such that the media samples in audio and subtitle CMAF tracks whose durations overlap the earliest video media sample presentation time will only be partially presented, starting at the earliest video media sample presentation time.
- To maintain synchronization of all tracks of a single CMAF Presentation within a DASH Period, the presentation time offset of every CMAF switching set from the same CMAF Presentation needs to be equivalent, which means equal presentation time offsets, but possibly different integer values and timescales per CMAF switching set. However, DASH permits to map multiple CMAF Presentations to one Period (for example a main content and an alternative content). Hence, CMAF Tracks in one Period belonging to different CMAF Presentations are preferably differentiated accordingly.
- For the mapping a CMAF Presentation to a DASH Period, the video track is assumed to be the primary track and hence, the Period duration is determined by the duration of the CMAF video track.
- In certain cases, only a time span of a CMAF Presentation may be mapped to single Period, or a CMAF Presentation is provided in multiple DASH Periods. In the context of this profile, a time span of a CMAF Presentation mapped to a DASH Period also needs to conform to a CMAF Presentation. For this to hold, only specific time spans of a CMAF Presentation can be mapped, namely
- The start of the Period needs to coincide with the start of a CMAF video fragment.
- The end of a Period needs to coincide with the end of a CMAF video fragment in case of the DASH CMAF core profile.
- Sequential and continuous playback of independent CMAF Presentations in DASH can be accomplished by each CMAF Presentation being mapped to a single Period and playing back a sequence of Periods. Specifically, it is assumed that the CMAF Presentations are played in sequence as follows:
- For any CMAF Presentation contained in a Period $p=1, \dots, P$, the Period start time $PST[p]$ reflects the actual time that elapses after playing the media of all prior CMAF Presentations/Periods in the sequence of these presentations relative to the $PST[1]$ of the first CMAF Presentation/Period in the sequence.
- The Period extends until the PST of the next Period/CMAF Presentation, or until the end of the sequenced presentation in the case of the last Period/CMAF Presentation.
- The difference between the $PST[p]$ of a Period/CMAF Presentation and the $PST[p+1]$ time of the following Period/CMAF Presentation, is the Period/CMAF Presentation duration $PD[p]$ in Media Presentation time of the media content represented, i.e. $PD[p] = PST[p+1] - PST[p]$.
- Period boundaries permit playback of content for timeline continuous CMAF Presentations or for timeline discontinuity. Continuity can be considered in different domains:

- Continuity in terms that the presentation time is continuous across Period boundaries. DASH permits both, timeline continuity and discontinuity.
- Continuity in terms of remaining in the Initialization of a single CMAF Switching Set/DASH Adaptation Set, or to terminate and re-initialize. DASH again permits both, continuation with the same Initialization or changing the Initialization.
- Whereas CMAF does not provide any specific support in cases where a decode time discontinuity, for example resulting from one or more missing or damaged CMAF fragments in a continuous CMAF presentation, DASH supports the signalling of such gaps and the maintenance of the timelines. For example, these can be split into multiple periods, or the DASH Segment Timeline functionality can be used.

8.12.3.3 Mapping of CMAF Structures

Different mappings of CMAF Presentations and structures to DASH MPDs may exist. This profile defines a consistent mapping of CMAF Presentations and structures to a DASH Media Presentation.

Specifically, the DASH profiles for CMAF content as defined in this subclause address and define the following:

- The mapping of a single CMAF Presentation to a single DASH Period.
- The mapping of multiple independent CMAF Presentations into a single DASH Period in order to allow the client to select a CMAF Presentation for playback, e.g. in case a main content and an alternative content is provided.
- The mapping of a single CMAF Switching Set to a single DASH Adaptation Set. This includes the signalling of CMAF Principal Header information in DASH MPD Adaptation Set parameters.
- The mapping of a single CMAF Track to a single DASH Representation. This includes the signalling of CMAF Header information in DASH MPD Representation parameters.
- The exact time mapping of a CMAF Presentation to a DASH Period and the relevant Period related MPD signalling parameters.
- The creation of multiple Periods from a single CMAF Presentation and the appropriate MPD and cross-Period signalling.
- The sequencing of multiple independent CMAF Presentations into a DASH Media Presentation for continuous playback and the relevant signalling.
- The insertion of an independent CMAF Presentation into the middle of another CMAF Presentation as part of DASH Media Presentation and the relevant signalling.
- Additional mapping considerations.

In the context of this profile, CMAF Switching Sets support seamless switching as defined in subclause 4.5. Seamless switching can be achieved in DASH operations as follows:

- 1) The content author provides the Representations A and B as referred in subclause 4.5 being conformant to a CMAF Track that is part of the same CMAF Switching Set.
- 2) The client can switch at the start of DASH Segment or Subsegment (as long as the associated SAP Type is 1 or 2), i.e. at presentation times that coincide with the start of a CMAF Fragment.

8.12.3.4 Mapping of CMAF Addressable Objects

CMAF Addressable Objects may be mapped directly to DASH Segments. The profile defined in this subclause provides a consistent mapping to CMAF Addressable Resources to DASH Segments.

Specifically,

- CMAF Headers are directly mapped to DASH Initialization Segments.
- CMAF Segments and CMAF Chunks are directly mapped to DASH Media Segments.
- CMAF Segments are directly mapped to DASH Media Segments.
- CMAF Chunks may mapped to DASH Media Segments. In addition, the presence of CMAF chunks in CMAF Segments may be signalled in the DASH MPD.
- CMAF Track files are directly mapped to DASH Self-Initializing Segments.

The profile also provides the information how the DASH Representation information (URLs, timing information) is obtained for CMAF Addressable Resources.

8.12.3.5 Mapping of Media, Encryption and CMAF Header Information to MPD

CMAF introduces the concept of *media profiles*. A CMAF media profile is defined in ISO/IEC 23000-19. Based on the definitions and alignment defined in the DASH profiles for CMAF content in this subclause 8.12, it is considered sufficient a for a media codec to be delivered and integrated in DASH as long as a defines a CMAF media profile is defined and one of these defined profiles is used.

In order to integrate the media profile in DASH, for each media profile the essential and relevant information in a DASH MPD can be documented. This DASH profile defines some generic media profile independent requirements and recommendations for the mapping of CMAF Header information into the MPD.

For a media codec to be used together with the DASH profile for DASH-based profile, the following information needs to be provided:

- A CMAF media profile definition with all the requirements according to ISO/IEC 23000-19 for a media profile.
- The static mapping of parameters to a DASH MPD, in particular to the MPD parameters, such as @mimeType, @codecs, @containerProfiles, etc.
- The dynamic mapping of parameters to a DASH MPD from a CMAF Principal Header, in particular to the MPD parameters, such as @maxWidth, @maxHeight, etc.

Similar aspects apply for encryption and DRM information, for which an adequate mapping is provided.

8.12.4 DASH profiles for CMAF content - Common constraints

8.12.4.1 General

This subclause 8.12.4 defines detailed requirements and recommendations for the mapping of CMAF Presentations to DASH Media Presentations. According to the definition and usage of DASH profiles, the constraints may also apply to subsets of Media Presentations for example to Representations, Adaptation Sets or Periods.

In the remainder of the subclause 8.12.4, the constraints are summarized as *DASH profile for CMAF content constraints*.

8.12.4.2 Segment and Representation Constraints

If the *DASH profile for CMAF content constraints* apply to a DASH Representation, then the following holds for this Representation:

- Each Media Segment of the Representation shall conform to a CMAF Addressable Media Object as defined in ISO/IEC 23000-19.

NOTE 1 By conforming to a CMAF Addressable Media Object the Media Segment automatically also conforms to a DASH Delivery Unit Media Segment as defined in subclause 6.3.4.2.

- The Initialization Segment, if present, shall conform to a CMAF Header as defined in ISO/IEC 23000-19.

NOTE 2 By conforming to a CMAF Header the Initialization Segment automatically also conforms to a DASH Initialization Segment as defined in subclause 6.3.3.

- The Representation shall conform to a CMAF Track as defined in ISO/IEC 23000-19 as well as to a DASH Representation as defined in subclause 5.3.5.
- The `@timescale` shall be set to the timescale of the Media Header Box ('mdhd') of the CMAF track.
- If every DASH Segment conforms to CMAF Fragment constraints and if `@startWithSAP` is present, it shall be set to value 1 or 2.
- If the **SegmentTimeline** element is present for this Representation, then the following holds:
 - For every CMAF Fragment i , with $i = 1, 2, 3, \dots, N$, an entry in an **S** element shall be present with
 - `@t` set to earliest presentation time: $tf[k, i]$
 - `@d` set to CMAF Fragment duration: $df[k, i]$
 - If each CMAF Chunk is an addressable media object, then `@k` shall be set to the number of chunks in this CMAF Fragment $C[k, i]$ according to the Segment Sequence definition in subclause 5.3.9.6.4.
 - If consecutive CMAF Fragments have the same duration, their corresponding **S** element should be combined to a single **S** element.
- If the `@duration` attribute is present for this Representation and the value of the attribute is referred to as `dur`, then
 - the following shall hold for each CMAF Fragment i , with $i = 1, 2, 3, \dots, N$, the following shall hold:

$$((i-1)+0.5)*dur \leq (tf[k, i+1]-tf[k, 1]) \leq (i+0.5)*dur,$$
 - the Representation shall have assigned a `@startWithSAP` attribute with the value set to 1 or 2.
- If the Representation is a Self-Initializing Media Segment, then

- the Representation shall have assigned a `@subsegmentStartsWithSAP` attribute with the value set to 1 or 2.
- the Representation shall conform to a CMAF Track file;
- the Representation shall conform to the Indexed self-initializing Media Segment as defined in subclause 6.3.5.2 and exactly one Segment Index ('`sidx`') box shall be used and shall be placed before any Movie Fragment ('`moof`') boxes.
- The following applies for the Segment Index ('`sidx`')
 - The `reference_ID` shall be the `trackID` of the CMAF Track
 - The `timescale` shall be identical to the `timescale` of the Media Header Box ('`mdhd`') of the CMAF track.
 - the `earliest_presentation_time` shall be set to `tf[k,1]`.
 - `reference_count` shall be set to the number of CMAF Fragments, `N`, in the CMAF Track
 - Each CMAF Fragment shall be mapped to exactly one Subsegment as follows
 - The `reference_type` shall be set to 0.
 - The `referenced_size` is set to accordingly, with value being at least the size of the CMAF Fragment but may be larger if additional data is included in the track.
 - `starts_with_SAP` shall be set to 1.
 - The `subsegment_duration` is set to `df[k,i]`.
 - `SAP_type` shall be set to 1 or 2.
 - `SAP_delta_time` shall be set to 0.
- If multiple CMAF Chunks are present in Media Segments, Resynchronization Points should be signaled as defined in subclause 5.3.13.
- Event message streams may be signalled with **InbandEventStream** elements. If Event message streams are relevant for client processing, they should be signalled using the with **InbandEventStream** elements. For details on signalling and the placement of Event Message boxes, please refer to subclause 5.10.

8.12.4.3 Adaptation Set Constraints

If the *DASH profile for CMAF content constraints* apply to an Adaptation Set, then the following holds for this Adaptation Set:

- Each Representation shall conform to the Representation constraints as defined in 8.12.4.2.
- All Representations in the Adaptation Set shall conform to the CMAF Track requirements for a CMAF Switching Set as defined in ISO/IEC 23000-19.

- A CMAF Principal Header should be provided by one of the two following means:
- An `@initializationSetRef` attribute that references an Initialization Set that itself contains a `@initialization` attribute. The value of this attribute references a CMAF Principal Header for the Adaptation Set.
- An `@initializationPrincipal` attribute that references a CMAF Principal Header for the Adaptation Set.

NOTE 1 If no CMAF Principal Header is explicitly signalled, the Initialization Segment of the Representation with the highest value for the `@bandwidth` attribute can serve as CMAF Principal Header.

- The `@contentType` shall be set according to the `hdlr` type of the CMAF Principal Header of the Switching Set, i.e. to
 - `vide to video`,
 - `soun to audio`,
 - `subt to application`
 - `text to text`
- The `@mimeType` shall be set to "`<contentType>/mp4`".

NOTE 2 Setting `@mimeType` to "`<contentType>/mp4, profiles='cmfc'`" is correct, but can result that the media stream is not recognized. Hence, conformance to the CMAF profile is preferably documented by the `@containerProfiles` signalling.

- The `@containerProfiles` parameter should be present. If present, it shall include at least one profile string, namely a structural brand being either '`cmfc`' or '`cmf2`'. In addition, it should include a CMAF media profile brand.

NOTE 3 A CMAF Media profile brand is not required as conformance to a structural CMAF brand can be sufficient.

- The `@codecs` parameter shall be set to according to the sample entry codingname field of the CMAF Principal Header.

NOTE 4 By doing so, it is expected that this parameter is sufficient for capability exchange and media pipeline initialization. Representations can potentially signal different values for the `@codecs` parameter.

- `@segmentProfiles` may be present to signal the conformance to CMAF addressable media object types.
- If the content in an Adaptation Set is protected, the following applies:
 - the common encryption requirements in ISO/IEC 23000-19 shall apply,
 - the **ContentProtection** element shall be present and shall be set as follows:
 - A **ContentProtection** element with `@schemeIdUri` equal to "`urn:mpeg:dash:mp4protection:2011`" and `@value` equal to an encryption

scheme defined in ISO/IEC 23001-7. It may contain a `@cenc:default_KID` attribute. If present, the value of this attribute shall match the 'tenc' box `default_KID` value.

- At least one **ContentProtection** element with `@schemeIdUri` equal to a UUID, according to IETF RFC 4122, value that signals that content keys can be obtained with a DRM system identified by the UUID, according to IETF RFC 4122. It may contain a `@cenc:default_KID` attribute and a `cenc:pssh` element. If present, the values shall match the `tenc` box `default_KID` value and the `moov` box `pssh` value respectively.
- If the `@contentType` is `video`, then the following apply
 - Either `@width` and `@height` or `@maxWidth` and `@maxHeight` shall be present.
 - `@maxWidth` or `@width` shall be set to the width in the CMAF `TrackHeaderBox` of the CMAF Principal Header.
 - `@maxHeight` or `@height` shall be set to the height in the CMAF `TrackHeaderBox` of the CMAF Principal Header.

NOTE 5 The encoded/decoded picture size may be slightly larger than the one signalled in the above parameters.

NOTE 6 Using the maximum values instead of `@width` and `@height` allows to take into account that the Adaptation Set itself may have different encoded resolutions.

- `@frameRate` should be present. If present, it shall be set as follows:
 - If the values `moov/mvex/trex/default_sample_duration` is not set to 0 and the `moov/trak/mdia/mdhd/timescale` is present, then
 - Get `moov/mvex/trex/default_sample_duration` (e.g. 1001) from CMAF Principal Header
 - Get `moov/trak/mdia/mdhd/timescale` (e.g. 24000) from CMAF Principal Header
 - The value shall be set as `"timescale/default_sample_duration"` (e.g. 24000/1001)
 - else Media Profile specific settings may apply, for example in the Decoder Configuration Record such as the SPS data.
 - Media profile specific MPD settings may apply.
- If the `@contentType` is `audio`, then the following applies
 - `@lang` shall be set as follows:
 - If an `ExtendedLanguageBox (elng)` in the `MediaHeaderBox (mdia)` of the CMAF Principal Header is present in the `TrackBox`, then it is set to the content of the `extended_language` field

- otherwise, as the language field in the MediaHeaderBox of the CMAF Header is present, but neither of the following applies: (i) the language field in the MediaHeaderBox of the CMAF header is set to 'und' (undetermined), and (ii) the language field in the MediaHeaderBox of the CMAF header is set to 'mul' (multiple), it is set of the value of the language field.
 - Media Profile specific MPD settings may apply, for example the usage of Preselections if 'mul' is signalled and the available languages are signalled in the codec-specific AudioSampleEntry.
 - If the CMAF Principal Header contains one or more KindBox ('kind') in the UserDataBox ('udta') of the TrackBox with one or more instantiations where the schemeURI field is "urn:mpeg:dash:role:2011", then for each instantiation a **Role** descriptor should be present with @schemeIdURI field set to "urn:mpeg:dash:role:2011" and the @value field set to the value field of the instantiation.
- NOTE 7 There are cases where the Role was set in the encoded bitstream on a best knowledge basis, but when multiple encodings are combined in single MPD, the direct usage of the role defined in the media is not suitable in the MPD. In such cases the Role could be omitted or set differently.
- NOTE 8 The role could be set in the MPD even if not role is defined in the media. In order to avoid contradicting or non-matching Roles in the media and the MPD, the encoder is preferably not adding a role if it is unsure on an appropriate setting.
- If the @bitstreamSwitching is set to true for this Adaptation Set, then the included CMAF Switching Set shall follow the "CMAF switching set single initialization constraints" as defined in ISO/IEC 23000-19.
 - Either the @segmentAlignment or @subsegmentAlignment shall be set to true.
 - Event Message Streams in the CMAF Track may be signalled with **InbandEventStream** elements if processing by the receiving DASH client is expected.

8.12.4.4 Period Constraints

If the *DASH profile for CMAF content constraints* apply to a Period, then the following holds for this Period:

- A Period may contain one or multiple CMAF Presentations.
- NOTE 1 Multiple independent CMAF Presentations could be added into a single DASH Period in order to allow the client to select a CMAF Presentation for playback, e.g. in case a main content and an alternative content is provided.
- CMAF Presentations in one Period are differentiated by different DASH Subsets. Each CMAF Presentation is mapped to exactly one DASH Subset as defined in subclause 5.3.8. If the **Subset** element is not present, the Period contains exactly one CMAF Presentation.
- NOTE 2 DASH Subset signaling could be ignored according to certain DASH profiles. However, ignoring this signaling in the client does not impact the processing. Subset signaling is only necessary for consistently documenting CMAF Presentations from an MPD.
- Each Switching Set of each CMAF Presentation shall be mapped to exactly one Adaptation Set in the Period according to the *DASH profile for CMAF content constraints* for an Adaptation Set as defined in subclause 8.12.4.3.

- All CMAF Switching Sets/Adaptation Sets within one Subset shall conform to CMAF Presentation requirements as defined in ISO/IEC 23000-19 and shall have the same normalized value of the `@presentationTimeOffset`, *PTO*. The normalized value *PTO* refers to the division of the value of the `@presentationTimeOffset` with value of `@timescale` attribute.
- All Adaptation Sets set to the same integer value for `@segmentAlignment` or `@subsegmentAlignment` shall conform to aligned CMAF Switching Set constraints as defined in ISO/IEC 23000-19.
- For each of the CMAF Presentations (i.e. for each DASH Subset) mapped to a regular Period with *Period duration* signalled in the MPD as defined in subclause 5.3.2.1, the following applies:
 - The normalized value *PTO* (i.e., *PTO* divided by the timescale) defines the CMAF presentation time zero, i.e. the time of the CMAF Track's presentation time that is assigned to the start of the CMAF Presentation.
- For each *video* CMAF Switching Sets/Adaptation Sets, the following applies:
 - If the Period is not a *Partially Unavailable Period* as defined in subclause 5.3.2.5, then the normalized value of the earliest presentation time of each CMAF Track/Representation shall be identical to the value of *PTO* and shall coincide with the start of a CMAF Fragment. Specifically,
 - The value of `@eptDelta` shall be 0 and the attribute may be absent.
 - If the **SegmentTimeline** element is present for this Representation, then the nominal value of the first **S@t** element shall be identical to the value of *PTO*.
 - If the media is contained in a Self-Initializing Media Segment, the value of *PTO* and shall coincide with the start of a Subsegment.
- For each *non-video* CMAF Switching Sets/Adaptation Sets, the following applies:
 - The mapping of CMAF Switching Sets/Adaptation Sets to DASH Periods is such that the media samples in the contained CMAF Tracks which presentation time is earlier than the earliest presentation time of the video or later than the latest presentation time of the video are expected to not be presented, or only be partially presented, starting at the value of *PTO* and ending at the Period duration.
 - However, there are cases for which gaps at the start or end of the Period are permissible. Sufficient information in the MPD should be provided to identify such a potential overlap at the start or the end of the Period. Details are provided in subclause 8.12.4.5.
- Early terminated Periods as defined in subclause 5.3.2.1 are permitted but should be avoided in general and only be used for error cases. If used, the **Period@duration** shall signal the duration of the shortest CMAF Track in the Period.

8.12.4.5 Multi-Period and Media Presentation Constraints

If the *DASH profile for CMAF content constraints* apply to a Media Presentation, then the following holds:

- Each Period in the MPD shall conform to the Period constraints in subclause 8.12.4.4.

- If an Adaptation Set in Period i includes a period-connectivity or period-continuity signal as defined in subclause 5.2.3.4 for any preceding Period $j < i$ and an Adaptation Set with the same value of the **AdaptationSet@id** exists in this preceding Period j , then each Initialization Segment of the Adaptation Set with the same value of the **AdaptationSet@id** in the preceding Period shall be a compatible CMAF Header for a CMAF Track for CMAF Switching Set that is represented by the Adaptation Set that includes period-connectivity signalling.
- If an Adaptation Set in Period i includes a period-continuity signal as defined in subclause 5.2.3.4 for any preceding Period $j < i$ and an Adaptation Set with the same value of the **AdaptationSet@id** exists in this preceding Period j , and the Adaptation Set in Period j also includes a Representation with the same @id value as a Representation in the Adaptation Set in Period i , then the concatenation of the two Representations without the Initialization Segment of the Representation in Period P2 shall be a conforming CMAF Track.
- If a *non-video* Adaptation Set in Period i includes a period-continuity signal as defined in subclause 5.2.3.4 for the preceding Period $i-1$ and an Adaptation Set with the same value of the **AdaptationSet@id** exists in this preceding Period $i-1$ then
 - The value of @pdDelta in the Adaptation Set of the preceding Period $i-1$ shall be the identical to the value of @eptDelta in the Adaptation Set in the following Period i .
 - The absolute values for both attributes should be kept as small as possible.
 - Positive values of @eptDelta as well as negative values of @pdDelta shall be signalled in the MPD as they indicate a gap at the Period boundary. Negative values should be signalled as they provide an indication for an overlap at the Period boundary.

NOTE The setting of the values being positive or negative could be decided, for example, whether the Period is used as a splice-in or splice-out point.

else

- If the Period is not a *Partially Unavailable Period* as defined in subclause 5.3.2.5, then
 - if @duration signalling is used or a Self-Initializing Media Segment is provided, and the value of @eptDelta is not 0, then @eptDelta shall be present and shall be negative (indicating an overlap at the start of the Period).
 - if the **SegmentTimeline** element is present for this Representation, then the nominal value of the first **S@t** element shall not be larger than the value of *PTO*.
- If the Period is not a *Live-Edge Period* as defined in subclause 5.3.2.5, then
 - if @duration signalling is used or a Self-Initializing Media Segment is provided, the value of @pdDelta is not 0, then @pdDelta shall be present and shall be positive (indicating an overlap at the end of the Period).
 - If the **SegmentTimeline** element is present for this Representation and the @r attribute of the last **S** element is non-negative, and if the value of @pdDelta is different than be the difference between $@t + (@r-1) * @d$ of the last **S** element minus the value of the @presentationTimeOffset and the Period duration

normalized with the value of `@timescale`, then `@pdDelta` shall be present and shall be positive (indicating an overlap at the end of the Period).

8.12.5 DASH CMAF Core Profile

The DASH core profile for CMAF Content addresses a restrictive mapping of CMAF Presentation timelines to DASH Media Presentations.

The DASH core profile for CMAF Content is identified by the URN "urn:mpeg:dash:profile:cmf:2019". The `profiles` parameter may be present on different levels in the MPD.

For the DASH CMAF Core Profile, all requirements in clause 8.12.4 shall apply.

In addition, as the Period boundaries coincide with CMAF video tracks, the following restrictions apply:

- For each of the CMAF Presentations (i.e. for each Subset) mapped to a regular Period with *Period duration* signalled in the MPD as defined in subclause 5.3.2.1, the following applies:
- For each *video* CMAF Switching Sets/Adaptation Sets, the following applies:
 - If the Period is not a *Live-Edge Period* as defined in subclause 5.3.2.5, then the media time duration (normalized by the value of the `@timescale` attribute) of each CMAF Track/Representation shall be identical to the *Period duration*. Specifically,
 - The value of `@pdDelta` shall be 0 and the attribute may be absent.
 - If the **SegmentTimeline** element is present for this Representation and the `@r` attribute of the last **S** element is non-negative, then the value of `@pdDelta` shall be the difference between $@t + (@r - 1) * @d$ of the last **S** element minus the value of the `@presentationTimeOffset` and the Period duration normalized with the value of `@timescale`.
 - If the media is contained in a Self-Initializing Media Segment, the last included Subsegment shall extend exactly until *Period duration* normalized by the value of the `@timescale`.

8.12.6 DASH CMAF Extended Profile

The DASH extended profile for CMAF Content is more permissive on timeline mapping than the DASH CMAF Core Profile and addresses cases for which splicing of content from different sources need to happen. In this case it is not always practical and feasible to map a CMAF Presentation exactly on a DASH Period.

The DASH extended profile for CMAF Content is identified by the URN "urn:mpeg:dash:profile:cmf-extended:2019". The `profiles` parameter may be present on different levels in the MPD.

For the DASH CMAF Extended Profile, all requirements in clause 8.12.4 shall apply. No other restrictions apply.

8.12.7 Conformance Checking Considerations

If DASH content conforming to any of these CMAF media profiles is to be checked for compliance:

- The DASH content is checked against the format requirements for the defined DASH profile.

- In addition, the DASH content is also checked against CMAF conformance for the DASH CMAF Core Profile or DASH CMAF Extended Profile using the requirements as defined in subclause 8.12.5 and 8.12.6, respectively.

NOTE Content conforming to this profile is expected to conform to both, the DASH content constraints as well as the to the CMAF constraints.

Annex A (informative)

Example DASH Client behaviour

A.1 General

The information on client behaviour is purely informative and does not imply any normative procedures on DASH Client implementations. However, this information may serve as a guideline to better understand certain features of the formats in the normative parts of this document.

A.2 Overview

A DASH Client is guided by the information provided in the MPD. This example assumes that the **MPD@type** is 'dynamic'. The behaviour in case **MPD@type** being 'static' is basically a subset of the description here.

The description in this annex assumes that the client has access to the MPD at time *FetchTime*, at its initial location if no **MPD.Location** element is present, or at a location specified in any present **MPD.Location** element (see Annex A.11 for details). *FetchTime* is defined as the time at which the server processes the request for the MPD from the client. The client typically should not use the time at which it actually successfully received the MPD but should take into account delay due to MPD delivery and processing. The fetch is considered successful either if the client obtains an updated MPD or if the client verifies that the MPD has not been updated since the previous fetching.

The following example client behaviour may provide a continuous streaming experience to the user:

- 1) The client parses the MPD, selects a collection of Adaptation Sets suitable for its environment based on information provided in each of the **AdaptationSet** elements. The selection of Adaptation Sets may also take into account information provided by the **AdaptationSet@group** attribute and any constraints of a possibly present **Subset** element.
- 2) Within each Adaptation Set, it selects one Representation, typically based on the value of the **@bandwidth** attribute, but also taking into account client decoding and rendering capabilities. Then the client creates a list of accessible Segments for each Representation for the actual client-local time *NOW* measured in wall-clock time taking into account the procedures introduced in A.3.
- 3) The client accesses the content by requesting entire Segments or byte ranges of Segments. The client requests Media Segments of the selected Representation by using the generated Segment list.
- 4) The client buffers sufficient media in order to anticipate continuous playout and may use the value of **@minBufferTime** attribute, the associated **@bandwidth** value and the measured access bitrate to determine a sufficiently large buffer before starting the presentation. Then, once it has identified a Stream Access Point (SAP) for each of the media streams in the different Representations, it starts rendering (in wall-clock-time) of this SAP not before **MPD@availabilityStartTime + PeriodStart + T_{SAP}** and not after **MPD@availabilityStartTime + PeriodStart + T_{SAP} + @timeShiftBufferDepth** provided the observed throughput remains at or above the sum of the **@bandwidth** attributes of the selected Representations (if not, longer buffering may be needed). For services with

MPD@type='dynamic', rendering the SAP at the sum of **MPD**@availabilityStartTime + *PeriodStart* + T_{SAP} and the value of **MPD**@suggestedPresentationDelay is recommended, especially if synchronized play-out with other devices adhering to the same rule is desired.

- 5) Once the presentation has started, the client continues consuming the media content by continuously requesting Media Segments or parts of Media Segments. The client may switch Representations taking into account updated MPD information and/or updated information from its environment, e.g. change of observed throughput. With any request for a Media Segment containing a stream access point, the client may switch to a different Representation. Seamless switching can be achieved, as the different Representations are time-aligned. Advantageous switching points are announced in the MPD and/or in the Segment Index, if provided.
- 6) With the wall-clock time *NOW* advancing, the client consumes the available Segments. As *NOW* advances, the client possibly expands the list of available Segments for each Representation according to the procedures specified in A.3 If the following conditions are both true, an updated MPD should be fetched:
 - i) if the attribute **MPD**@minimumUpdatePeriod is present, and
 - ii) the current playback time gets within a threshold (typically described by at least the sum of the value of the @minBufferTime attribute) and the value of the @duration attribute (or the equivalent value in case the **SegmentTimeline** is used) of the media described in the MPD for any consuming or to be consumed Representation.
- 7) If the conditions in 6) are true, the client should fetch a new MPD, and update *FetchTime*. Once received, the client takes into account the possibly updated MPD and the new *FetchTime* in the regeneration of the accessible Segment list for each Representation.

In the following clauses a brief overview is provided of Segment list generation, seeking, support for trick modes, switching Representations, reaction to error codes, encoder clock drift control, playback across period boundaries, usage of bandwidth and buffer time in the DASH Client, as well as location and reference resolution are provided.

A.3 Segment list generation

A.3.1 General

Assume that the DASH Client has access to an MPD. This clause describes how a client may generate a Segment list for one Representation as shown in Table A.1 from an MPD obtained at *FetchTime* at a specific client-local time *NOW*. In this description, the term *NOW* is used to refer to “the current value of the clock at the reference client when performing the construction of an MPD Instance from an MPD”. A client that is not synchronized with a DASH server, which is in turn is expected to be synchronized to UTC, may experience issues in accessing Segments as the Segment availability times provided by the server and the local time *NOW* may not be synchronized. Therefore, DASH Clients are expected to synchronize their clocks to a globally accurate time standard.

Table A.1 Segment list in example client

Parameter Name	Cardinality	Description
Segments	1	Provides the Segment URL list.
InitializationSegment	0, 1	Describes the Initialization Segment. If not present, each Media Segment is self-initializing.
URL	1	The URL where to access the Initialization Segment (the client may add a byte range to the URL request if one is provided in the MPD).
MediaSegment	1 ... N	Describes the accessible Media Segments.
startTime	1	The MPD start time of the Media Segment in the Period relative to the start time of Period.
duration	1	The MPD duration for the Segment
URL	1	The URL where to access the Media Segment, possibly combined with a byte range.
IndexSegment	1 ... N	Describes the accessible Index Segments, if present.
URL	1	The URL where to access the Index Segment, possibly combined with a byte range.

According to subclause 5.3.9, there are three different ways to describe and generate a Segment List. This description focuses on the first two where either a **SegmentList** element or a **SegmentTemplate** element is present. The case with a single Media Segment using **BaseURL** element and **SegmentBase** element is considered straightforward.

Segments are available at their assigned URL if at wall-clock time *NOW* the Segment availability start time is smaller than or equal to *NOW* and the Segment availability end time is larger than or equal to *NOW*.

Furthermore, assume that for a Representation in a Period, the Segment list is indexed with $i=1, \dots, N$.

A.3.2 Period Start and End Times

Assume that for an MPD with fetch time *FetchTime*:

- the *MediaPresentationDuration* is provided either as the value of **MPD@mediaPresentationDuration** if present, or as the sum of *PeriodStart* + **Period@duration** of the last Period.
- the Period start time is provided as *PeriodStart* according to subclause 5.3.2.1 for any Period in the MPD.
- the Period end time referred as *PeriodEnd* is determined as follows: for any Period in the MPD except for the last one, the *PeriodEnd* is obtained as the value of the *PeriodStart* of the next Period. For the last Period in the MPD:
 - if the **MPD@minimumUpdatePeriod** attribute is not present, then *PeriodEnd* is defined as the end time of the Media Presentation, i.e. **MPD@availabilityStartTime** + *MediaPresentationDuration*.
 - if the **MPD@minimumUpdatePeriod** attribute is present, then *PeriodEnd* is defined as the smaller value of *FetchTime* + **MPD@minimumUpdatePeriod** and **MPD@availabilityStartTime** + *MediaPresentationDuration*.

A.3.3 Start Time and Duration

In case the Segment base information contains the @duration attribute, then

- the regular duration d is obtained as $d = \text{@duration} / \text{@timescale}$,
- the MPD start time $\text{MediaSegment}[i].\text{startTime}$ is obtained as $(i-1) * d$,
- the MPD duration $\text{MediaSegment}[i].\text{duration}$ is obtained as d unless this Segment is the last Segment in this Period, then the $\text{MediaSegment}[i].\text{duration}$ is obtained as $\text{PeriodEnd} - \text{MediaSegment}.\text{StartTime}[i]$.

In case the Segment base information contains a **SegmentTimeline** element with N_s **S** elements referred as $s=1, \dots, N_s$, then

- the $t[s]$ is the value of @t of the s -th **S** element divided by the value of the @timescale attribute,
- the o is the value of @presentationTimeOffset for this Representation divided by the value of the @timescale attribute,
- the $d[s]$ is the value of @d of the s -th **S** element divided by the value of the @timescale attribute,
- if the value of @r is greater than or equal to 0,
 - the $r[s]$ is one more than the value of @r of the s -th **S** element, and
 - $N=0$
- for $s=1, \dots, N_s$
 - $N=N+1$
 - $\text{MediaSegment}[N].\text{startTime} = t[s] - o$
 - $\text{MediaSegment}[N].\text{duration} = d[s]$
 - for $j=1, \dots, r[s]$
 - $N=N+1$
 - $\text{MediaSegment}[N].\text{startTime} = \text{MediaSegment}[N-1].\text{startTime} + d[s]$
 - $\text{MediaSegment}[N].\text{duration} = d[s]$
- else
 - the MPD duration $\text{MediaSegment}[i].\text{duration}$ is obtained as $d[0]$ unless this Segment is the last Segment in this Period, then the $\text{MediaSegment}[i].\text{duration}$ is obtained as $\text{PeriodEnd} - \text{MediaSegment}.\text{StartTime}[i]$.

If neither the @duration nor the **SegmentTimeline** element is given, then

- $N=1$,
- $\text{MediaSegment}.\text{startTime}[1]=0$,

— $\text{MediaSegment.duration}[1] = \text{PeriodEnd} - \text{PeriodStart}$.

If the Representation contains or inherits one or more **SegmentList** elements, providing a set of explicit URL(s) for Media Segments, then all N Segment URLs are provided.

If the Representation contains or inherits a **SegmentTemplate** element with $\$Number\$$ then the URL of the Media Segment i , $\text{MediaSegment.URL}[i]$, is obtained by replacing the $\$Number\$$ identifier by $i-1 + @startNumber$ in the **SegmentTemplate@media** string.

If the Representation contains or inherits a **SegmentTemplate** element with $\$Time\$$ then the URL of the Media Segment i , $\text{MediaSegment.URL}[i]$, is obtained by replacing the $\$Time\$$ identifier by $\text{MediaSegment}[i].startTime$ in the **SegmentTemplate@media** string, as described in subclause 5.3.9.5.3.

A.3.4 Media Segment list restrictions

The Media Segment List is restricted to a list of accessible Media Segments, which may be a subset of the Media Segments of the complete Media Presentation. The construction is governed by the current value of the clock at the client *NOW* which is greater than or equal to the *FetchTime* of the MPD.

Segments may only be accessed during their Segment availability times. Generally, Any Segment may only be available for any time *NOW* between **MPD@availabilityStartTime** and **MPD@availabilityEndTime**. For times *NOW* outside this window, no Segments are available.

Assume the parameter *availabilityTimeOffset* is determined as the sum of all values of **@availabilityTimeOffset** on all levels that are processed in determining the URL for the corresponding segment. If the attribute **@availabilityTimeOffset** is not present, the value is of *availabilityTimeOffset* is 0. Then for services with **MPD@type='dynamic'**, the Segment availability start time $T_{\text{avail}}[i]$ for a Segment i in a specific Period is determined as **MPD@availabilityStartTime** + *PeriodStart* + $\text{MediaSegment}[i].startTime$ + $\text{MediaSegment}[i].duration$ - *availabilityTimeOffset* and the Segment availability end time is determined as **MPD@availabilityStartTime** + *PeriodStart* + $\text{MediaSegment}[i].startTime$ + **@timeshiftBufferDepth** + $2 * \text{MediaSegment}[i].duration$.

In case of MPD updates, assume the variable *CheckTime* associated to an MPD with *FetchTime* is defined as the sum of the fetch time of this operating MPD and the value of the attribute **MPD@minimumUpdatePeriod**, i.e. $\text{CheckTime} = \text{FetchTime} + \text{MPD@minimumUpdatePeriod}$. The *CheckTime* is defined on the MPD-documented media time axis; when the client's playback time reaches $\text{CheckTime} - \text{MPD@minBufferTime}$, it should fetch a new MPD.

Therefore, based on an MPD that was fetched at fetch time *FetchTime* and has associated a check time *CheckTime*, the largest index i_{max} that is accessible at time *NOW* for the last Period in the MPD is $i_{\text{max}} = \max_i \{ T_{\text{avail}}[i] \leq \min(\text{CheckTime}, \text{NOW}) \}$.

A.4 Seeking

Assume that a client attempts to seek to a specific Media Presentation time T_M in a Representation relative to the *PeriodStart* time. According to subclause 7.2.1, the presentation times within each Period are relative to the *PeriodStart* time of the Period minus the value of the **@presentationTimeOffset**, T_0 , of the containing Representation.

Based on the MPD, the client has access to the MPD start time and Media Segment URL of each Segment in the Representation, along with Index Segment URL, if present. The Segment number of the Segment most likely to contain media samples for Media Presentation time T_M is obtained as the maximum

Segment index i^* , for which the start time `MediaSegment[i].startTime` is smaller than or equal to the T_M . The Segment URL is obtained as `MediaSegment[i*].URL`.

Timing information in the MPD may be approximate due to issues related to placement of Stream Access Points, alignment of media tracks and media timing drift. As a result, the Segment identified by the procedure above may begin at a time slightly after T_M and the media data for presentation time may be in the previous Media Segment. In case of seeking, either the seek time may be updated to equal the first sample time of the retrieved Media Segment, or the preceding Media Segment may be retrieved instead. However, during continuous playout, including cases where there is a switch between alternative versions, the media data for the time between T_M and the start of the retrieved Segment is always available.

For accurate seeking to a presentation time T_M , the DASH Client needs to access Stream Access Points (SAP). To determine the SAP in a Media Segment in case of DASH, the client may, for example, use the information in the Segment Index, if present, to locate the stream access points and the corresponding presentation time in the Media Presentation.

In the case that the Media Presentation is based on the ISO base media file format and a Segment is a movie fragment, it is also possible for the client to use information within the 'moof' and 'mdat' boxes, for example, to locate Stream Access Points in the media and obtain the necessary presentation time from the information in the movie fragment and the Segment start time derived from the MPD. If no SAP with presentation time before the requested presentation time T_M is available, the client may either access the previous Segment or may just use the first SAP as the seek result. When Media Segments start with a SAP, these procedures are simplified.

In the case that the Media Presentation is based on MPEG-2 TS, the presentation units corresponding to the desired presentation time T_M can be identified by using the indexing information, if present, in conjunction with the differential value of the presentation time stamps (PTS) within the Media Segment. For example, if $T_{M,S}$ denotes the presentation time corresponding to the last SAP leading the desired seek time tp , with a corresponding PTS denoted as PTS_s , then the desired seek position within the media has a PTS expressed as: $[(T_M - T_{M,S}) * \text{timescale} + PTS_s \% 2^{33}]$.

Also, not necessarily all information of the Media Segment needs to be downloaded to access the presentation time T_M . The client may for example initially request the Segment Index from the beginning of the Media Segment using partial HTTP GET. By use of the Segment Index, Segment timing can be mapped to byte ranges of the Segment. By continuously using partial HTTP GET requests, only the relevant parts of the Media Segment may be accessed for improved user experience and low start-up delays.

A.5 Support for trick modes

The client may pause or stop a Media Presentation. In this case, the client simply stops requesting Media Segments or parts thereof. To resume, the client sends requests to Media Segments, starting with the next Subsegment after the last requested Subsegment.

If a specific **Representation** or **SubRepresentation** element includes the `@maxPlayoutRate` attribute, then the corresponding Representation or Sub-Representation may be used for the fast-forward trick mode. The client may play the Representation or Sub-Representation with any speed up to the regular speed times the specified `@maxPlayoutRate` attribute with the same decoder profile and level requirements as the normal playout rate. If a specific **Representation** or **SubRepresentation** element includes the `@codingDependency` attribute with value set to 'false', then the corresponding Representation or Sub-Representation may be used for both fast-forward and fast-rewind trick modes.

Sub-Representations in combination with Index Segments and Subsegment Index boxes may be used for efficient trick mode implementation. Given a Sub-Representation with the desired `@maxPlayoutRate`,

ranges corresponding to **SubRepresentation@level** all level values from **SubRepresentation@dependencyLevel** may be extracted via byte ranges constructed from the information in Subsegment Index Box. These ranges can be used to construct more compact HTTP GET request.

A.6 Switching Representations

Based on updated information during an ongoing Media Presentation, a client may decide to switch Representations. Switching to a “new” Representation is equivalent to tuning in or seeking to the new Representation from the time point where the “old” Representation has been presented. Once switching is desired, the client should seek to a SAP for each media stream in the “new” Representation at a desired presentation time *tp* later than and close to the current presentation time. Presenting the “old” Representation up to (but not included) the presentation time of the SAP in the “new” Representation and presenting the “new” Representation from the presentation time of the SAP enables seamless switching.

If **@segmentAlignment** is set to true and the **@startWithSAP** is set to 1 or 2 then the client may switch at any Segment boundary

- by just concatenating Segments with consecutive Segment numbers from different Representations, if **@bitstreamSwitching** flag is set to true on the parent Adaptation Set, or
- by loading the Initialization Segment or Bitstream Switching Segment for the new Representation before processing the new Segment.

No overlap downloading and decoding is required.

If **@segmentAlignment** is set to true and the **@startWithSAP** is set to 3 and the **Representation@mediaStreamStructureId** is identical for the two Representations, then the client may switch at any Segment boundary by just concatenating Segments with consecutive Segment numbers from different Representations, without re-initialization of the media decoder. **@bitstreamSwitching** should be set to true in this case.

The same can be achieved on Subsegment level with **@subsegmentAlignment** set to true and **@subsegmentStartsWithSAP** the same values and conditions as above.

A.7 Reaction to error codes

The DASH access client provides a streaming service to the user by issuing HTTP requests for Segments at appropriate times. The DASH access client may also update the MPD by using HTTP requests. In regular operation mode, the server typically responds to such requests with status code 200 OK (for regular GET) or status code 206 Partial Content (for partial GET) and the entity corresponding to the requested resource. Other Successful 2xx or Redirection 3xx status codes may be returned.

HTTP requests may result in a Client Error 4xx or Server Error 5xx status code. Some guidelines are provided in this subclause as to how an HTTP client may react to such error codes.

If the DASH access client receives an HTTP client or server error (i.e. messages with 4xx or 5xx error code), the client should respond appropriately (e.g. as indicated in IETF RFC 7231) to the error code. In particular, clients should handle redirections (such as 301 and 307) as these may be used as part of normal operation.

If the DASH access client receives a repeated HTTP error for the request of an MPD, the appropriate response may involve terminating the streaming service.

If the DASH access client receives an HTTP client error (i.e. messages with 4xx error code) for the request of an Initialization Segment, the Period containing the Initialization Segment may not be available anymore or may not be available yet.

Similarly, if the DASH access client receives an HTTP client error (i.e. messages with 4xx error code) for the request of a Media Segment, the requested Media Segment may not be available anymore or may not be available yet. In both cases, the client should check if the precision of the time synchronization to a globally accurate time standard is sufficiently accurate. If the clock is believed accurate, or the error re-occurs after any correction, the client should check for an update of the MPD.

Upon receiving server errors (i.e. messages with 5xx error code), the client should check for an update of the MPD. If multiple **BaseURL** elements are available, the client may also check for alternative instances of the same content that are hosted on a different server.

A.8 Encoder clock drift control

Non-alignment between the end of a Representation in one Period and the start time of the next Period may be caused by encoder clock inaccuracy. The client should align the Media Presentation time at each Period start. In addition, significant deviations of the start time of Segments to the media time should be detected and drift-compensating measures may be applied even before the start of the next Period is reached.

Over a longer operation time, a difference in clock accuracy of the encoder and decoder may cause the playback to lag behind real-time or to interrupt temporarily due to the client trying to access data faster than real-time.

For ISO base media file format based, clients may avoid these anomalies by using the Producer Reference Time boxes as follows. The pace r_1 of the encoder clock in relation to the UTC is recovered from Producer Reference Time boxes. If the relative pace r_1 is less than 1, equal to 1, or greater than 1, the encoder clock runs more slowly than the UTC, at an identical pace compared to the UTC, or faster than the UTC, respectively. The pace r_2 of the receiver playout clock in relation to UTC is created by accessing a UTC source. A timescale multiplication factor c is equal to r_1/r_2 . A presentation time on a timeline of the receiver playout clock is derived for each sample or access unit by multiplying the composition time of the sample (as indicated by the file format structures) or the presentation time of the access unit (as indicated by the respective Program Elementary Stream header) by the timescale multiplication factor c .

In case of MPEG-2 TS segments, PCR-based drift control may be used.

A.9 Playback across Period boundaries

From a client perspective, Period boundaries may require processing that makes fully continuous playout impossible or at last practically complex. For example, the content may be offered with different codecs, different language attributes, different protection and so on. The client is expected to play the content continuously across Periods, but there may be implications in terms of implementation to provide fully continuous and seamless playout. It may be the case that at Period boundaries, the presentation engine needs to be reinitialised, for example due to changes in formats, codecs or other properties. This may result in a re-initialisation delay. Such a re-initialisation delay should be minimized.

If the client presents media components of a certain Adaptation Set with a specific value f_{oo} for the **AdaptationSet@id** in one Period, and if the following Period has assigned an identical Asset Identifier, then the client is suggested to identify an associated Period and, in the absence of other information, continue playing the content in the associated Adaptation Set, i.e. the Adaptation Set with value f_{oo} for the **AdaptationSet@id**.

If the client presents media components of a certain Sub-Representation in one Period, and if the following Period has assigned an identical Sub-Asset Identifier, then the client is suggested to identify an associated Period and, in the absence of other information, continue playing the content in the associated Sub-Representation.

If furthermore the Adaptation Sets are *period-continuous* or *period-connected* as defined in subclause 5.3.2.4, i.e. the presentation times are continuous and this is signalled in the MPD, then the client is expected to seamlessly play (as defined in subclause 4.5.1) the content across the Period boundary. Most suitably the client may continue playing the Representation in the Adaptation Set with the same `@id`, but there is no guarantee that this Representation is available. In this case, the client is expected to seamlessly switch (as defined in subclause 4.5.1) to any other Representation in the Adaptation Set. If period continuity is signalled and if continuously playing, then the client should ignore the value of the `@presentationTimeOffset` attribute and just continue processing the incoming Segments. If period connectivity is signalled and if continuously playing, then the client is expected to inform the media decoder on a timeline discontinuity obeying the value of `@presentationTimeOffset` attribute, but it may continue processing the incoming Segments without for example re-initializing the media decoder.

A.10 Usage of Bandwidth and Min Buffer Time in DASH Client

In a simple and straightforward implementation, a DASH Client decides downloading the next segment based on the following status information:

- the currently available buffer in the media pipeline, *buffer*,
- the currently estimated download rate, *rate*,
- the value of the attribute `@minBufferTime`, *MBT*,
- the set of values of the `@bandwidth` attribute for each Representation *i*, *BW[i]*.

The task of the client is to select a suitable Representation *i*.

The relevant issue is that starting from a SAP on, the DASH Client can continue to playout the data. This means that, at the current time, it does have *buffer* data in the buffer. Based on this model, the client can download a Representation *i* for which $BW[i] \leq rate * buffer / MBT$ without emptying the buffer.

In this model, some idealizations typically do not hold in practice, such as constant bitrate channel, progressive download and playout of Segments, no blocking and congestion of other HTTP requests, etc. Therefore, a DASH Client should use these values with care to compensate such practical circumstances; especially variations in download speed, latency, jitter, scheduling of requests of media components, as well as to address other practical circumstances.

One example is if the DASH Client operates on Segment granularity. As in this case, not only parts of the Segment (i.e. *MBT*) need to be downloaded, but the entire Segment, and if the *MBT* is smaller than the Segment duration, then rather the segment duration needs to be used instead of the *MBT* for the required buffer size and the download scheduling, i.e. download a Representation *i* for which $BW[i] \leq rate * buffer / max_segment_duration$.

A.11 Location and reference resolution

This document defines several functionalities in order to provide a consistent set of URLs for MPD updates, DASH segments and other resources in the MPD.

Specifically, in subclause 5.3.1.2, the **MPD.Location** element is defined and rules for MPD updates using this element are defined in subclause 5.4.1. In subclause 5.6.4, reference resolution is defined, i.e. the

usage of MPD URLs as well **BaseURL** elements on different levels. Subclause 5.6.5 defines the usage of multiple **BaseURL** elements.

Based on these descriptions, a DASH Client operating on MPD updates and segment requests is expected to operate as follows:

- For MPD update requests (when due) according to clause 5.4, the DASH Client requests the MPD according to the following priority
 - If at least one **MPD.Location** element is present,
 - the value of any **MPD.Location** element is used as the MPD request URL
 - else
 - If the HTTP request results in an HTTP redirect using a 3xx response code, the redirected URL replaces the original manifest URL,
 - Else
 - If present and known, the original manifest URL is used for updates
 - else updates cannot be done, and client may terminate the service
 - In addition, the manifest URL as derived above provides an implicit base URI
 - Any present **BaseURL** element does not apply to MPD updates
- Based on the knowledge of the MPD request URL according to the previous algorithm, the DASH Client requests Segments according to the following priority
 - If present, any absolute base URL or an absolute URL is used,
 - Else
 - If the base URL is known
 - the base URL from the above MPD request URL provides the base URL and the relative base URL is used for the segment requests
 - else
 - the service may be terminated.

A.12 Resynchronization and Early Access to Segments

A.12.1 General

Several cases are identified when Resynchronization within a Segment is important and beneficial. Examples include:

- Low latency streaming and fast access to the service
- Fast channel acquisition in broadcast services
- Low latency streaming and resynchronization after losses or buffer underruns or in advance of a potential stall.
- Fast down-switching in the face of imminent stall
- Fast and efficient seeking to time (e.g. better seek-to accuracy).

Resynchronization and restart of playback in the case of the ISO BMFF Segments requires multiple processes:

1. Finding the box structure within the Segment
2. Finding a proper Resynchronization Point including with all relevant information that is needed to start parsing and decoding
3. Finding the earliest presentation time that is presented
4. Processing of Event messages, if applicable
5. Obtaining all decryption relevant information, if applicable
6. Start decoding on elementary stream level

In addition, a DASH client is preferably aware of the availability of Resynchronization Points.

This subclause also addresses the issues on the early availability and accessibility of initial parts of Segments.

The Resynchronization Point feature as defined in subclause 5.3.13 supports fast resynchronization. This subclause provides a few client implementation guidelines for fast Resynchronization in the client.

A.12.2 Box Structure Resynchronization

Box structure resynchronization is an important feature for many playback pipelines. Without aligning to the box structure, such pipelines fail. Secondly, resynchronization to ISO BMFF segments and proper elementary stream decoding is also required for proper playback of the content.

There are different ways to resynchronize on box structure at a specific time.

1. If the Segment Index is provided, then resynchronization can be done at presentation times and byte offsets. However, a Segment Index is typically not available for dynamic services.
2. If a timed metadata track is provided, then resynchronization can be done at presentation times and byte offsets. This may apply for regular live services, but not be suitable in case of low-latency.
3. If the start of the Segment is accessible, the Segment can be downloaded from the beginning and be parsed until the proper Index and Time is found. However, such downloading consumes additional unnecessary bandwidth or may not be fast enough to be useful.
4. A resynchronization information is provided by the underlying protocol, that for example provides the Index to each Resynchronization Point and this information is passed to the DASH client.
5. If the start of the Segment is not available, then finding a Resynchronization Point based on a proper pattern is possible. Once found, regular parsing can start and find the proper box structure that allows one to process, in particular 'emsg', 'prft', 'moof' and 'mdat'.

A.12.3 Usage of Resynchronization Feature

If a timed metadata track is provided indicating Resynchronization Points, then resynchronization can be done at the specified Index and Time values, taking into account the type of the Resynchronization Point.

If the client wants to resynchronize to a Time within a Segment and does not have any information of the Index or a supplementary protocol on the Index within the Segment that corresponds to the time, the client may resync by parsing the Segment.

Before parsing the Segment, it is recommended to download only a part of the Segment that will include the Resynchronization Point. For this purpose, the information in the Resync@dT, Resync@dMax and

`Resync@dMin` can be used. If the **Resync**@marker is set, then the client can parse the byte stream as follows to find the Index of the next Resynchronization Point:

1. Find an occurrence of the 'styp' byte string in a segment, say at byte offset B^* .
2. Verify against a random emulation as follows: the next box type is compared against the list of any of the expected box types: 'styp', 'prft', 'emsg', 'moof', 'mdat', 'free', 'skip'.
3. If one of the known box types is found, byte offset B^*-4 octets is the Index of the Resynchronization Point.
4. If this is not one of the known box types noted before, this occurrence of 'styp' box is considered an invalid Resynchronization Point and ignored. Restart from step 1 above.

Once found, the remaining issue is the determination of:

- the earliest presentation time `Time` of the Resynchronization Point. This is easily accomplished by the use of the 'tfdt' and other information in the movie fragment header.
- the type of the Resynchronization Point. Several options exist:
 - A detection based on the information in the 'moof'.
 - The use of compatibility brands for SAP types. If CMAF is in use as defined in ISO/IEC 23000-19, the following can be deduced
 - 'cmff': indicates that the SAP is 1 or 2
 - 'cmfl': indicates that the SAP is 0
 - 'cmfr': indicates that the SAP is 1, 2 or 3

Once proper `Time` and `Index` is found, an early Resynchronization the media pipeline can be initiated.

A.12.4 Early availability and access of Segments

For a Representation that includes an `@availabilityTimeOffset`, the following client behaviour is expected:

- If no **Resync** with `@dT` and `@dImin` is present and the `@availabilityTimeComplete` set to FALSE, then
 - clients are expected to not access the Segment with byte range requests,
 - clients cannot expect to receive any Segment data prior to Segment availability start time, even if a request for Segment is issued.
- If a Resync element is present **Resync** with `@dT` and `@dImin`,
 - clients can expect that available Segment data will be received over time, if a request for the Segment is issued.
 - clients are expected to not access the Segments with byte range requests that are not yet available.
 - If furthermore `@rangeAccess` on **BaseURL** attribute associated to the Representation is set to true
 - The clients are permitted to access byte range requests that are available at the current time *NOW*.

A.13 Timing, processing, and client reference model for DASH Event streams and timed metadata tracks

A.13.1 General

This annex describes DASH events and timed metadata track timing and processing. It describes the timing models for MPD and inband event streams, as well as the timing of timed metadata tracks. This subclause also outlines the DASH player’s reference architecture for processing DASH event streams as well as timed metadata tracks. Specifically, two possible dispatch modes of events are introduced. Finally, it defines a reference API for an application to subscribe to Event streams and/or timed metadata tracks as well as an API for dispatching event instances and metadata samples.

A server/application provider should consider the information provided in this annex for building interactive applications as the timing and processing model of Event streams and timed metadata tracks.

A.13.2 DASH Player architecture for processing DASH events and timed metadata tracks

Figure A.1 demonstrates a generic architecture of DASH Player including the DASH Event stream and timed metadata track processing.

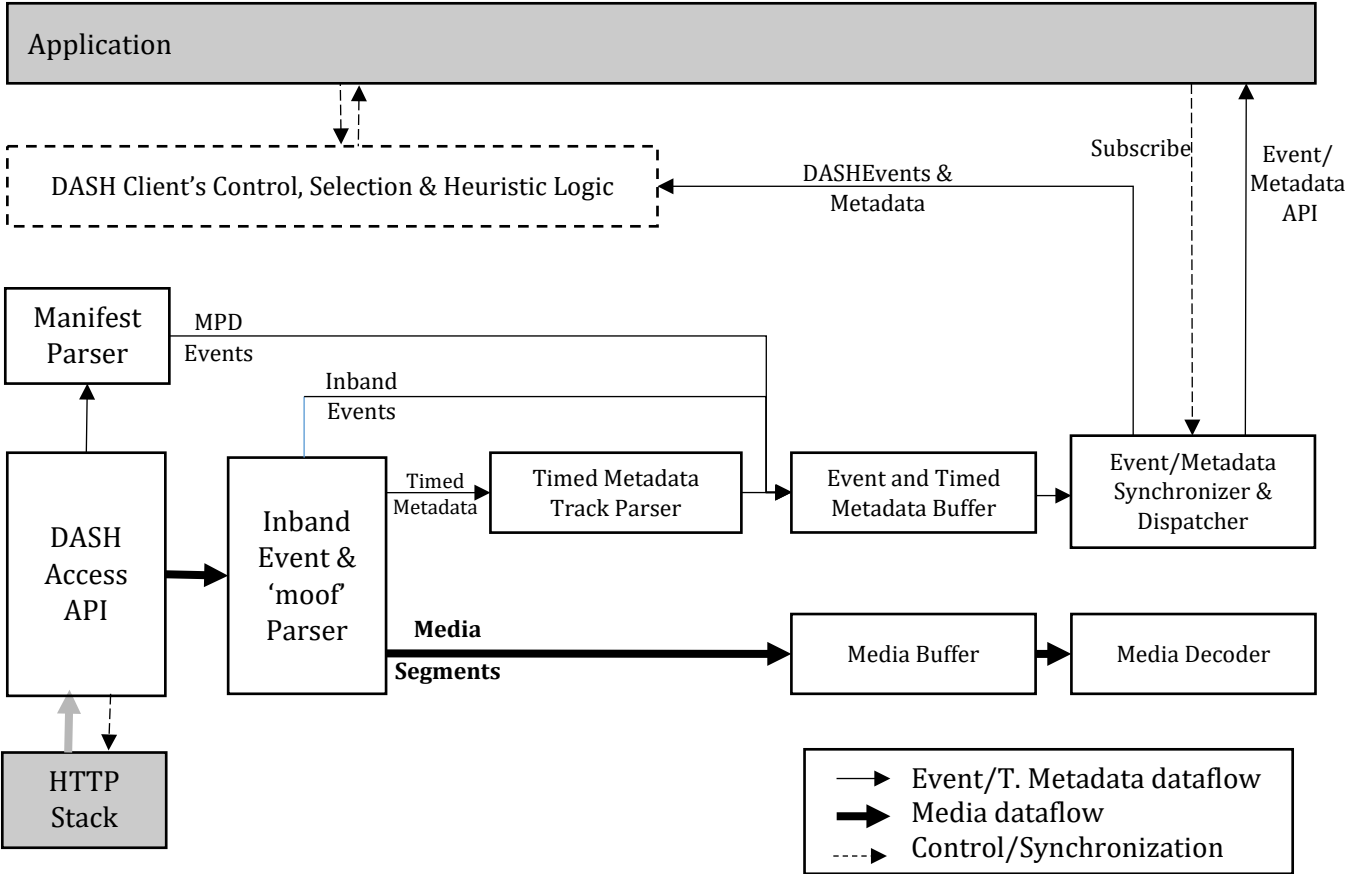


Figure A.1 DASH Player architecture including the Event Stream timed metadata track handling

In Figure A.1 the following applies:

- The DASH Player processes the received MPD. For every Period, the MPD may include one or multiple (Inband-)Event Streams (each scoped by a scheme/value pair), and Adaptation Sets that carry Representations/tracks for timed metadata. Some or all of these event streams/timed metadata tracks may be suitable for consumption by the application. In this subclause, we refer to any of these streams/tracks as application event or metadata streams (AEMS).
- The Application subscribes to all AEMSs of interest and also specifies the desired dispatch mode for each AEMS.
- If the MPD includes any MPD Event streams, the DASH Player parses each Event in the Event Stream accordingly and appends the relevant to the Event & Timed Metadata Buffer, based on their presentation time.
- Based on the information in the MPD, the DASH Player selects and schedules the fetching of Media Segments and appends them to the Media Buffer. This is typically done to maintain a stable playback buffer, but media segments are typically only parsed close to the time before their playback is scheduled. Parsing a Segment includes:
 - Parsing high-level boxes such as Segment Index ('*sidx*'), Event Message ('*emsg*'), Producer Reference Time ('*prft*'), movie fragment header ('*moof*') boxes, and interpreting the information in the DASH client.
 - For inband event streams, the '*emsg*' and typically the '*moof*' need to be parsed. The DASH client then uses this information and appends the relevant data to the Event & Timed Metadata Buffer.
- For an Application-related timed metadata track, the entire Representation/track is parsed including the Initialization Segment (i.e. track header) as well as the Media Segments (i.e. the movie fragments and media data containers). The DASH client then uses this information and appends the relevant data to the Event & Timed Metadata Buffer.
- Event & Metadata Buffer passes the events and timed metadata samples to Event & Metadata Synchronizer and Dispatcher function. An example of the Buffer's data object is defined in subclause A.13.11.
- The DASH Player-specific Events are dispatched to DASH Player's Control, Selection & Heuristic Logic, while the Application-related Events and timed metadata track samples are dispatched to the application as the following. If an Application is subscribed to a specific AEMS, dispatch the corresponding event instances or timed metadata samples, according to the dispatch mode:
 - For on-receive dispatch mode, dispatch the entire event or timed metadata information as soon as they are appended to the Event and Timed Metadata Buffer.
 - For on-start dispatch mode, dispatch the message data of the event their associated presentation time or latest before the event duration has ceased, or timed metadata samples at their presentation time using the synchronization signal from the media decoder.

NOTE The metadata buffer maintains a sequence of events and timed metadata samples. The maintain and purging management of this buffer is synchronized with the Media buffer management.

A.13.3 Inband Event timing parameters

Figure A.2 presents the timing of an inband Events along the media timeline:

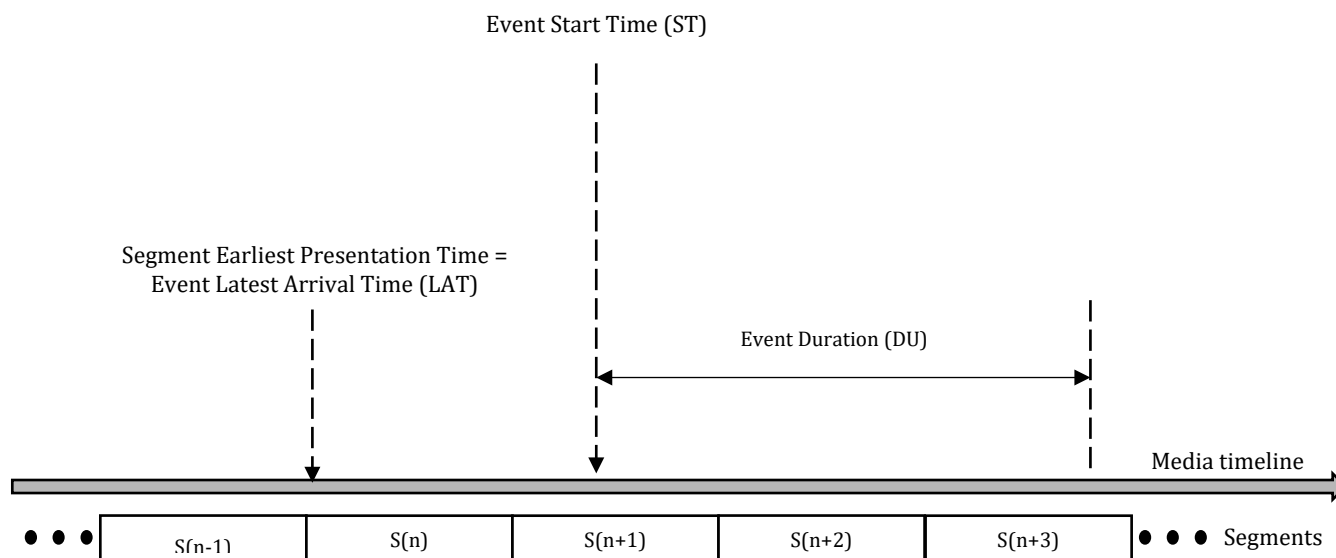


Figure A.2 Inband event timing parameter on the media timeline

As shown in Figure A.2, every inband Event can be described by three timing parameters on the media timeline:

- Assuming an inband Event is inserted at the beginning of a Segment. Then the Event Latest Arrival Time (*LAT*) is defined as the earliest presentation time of the Segment containing the Event Message box. Since each Media Segment has the earliest presentation time equal to (*LAT*), *LAT* of the Segment carrying the Event Message box can be considered as the time instance of that box on the media timeline. The DASH Player is expected to fetch and parse the Segment before or at its earliest presentation time. Therefore, an Event inserted in a Segment with EPT will be available in the client no later than EPT of the carrying Segment on the media timeline. Therefore, the Event inserted in a Segment will be ready to be processed and fetched **no later than** *LAT* on the media timeline. In the case in which the event is inserted at the beginning of a movie fragment that is not coinciding with the start of a Media Segment, and the movie fragments are delivered in low latency mode, i.e. HTTP chunked encoding mode is used, then the *LAT* is the earliest presentation time of the corresponding movie fragment.
- Event Presentation/Start Time (*ST*) which is the moment in the media timeline that the Event becomes active. *ST* is the moment in the media timeline that the Event becomes active. This value can be calculated using the parameters included in the `DashEventMessageBox`.
- Event duration (*DU*): the duration for which the Event is active. *DU* is signalled in the Event Message box using a specific value.

A.13.4 Event message box format and event timing parameters

The event presentation time ST of an event can be calculated using the values in the corresponding 'emsg' box of subclause 5.10.3.3.1:

$$ST = PS - \frac{PTO}{TS} + \begin{cases} LAT + \frac{PT}{TS} & \text{for version 0} \\ \frac{PT}{TS} & \text{for version 1} \end{cases}$$

where:

- PST is the Period start time of the Period containing the event stream.
- PTO is the presentation time offset of the Event Stream carrying Representation in the Period provided by **SegmentBase**@presentationTimeoffset.
- PTD is the value of the event message box's field emsg.presentation_time_delta in version 0 of the event message box.
- TS is the value of the event message box's field emsg.timescale.
- PT is the value of the event message box's field emsg.presentationtime in version 1.
- LAT is the earliest presentation time of the Segment containing the Event Message box.

NOTE 1 ST is always equal to or larger than LAT in both versions of 'emsg'.

NOTE 2 Since the media sample timescales might be different from the 'emsg's timescale, ST might not line up exactly with a media sample presentation time if different timescales are used.

In addition, the following common variable names are introduced instead of some of above variables to harmonize parameters between Inband events, MPD events, and timed metadata samples:

- $scheme_id$ = scheme_id_uri
- $value$ = value
- $presentation_time$ = ST
- $duration$ = event_duration/timescale
- $message_data$ = message_data[]

A.13.5 MPD Events timing model

MPD Events follow an equivalent data model to inband Events but are carried in the MPD within a **Period** element. Each Period event can have one or multiple **EventStream** elements, defining the **EventStream**@schemeIdUri, **EventStream**@value, **EventStream**@timescale, and contained sequences of Event elements. Each event may have **Event**@presentationTime, **Event**@duration, **Event**@id, and **Event**@messageData attributes as specified in subclause 5.10.2. As is shown in Figure A.3, each MPD Event has three timing parameters along the media timeline:

- The PeriodStart Time (LAT) of the **Period** element containing the **EventStream** element.

- Event Start Time (*ST*): the moment in the media timeline that a given MPD Event becomes active and can be calculated from the **attributeEvent@presentationTime**.
- Event duration (*DU*): the duration for which the event is active that can be calculated from the attribute **Event@duration**.

Note that the first parameter is inherited from the Period containing the Events and only the 2nd and 3rd parameters are explicitly included in the Event element. Each **EventStream** also has **EventStream@timescale** to scale the above parameters.

Figure A.3 demonstrates these parameters in the media timeline.

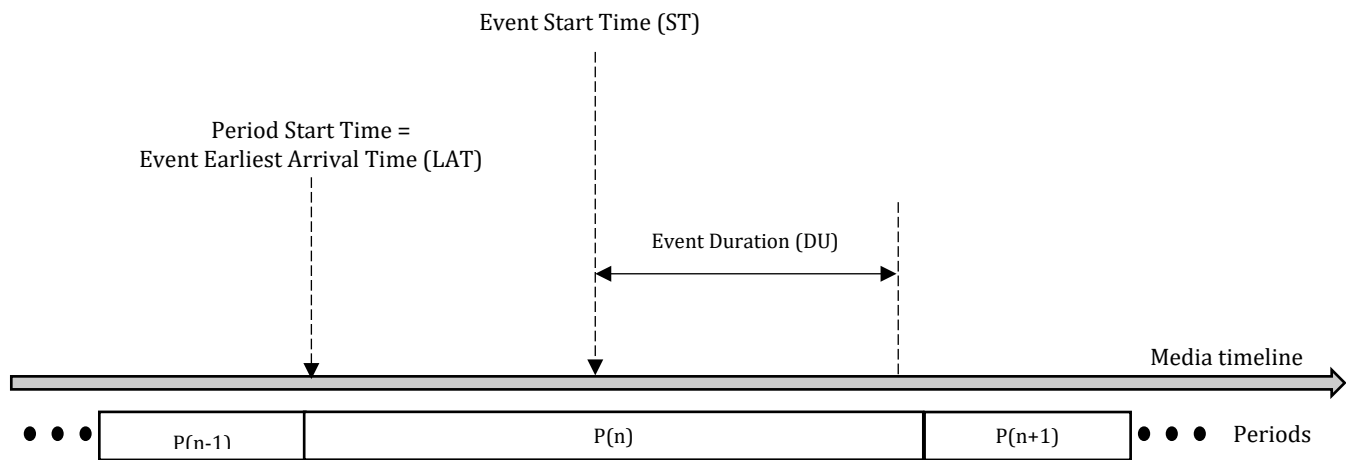


Figure A.3 MPD events timing model

The *ST* of an MPD event, relative to period start time of Period containing the Event, can be calculated using values in its **EventStream** and **Event** elements:

$$ST = PST + \frac{PT - PTO}{TS}$$

where:

- *PST* is the Period start time of the Period containing the event stream.
- *PTO* is the presentation time offset of the Event Stream provided by **EventStream@presentationTimeoffset**.
- *TS* is the value of **EventStream@timescale**.
- *PT* is the value of **EventStream@presentationTime**.

In this Annex, we use the following common variable names instead of some of above variables to harmonize parameters between Inband events, MPD events, and timed metadata samples:

- *scheme_id* = **EventStream@schemeIdUri**
- *value* = **EventStream@value**
- *presentation_time* = *ST*

- $duration = \mathbf{Event@duration} / \mathbf{EventStream@timescale}$
- $id = \mathbf{Event@id}$
- $message_data[] = \text{decode64}(\mathbf{Event@messageData})$

in which $\text{decode}()$ function is:

$$\text{decode}(x) = \begin{cases} x & \text{Event@contentEncoding not present} \\ \text{base64 decoding of } (x) & \text{Event@contentEncoding = base64} \end{cases}$$

Note that the DASH client is expected to Base64 decode the **Event@messageData** value if the received **Event@contentEncoding** value is base64.

Note that the Event duration may be unknown.

A.13.6 Timed metadata sample timing model

An alternative way to convey information synchronized to a media is using timed metadata tracks. Timed metadata tracks are ISOBMFF formatted tracks that obey the following characteristics according to ISO/IEC 14496-12:

- The sample description box 'std' in the **MovieBox** contains a **sampleEntry** that is a **URIMetaSampleEntry**, to signal that the media samples contain metadata based on a URI in a **URIBox** to signal that scheme.
- The Handler Box 'hdlr' has **handler_type** set to **meta** to signal the fact that the track contains metadata
- The null media header 'nmhd' is used in the 'minf' box
- Contain metadata (non-media data relating to presentation) is embedded in ISOBMFF samples

Figure A.4 shows the timing model for a simple ISOBMFF timed metadata sample.

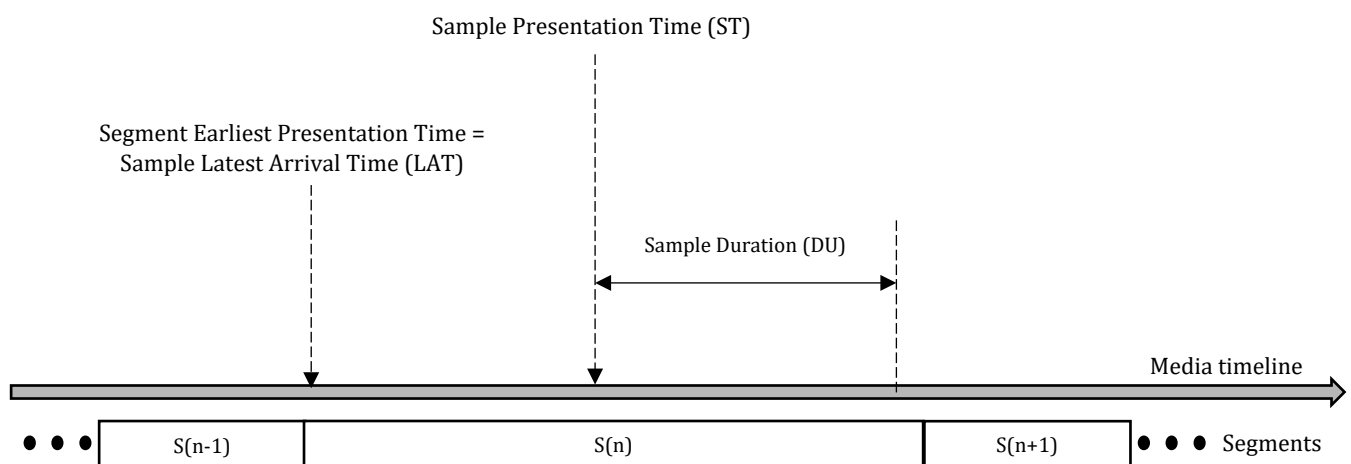


Figure A.4 Timing parameters of a timed metadata sample on the media timeline

As shown in Figure A.4, the metadata sample timing includes metadata sample presentation time (ST) and metadata sample duration (DU). Also, one or more metadata samples are included in a segment with the Segment's earliest presentation time (LAT).

Note that the metadata sample duration cannot go beyond DASH Segments/ISOBMFF fragment duration that carries the sample for fragmented metadata tracks, i.e. to the next fragment.

In this document, we use the following variable names instead of some of the above variables to harmonize parameters between Inband events, MPD events, and timed metadata samples used in the dispatch process:

- *scheme_id* = track URI, signalled in `URIBox` in `URIMetaSampleEntry`
- *timescale* = track timescale in 'mdhd' box.
- *presentation_time* = sample presentation time/timescale
- *duration* = sample duration/timescale
- *message_data* = sample data (extracted from 'mdat')

A.13.7 Events and timed metadata sample dispatch timing modes

Figure A.5 shows two possible dispatch timing models for DASH events and timed metadata samples.

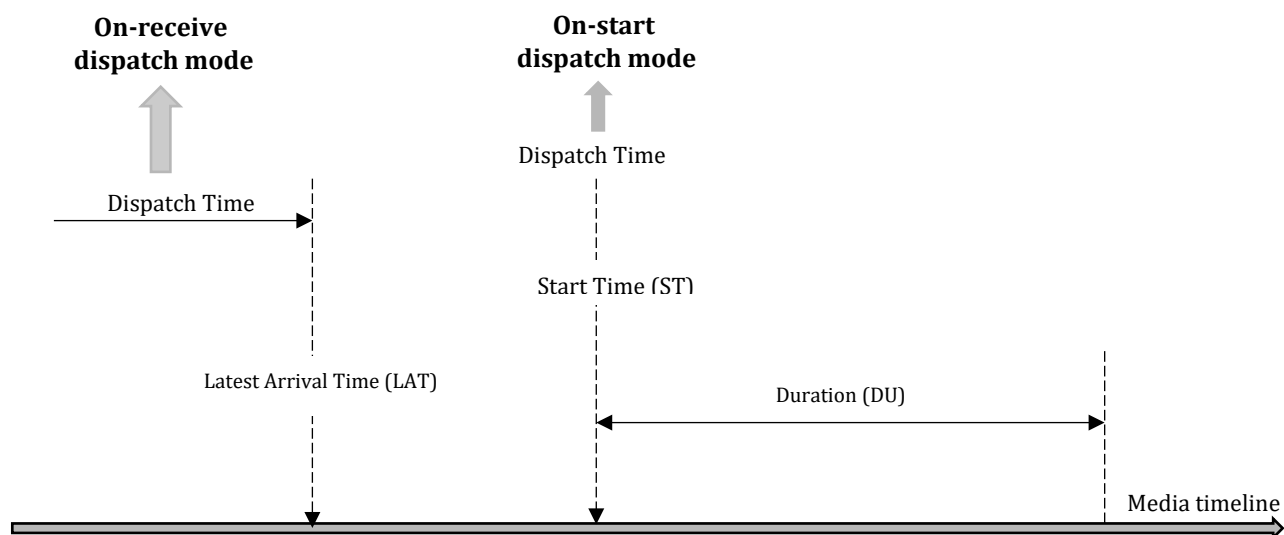


Figure A.5 The Application events and timed metadata dispatch modes

In Figure A.5, the following two modes are shown:

- **on-receive** Dispatch Mode: Dispatching at LAT or at the earliest time possible. Since the segment carrying an 'emsg'/metadata sample has to be parsed at or before LAT on the media timeline, the event/metadata sample shall be dispatched at or before this time to the Application in this mode. The Application has a duration of $ST-LAT$ for preparing for the event. In this mode, the DASH Player doesn't need to maintain the state of Application events or metadata samples. Applications must maintain the state for any event/metadata sample, its ST and DU , and monitor its activation duration if they need these parameters. Applications may also need to schedule each event/sample at its ST .

- **on-start** Dispatch Mode: Dispatching exactly at *ST*, which is the start/presentation time of the event/metadata sample. The DASH player shall dispatch the event to the application at the presentation time of the corresponding media sample, or in the case of the start of playback after that moment and during the event duration, at the earliest time within the event duration. In this mode, since Applications receives the event/sample at its start/presentation time, it may need to act on the received data immediately.

NOTE According to ISO/IEC 23009-1, the parameter duration has a different meaning in each dispatch mode. In the case of on-start, duration defines the duration starting from *ST* in which DASH Player is expected to dispatch the Event exactly once. In the normal playback, the player dispatches the Event at *ST*. However, if the DASH Player, for instance, seeks to a moment after *ST* and during the above duration, then it is expected to dispatch the Event immediately. In the case of on-receive, duration is a property of event instance and is defined by the `scheme_id` owner.

A.13.8 The Dispatch Processing Model

It is assumed that the application is subscribed to a specific event stream identified by a (scheme/value) pair with a specific `dispatch_mode`, either **on-start** or **on-receive**, as described in subclause A.13.7.

The processing model varies depending on *dispatch_mode*.

- Common process
 - The DASH Player implements the following process:
 - Parse the '`emsg`'/timed metadata sample and retrieve *scheme_uri/(value)*.
 - If Application is not subscribed to the *scheme_uri/(value)* pair, end the processing of this '`emsg`'.
- on-receive processing
 - The DASH Player implements the following process when *dispatch_mode* = *on-receive*:
 - Dispatch the event/timed metadata, including *ST*, *id*, *DU*, *timescale*, and *message_data* as described in subclause A.13.6.
- on-start processing
 - The DASH Player sets up an Active Event Table for each subscribed *scheme_uri/(value)* in the case of *dispatch_mode* = *on_start*. The **Active Event Table** maintains a single list of '`emsg`' *ids* that have been dispatched.
 - The DASH Player implements the following process when *dispatch_mode* = *on_start*:
 - Derive the event instance/metadata sample's *ST*
 - If the current media presentation time value is smaller than *ST*, then go to step v.
 - Derive the ending time $ET = ST + DU$.
 - If the current presentation time value is greater than *ET*, then end processing.
 - In the case of event: Compare the event's *id* with the entries of the Active Event Table of the same *scheme_uri/(value)* pair:

- If an entry with the identical *id* value exists, end processing;
- If not, add 'emsg' 's *id* to the corresponding the Active Event Table.
- Dispatch the event/metadata message_data at time ST, or immediately if the current presentation time is larger than ST, as described in subclause A.13.6.

A.13.9 The event/metadata buffer model

Along with the media samples, the event instances and timed metadata samples are buffered. The event/metadata buffer should be managed with the same scheme as the media buffer, i.e. as long as a media sample exists in the media buffer, the corresponding events and/or metadata samples should be maintained in the event/metadata buffer.

A.13.10 Prose description of APIs

The event/timed metadata API is an interface defined between a DASH client and a device application in the exchange of subscription data and dispatch/transfer of matching DASH Event or timed metadata information between these entities. The Event/timed metadata API is shown in Figure A.1.

NOTE 1 In this Annex, the term "DASH Player" is used.

The description of the API below is strictly functional, i.e. implementation-agnostic. For example, the `subscribeEvent()` method as defined below may be mapped to the existing `on(type, listener, scope)` method as defined for the `dash.js` under `MediaPlayerEvents`.

As part of this API and before any operations, the DASH Player provides a list of *scheme_id/(value)* listed in the MPD when it receives it. This list includes all MPD and inband events as well as *scheme_id* of all timed metadata tracks. At this point, the Application is aware of the possible events and metadata deliverable by the DASH Player.

The subscription state diagram of a DASH Player associated with the API is shown below in Figure A.6.

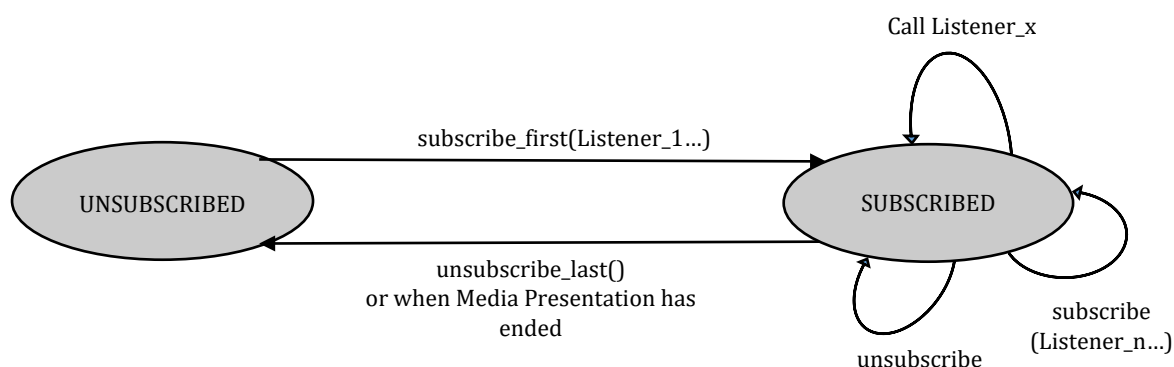


Figure A.6 State Diagram of DASH Player for the event/timed metadata API.

The scope of the above state diagram is the entire set of applicable events/timed metadata streams being subscribed/unsubscribed, i.e. it is not indicating the state model of DASH Player in the context of a single Event/timed metadata stream subscription/un-subscription.

The application subscribes to the reception of the desired event/timed metadata and associated information by the `subscribeEvent()` method. The parameters to be passed in this method are:

- *scheme_uri* – The scheme identifier for the event stream being subscribed to. This must be one returned by the list of events the DASH player supplied. By setting this value to

urn:mpeg:dash:event:catchall:2020, the Application may subscribe to all existing events and metadata schemes described in the MPD. In this case, the value of *value* is irrelevant.

- *value* – A value of the event or timed metadata stream within the scope of the above *scheme_uri*, optional to include. When not present, no default value is defined – i.e., no filtering criterion is associated with the Event scheme identification.
- *dispatch_mode* – Indicates when the event handler function identified in the *callback_function* argument should be called:
 - *dispatch_mode* = *on_receive* – provide the event/timed metadata sample data to the Application as soon as it is detected by DASH Player;
 - *dispatch_mode* = *on_start* – provide the event/timed metadata sample data to the App at the start time of Event message or at the presentation time of timed metadata sample.
 - The default mode for *dispatch_mode* should be set to *on_receive*, i.e. if the *dispatch_mode* is not passed during the *subscribe_first* operation, DASH Player should assume *dispatch_mode* = *on_receive* for that specific subscription.
- *callback_handler* – the name of the function to be (asynchronously) called for an event corresponding to the specified *scheme_uri/(value)*. The callback function is invoked with the arguments described below.

NOTE 2 This document does not include any explicit signalling for the desired dispatch mode in MPD or timed metadata track. In the current design, the Application relays its desired dispatch mode to DASH Player when it subscribes to an event stream or timed metadata track. In this approach, the scheme owner should consider the dispatch mode as part of the scheme design and define whether any specific dispatch mode should be selected during the design of the scheme.

Upon successful execution of the event/timed metadata subscription call, the DASH Player shall monitor the source of potential Event stream information, i.e., the MPD or incoming DASH Segments, for matching values of the subscribed *scheme_uri/(value)*. The parentheses around *value* are because this parameter may be absent in the event/timed metadata subscription call. When a matching event/metadata sample is detected, DASH Player invokes the function specified in the callback Function argument with the following parameters. It should additionally provide to the Application the current presentation time at DASH Player when performing the dispatch action. The parameters to be passed in this method are shown in Table A.2.

Table A.2 Event/timed metadata API parameters and datatypes

API Parameter	MPD event	Inband 'emsg'	Metadata	Data Type	'on-receive'	'on-start'
scheme_id	EventStream @schemeIdUri	scheme_id_uri	timed metadata track URI	string	Y	Y
value	EventStream @value	value		string	Y	Y
presentation_time	Event @presentationTime	presentation_time	timed metadata sample presentation time	unsigned int(64) in milliseconds	Y	N
duration	Event @duration	event_duration	timed metadata sample duration	unsigned int(32) in milliseconds	Y	N
id	Event @id	id		unsigned int(32)	Y	N
message_data	Event @messageData	message_data[]	timed metadata sample data in mdat	unsigned int(8) x messageSize	Y	Y
Y= Yes, N= NO, O= Optional						

When the duration of the event is unknown, the variable *duration* shall be set to its maximum value (xFFFFFFFF = 4,294,967,295).

NOTE 3 In the case of 'emsg' version 0, DASH Player is expected to calculate *presentation_time* from *presentation_time_delta*.

To remove a listener the `unsubscribeEvent()` function is called with the following arguments:

- *scheme_uri* - A unique identifier scheme for the associated DASH Event stream of interest to the Application.
- *value*
- *callback_handler*

If a specific listener is given in the *callback_function* argument, then only that listener is removed for the specified *scheme_uri*/(*value*). Omitting or passing null to the *callback_function* argument would remove all event listeners for the specified *scheme_uri*/(*value*).

A.13.11 Detailed processing

As shown in Figure A.1, the event/metadata buffer holds the events or metadata samples to be processed. We assume that this buffer has the same data structure to hold events or metadata. Table A.3 is used to define this Event/Metadata Internal Object (EMIO).

Table A.3 The Event/Metadata Internal Object (EMIO)

```

event-metadata-internal-object {
    string          scheme_id_uri;
    string          value;
    unsigned int(32) presentation_time;
    unsigned int(32) duration;
    unsigned int(32) id;
    unsigned int(8)  message_data();
}

```

The process for converting the received event/metadata sample to EMIO is as following:

1. For MPD event
 - a. For each period
 - i. Parse each **EventStream**
 - ii. Get **Eventstream** common parameters
 - iii. For each Event Stream:
 1. Parse each event
 2. For each event
 - a. Calculate presentation time and event duration
 - b. Add it to EMIO
2. For inband event
 - a. For each Segment
 - i. Parse event boxes as well as 'moof'
 - ii. Calculate EPT of segment
 - iii. For each event:
 1. Map 'emsg' box parameters to EMIO
3. For simple metadata samples
 - a. For each Segment
 - i. Parse 'moof'
 - ii. For each sample:
 1. Parse the format
 2. map the data to EMIO

A.13.12 Recommended dispatch modes for DASH-specific events

In subclause 5.10.4, several DASH-specific events are defined. For all DASH-specific events, it is proposed to use the on-receive dispatch mode.

Annex B
(normative)

MPD schema

The schema of the MPD for this document is provided at <https://standards.iso.org/iso-iec/23009/-1/ed-5/en> (DASH-MPD.xsd).

Annex C

(normative)

MIME type registration for MPD and other resources

C.1 General

This annex provides the formal MIME type registration for the MPD and other resources. It is referenced from the registry at <http://www.iana.org/>.

C.2 MIME type and subtype

The MIME Type and Subtype are defined as follows:

- MIME media type name: `application`
- MIME subtype name: `dash+xml`
- Required parameters: `None`
- Optional parameters: The 'profiles' parameter as documented in Annex C.3.
- Encoding considerations: `UTF-8`

- Security considerations:

The MPD is a Media Presentation Description and contains references to other resources. It is coded in XML, and there are risks that deliberately malformed XML can cause security issues. In addition, an MPD can be authored that causes receiving clients to access other resources; if widely distributed, this can be used to cause a denial-of-service attack.

The Media Presentation Description (MPD) format does not incorporate any active or executable content. However, other forms of material from outside sources can be referenced by an MPD, and this material can contain active or executable content. Such material is expected to be identified by its own MIME type, and the security considerations of that format should be taken into account.

If operating in an insecure environment and required by the content/service provider, elements and attributes of MPD may be encrypted to protect their confidentiality by using the syntax and processing rules specified in the W3C Recommendation “XML Encryption Syntax and Processing”.

If operating in an insecure environment and required by the content/service provider, the digital signing and verification procedures specified in the W3C Recommendation “XML Signature Syntax and Processing” may be used to protect data origin authenticity and integrity of the MPD.
- Interoperability considerations:

The specification defines a platform-independent expression of a presentation, and it is intended that wide interoperability can be achieved.
- Published specification:

ISO/IEC 23009-1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*
- Applications which use this media type:

Various
- Additional information:
 - File extension(s):

mpd
 - Intended usage:

common
- Other information/general comment:

None
- Author/Change controller:

ISO/IEC JTC1/SC29 (MPEG)

C.3 Profiles parameters

Parameter name: `profiles`

Parameter value: The 'profiles' parameter is an optional parameter that indicates one or more profiles to which the file claims conformance. The contents of this attribute shall conform to either the `pro-simple` or `pro-fancy` productions of IETF RFC 6381:2011, subclause 4.5. The profile identifiers reported in the MIME type parameter should match identically the profiles reported in the profiles attribute in the MPD itself (see Clause 8).

EXAMPLE

```
application/dash+xml;profiles="urn:mpeg:dash:profile:full:2011,urn:3GPP:PS:profile:DASH10"
```

C.4 MPD Anchors

C.4.1 General

URIs for resources with MIME type `application/dash+xml` may use URI fragment syntax to start a presentation at a given time and a given state.

An MPD anchor is a set of Representations being presented and a time offset from the start of a period on the media timeline. These are expressed using URI fragment syntax. This annex defines one temporal parameter, position, and two context parameters, state and selection, in order to express the state of a DASH media presentation.

URI fragment starts with the '#' character, and is a string terminating the URI. MPD fragments shall be an ampersand-separated list of key=value pairs, with syntax and semantics of key and value parameters defined in Table C.1 of C.4.2.

C.4.2 Parameters

Table C.1 — Parameters for MPD Anchors

Key	Value	Semantics
T	Time or time range which shall be in the same format as defined in W3C Media Fragments URI 1.0 (basic) including validity rules and recommended behaviour. Optionally, prefixed comma-separated pair of numbers. See C.4.4 for validity rules and recommended behaviour.	<p>If the parameter starts from an integer, it signifies the time since the beginning of the period indicated by the period parameter.</p> <p>If the <code>t</code> parameter is not present, its default value is <code>t=0</code> (i.e. start from the beginning of the Period).</p> <p>NOTE If <code>period</code> parameter is not present, the default Period is the first period of the presentation.</p> <p>If the parameter starts from prefix <code>posix:</code> it signifies the absolute time range defined in seconds of Coordinated Universal Time (ITU-R TF.460-6). This is the number of seconds since 01-01-1970 00:00:00 UTC. Fractions of seconds may be optionally specified down to the millisecond level.</p> <p>The <code>posix</code> notation documents the absolute time in the MPD.</p> <p>This notation shall only be used if MPD@availabilityStartTime is present.</p> <p>This <code>t=posix:xxx</code> notation parameter shall not be used if a <code>period</code> parameter is used. In addition, at most one of the two parameters, <code>period</code> and <code>t</code> shall be present in an anchor.</p> <p>A special value "now" indicates the latest available segment, i.e. "live edge".</p>
period	String	Value of a Period parameter <code>Period@id</code> . If <code>period</code> parameter is not present, the default value of the <code>@id</code> attribute value of the Period with the earliest <i>PeriodStart</i> .
track	string	Value of a single <code>AdaptationSet@id</code>
group	string	Value of a single <code>AdaptationSet@group</code>

Percent coding, defined in IETF RFC 5986, shall be used for all reserved characters in parameter values.

NOTE Ability to address elements in the MPD depends on the `Period@id`, `AdaptationSet@id` and **`AdaptationSet@group`**. Hence MPD authors are encouraged to put these attributes explicitly into the MPD if they intend to make MPDs addressable.

C.4.3 Examples

42nd second of Period1 `my.mpd#t=42&period=Period1`

`my.mpd#t=42&period=Period1`

42nd second from the start of the presentation, English 5.1 audio and video

`my.mpd#t=42&track=en51&track=vid`

A range from 60s to 180s of the presentation

```
my.mpd#t=60,180
```

Start a live stream at Wed, 21 Jan 2015 20:04:05 GMT

```
my.mpd#t=posix:1421870645
```

Start a live stream at Wed, 21 Jan 2015 20:04:05 GMT, English 5.1 audio and video

```
my.mpd#t=posix:1421870645&track=en51&track=vid
```

A live stream range from Wed, 21 Jan 2015 20:04:05 GMT to Wed, 21 Jan 2015 23:44:33 GMT

```
my.mpd#t=posix:1421870645,1421883873
```

Play the stream from the latest available segment to Wed, 21 Jan 2015 23:44:33 GMT

```
my.mpd#t=posix:now,1421883873
```

Play the stream from the earliest available segment to Wed, 21 Jan 2015 23:44:33 GMT

```
my.mpd#t=posix:0,1421883873
```

C.4.4 Handling UTC parameter

The following notation is used in this clause:

- *N* is the UTC time at the moment the URL is requested;
- *E* is the earliest available segment time in the current presentation at time *N*;
- *F* is the latest availability segment time in the current presentation;
- *S* is the start time in the `posix` anchor;
- *T* is the end time in the `posix` anchor.

If *T* is not specified, or larger than *F*, its value shall be considered to be equal to *F*.

If *S* is not specified, its value shall be considered to be equal to *E*. If *S* is “now”, its value is *N*.

C.5 MPD patch MIME type

The MIME Type and Subtype of the MPD Patch document as introduced in 5.15 are defined as follows:

- MIME media type name: `application`
- MIME subtype name: `dash-patch+xml`
- Required parameters: `None`
- Optional parameters: `None`
- Encoding considerations: `8 bit. This media type always uses UTF-8.`

- Security considerations:

MPD patches are based on the Patch Operations Framework defined in RFC 5261 and thus inherit the security considerations of that framework.

In addition, the MPD patch is a patch to Media Presentation Description document and contains references to other resources. It is coded in XML, and there are risks that deliberately malformed XML can cause security issues. In addition, an MPD can be authored that causes receiving clients to access other resources; if widely distributed, this can be used to cause a denial-of-service attack.

The MPD patch format does not incorporate any active or executable content. However, other forms of material from outside sources can be referenced by an MPD, and this material can contain active or executable content. Such material is expected to be identified by its own MIME type, and the security considerations of that format should be taken into account.

If operating in an insecure environment and required by the content/service provider, elements and attributes of MPD may be encrypted to protect their confidentiality by using the syntax and processing rules specified in the W3C Recommendation “XML Encryption Syntax and Processing”.

If operating in an insecure environment and required by the content/service provider, the digital signing and verification procedures specified in the W3C Recommendation “XML Signature Syntax and Processing” may be used to protect data origin authenticity and integrity of the MPD.

The MPD Patch schema allow the inclusion elements and attributes from other namespaces. Additionally, a patch could be used to introduce material in other namespaces into the MPD itself. Such material, if processed as part of the patch or MPD, raises additional security considerations.
- Interoperability considerations:

The specification defines a platform-independent expression of a presentation, and it is intended that wide interoperability can be achieved.
- Published specification:

ISO/IEC 23009-1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*
- Applications which use this media type:

Various
- Additional information:

File extension(s): mpp

Intended usage: common

- Other information/general comment: None
- Author/Change controller: ISO/IEC JTC1/SC29 (MPEG)

Annex D (normative)

DASH Metrics

D.1 General

This annex defines the ISO/IEC 23009-1 DASH Metrics. The normative aspects of the annex are defined in D.4, namely the semantics of the metrics and the associated keys to be used for requesting the collection of the metrics. The client reference model in D.2 and the observation points in D.3 serve as background information.

D.2 DASH-Metrics client reference model

The DASH-Metrics client reference model is depicted in highlighting so-called observation points (OPs) as defined in D.3.

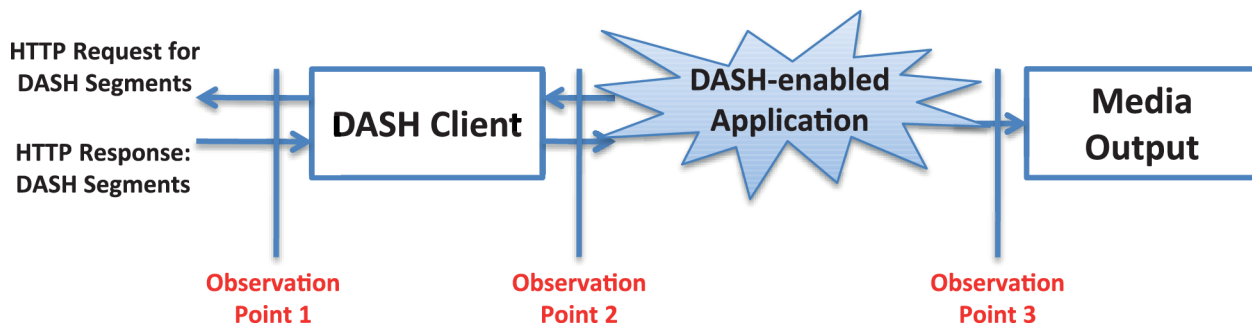


Figure D.1 — DASH-Metrics client reference model

The *DASH access client* as defined in 4.2 issues HTTP requests (for DASH data structures), and receives HTTP request responses (containing DASH data structures). Data structures may typically be MPDs, Segments or partial Segments. This input/output interface from the network towards the DASH Client is referred to as observation point 1 (OP1).

Furthermore, the DASH Client delivers encoded media samples to the *DASH-enabled application* for further processing and may receive also commands from it. This input/output interface of the DASH Client towards the DASH-enabled application is referred to as observation point 2 (OP2).

NOTE Further processing can include de-multiplexing (of audio/video) and/or decoding potentially involving several buffers.

Finally, the DASH-enabled application delivers decoded media samples to the *media output*, which displays the media to the user. This output interface towards the user is referred to as observation point 3 (OP3).

D.3 Definition of observation points

D.3.1 General

This clause defines the observation points as depicted in Figure D.1.

D.3.2 Observation point 1

The observation point 1 (OP1) is defined as:

- a set of TCP connections each defined by its destination IP address, initiation, connect and close times;
- a sequence of transmitted HTTP requests, each defined by its transmission time, contents, and the TCP connection on which it is sent; and
- for each HTTP response, the reception time and contents of the response header and the reception time of each byte of the response body.

NOTE The contents of the response body is fully defined by the contents of the request and response headers.

D.3.3 Observation point 2

The observation point 2 (OP2) consists of encoded media samples. Each encoded media sample is defined as:

- media type;
- decoding time;
- presentation time;
- the @id of the Representation from which the sample is taken; and
- the delivery time.

D.3.4 Observation point 3

The observation point 3 (OP3) consists of decoded media samples. Each decoded media sample is defined as:

- the media type;
- the presentation timestamp of the sample (media time);
- the actual presentation time of the sample (real time); and
- the @id of the Representation from which the sample is taken (the highest dependency level if the sample was constructed from multiple Representations).

D.4 Semantics of the DASH metrics

D.4.1 General

This subclause provides the general QoE metric definitions and measurement framework.

The semantics are defined using an abstract syntax. Items in this abstract syntax have one of the following primitive types (*Integer*, *Real*, *Boolean*, *Enum*, *String*) or one of the following compound types:

- *Objects*: an unordered sequence of (*key*, *value*) pairs, where the key always has string type and is unique within the sequence.
- *List*: an ordered list of items.
- *Set*: an unordered set of items.

Additionally, there are two kinds of timestamp defined, i.e. *real time* (wall-clock time) with type *Real-Time* and *media time* with type *Media-Time*.

Where lists are defined, the name '*entry*' is used to define the format of each entry, but since lists contain unnamed entries, this name would not appear in any concrete syntax.

Each metric is defined as a named list of entries that logically contains the metric information for the entire Media Presentation. Reporting of these lists, whether done at the end of the Media Presentation or incrementally during the Media Presentation, is outside the scope of this document.

D.4.2 TCP connections

Table D.1 contains the metric defining the list of TCP connections. The key in Table D.1 shall be used to refer to the metric as defined in Table D.1.

Table D.1 — List of TCP connections

Key			Type	Description
TcpList			List	List of HTTP request/response transactions
	<i>Entry</i>		Object	An entry for a single HTTP request/response
		tcpid	Integer	Identifier of the TCP connection on which the HTTP request was sent.
		dest	String	IP Address of the interface over which the client is receiving the TCP data.
		topen	Real-Time	The time at which the connection was opened (sending time of the initial SYN or connect socket operation).
		tclose	Real-Time	The time at which the connection was closed (sending or reception time of FIN or RST or close socket operation).
		tconnect	Integer	Connect time in ms (time from sending the initial synchronize message (SYN) to receiving the acknowledgement (ACK) or completion of the connect socket operation).

D.4.3 HTTP request/response transactions

Table D.2 contains the metric defining the List of HTTP Request/Response Transactions. The key in Table D.2 shall be used to refer to the metric as defined in Table D.2.

Table D.2 — List of HTTP request/response transactions

Key				Type	Description
HttpList				List	List of HTTP request/response transactions
	Entry			Object	An entry for a single HTTP request/response
		tcpid		Integer	Identifier of the TCP connection on which the HTTP request was sent.
		type		Enum	<p>This is an optional parameter and should not be included in HTTP request/response transactions for progressive download.</p> <p>The type of the request:</p> <ul style="list-style-type: none"> — MPD — XLink expansion — Initialization Segment — Index Segment — Media Segment — Bitstream Switching Segment — Remote xlink element — MPD patch — other
		url		String	The original URL (before any redirects or failures)
		actualurl		String	The actual URL requested, if different from above
		range		String	The contents of the byte-range-spec part of the HTTP Range header.
		trequest		Real-Time	The real time at which the request was sent.
		tresponse		Real-Time	The real time at which the first byte of the response was received.
		responsecode		Integer	The HTTP response code.
		interval		Integer	The duration of the throughput trace intervals (ms), for successful requests only.
		trace		List	Throughput trace, for successful requests only.
			Entry	Object	A single throughput measurement entry.
			s	Real-Time	Measurement period start.
			d	Integer	Measurement period duration (ms).
			b	List	List of integers counting the bytes received in each trace interval within the measurement period.

NOTE 1 Information additional to that specified in the type can be returned, for example if a client makes a request for a initialization information from a self-initializing Media Segment then Segment Index may also be received.

NOTE 2 All entries for a given object have the same URL and range and so can easily be correlated. If there were redirects or failures, there would be one entry for each redirect/failure. The redirect-to URL or alternative url (where multiple have been provided in the MPD) appears as the `actualurl` of the next entry with the same url value.

NOTE 3 The periods in *Entry* are expected to be those periods where the client was actively reading from the TCP connections (i.e. they are expected to not include periods where the TCP connection is idle due to zero receive window).

The end of the last measurement period in the `trace` shall be the time at which the last byte of the response was received.

The `interval` and `trace` shall be absent for redirect and failure records.

The key `HttpList(n, type)` where *n* is a positive integer is defined for an `HttpList` with an interval of *n* ms and *type* is one of `MPD`, `XLinkExpansion`, `InitializationSegment`, `MediaSegment`, `IndexSegment` `BitstreamSwitchingSegment` or other. If *type* is not present, all HTTP transactions are requested to be collected. If *type* is present, it specifies that the HTTP transactions concerning a resource equal to *type* are requested to be collected. Multiple keys `HttpList(n, type)` with different values of *n* and *type* may be present for a single `@metrics` attribute value.

An HTTP transaction that is not finished within a QoE metric collection period shall not be included in the metrics.

D.4.4 Representation switch events

Table D.3 defines the metric for Representation switch events. The key in Table D.3 shall be used to refer to the metric as defined in Table D.3.

Table D.3 — List of Representation switch events

Key			Type	Description
RepSwitchList			List	List of Representation switch events (a switch event is the time at which the first HTTP request for a new Representation, that is later presented, is sent)
	Entry		Object	A Representation switch event.
		t	Real-Time	Time of the switch event.
		mt	Media-Time	The media presentation time of the earliest access unit (out of all media content components) played out from the “to” Representation.
		to	String	value of Representation@id identifying the switch-to Representation.
		lto	Integer	If not present, this metrics concerns the Representation as a whole. If present, lto indicates the value of SubRepresentation@level within Representation identifying the switch-to level of the Representation.

D.4.5 Buffer level

Table D.4 defines the metric for buffer level status events. The key in Table D.4 shall be used to refer to the metric as defined in Table D.4.

Table D.4 — List of buffer level

Key			Type	Description
BufferLevel			List	List of buffer occupancy level measurements during playout at normal speed.
	Entry		Object	One buffer level measurement.
		t	Real-Time	Time of the measurement of the buffer level.
		level	Integer	Level of the buffer in milliseconds. Indicates the playout duration for which media data of all active media components is available starting from the current playout time.

The key is `BufferLevel(n)`, where n is a positive integer defined to refer to the metric in which the buffer level is recorded every n ms.

D.4.6 Play list

Decoded samples are generally rendered in presentation time sequence, each at or close to its specified presentation time. A compact Representation of the information flow can thus be constructed from a list of time periods during which samples of a single Representation were continuously rendered, such that each was presented at its specified presentation time to some specific level of accuracy (e.g. ± 10 ms).

Such a sequence of periods of continuous delivery is started by a user action that requests playout to begin at a specified media time (this can be a "play", "seek" or "resume" action) and continues until playout stops either due to a user action, the end of the content, or a permanent failure.

Table D.5 defines the play list event metric. The key in Table D.5 shall be used to refer to the metric as defined in Table D.5.

Table D.5 — Play list

Key			Type	Description
PlayList			List	A list of playback periods. A playback period is the time interval between a user action and whichever occurs soonest of the next user action, the end of playback or a failure that stops playback.
	Entry		Object	A record of a single playback period.
		start	Real-Time	Timestamp of the user action that starts the playback period.
		mstart	Media-Time	The presentation time at which playout was requested by the user action.
		starttype	Enum	Type of user action which triggered playout:

Key				Type	Description
					<ul style="list-style-type: none"> — NewPayout - New payout request (initial payout) — UserSeek - User seek request — Resume - Resume from pause — OtherRequest - Other user request (e.g. user-requested quality change) — StartOfCollection - Start of a metrics collection period (hence earlier entries in the play list not collected) — SpeedChange - User-requested change in playspeed, other than resume from paused
		trace		List	List of periods of continuous rendering of decoded samples.
		<i>Entry</i>		Objects	Single entry in the list.
			representatio nid	String	The value of the Representation@id of the Representation from which the samples were taken.
			subreplevel	Integer	If not present, this metrics concerns the Representation as a whole. If present, subreplevel indicates the greatest value of any Subrepresentation@level being rendered.
			start	Real-Time	The time at which the first sample was rendered.
			mstart	Media-Time	The presentation time of the first sample rendered.
			duration	Integer	The duration of the continuously presented samples (which is the same in real time and media time). "Continuously presented" means that the media clock continued to advance at the playout speed throughout the interval.
			playbackspeed	Real	The playback speed relative to normal playback speed (i.e. normal forward playback speed is 1.0).
			stopreason	Enum	The reason why continuous presentation of this Representation was stopped. Either:

Key					Type	Description
						<ul style="list-style-type: none"> — RepresentationSwitch - Representation switch (not relevant in case of progressive download) — Rebuffering - Rebuffering — UserRequest - User request — EndOfPeriod - End of Period — EndOfContent - End of content — EndOfCollection - End of a metrics collection period — Failure - Failure — ClientPlaybackAction - Client-initiated change in playback (e.g. seek or change in playspeed), other than pausing to rebuffer

NOTE The trace can include entries for different representations that overlap in time, because multiple representations are being rendered simultaneously, for example one audio and one video Representation.

D.4.7 Device information

This metric contains information about the displayed video resolution as well as the physical screen characteristics. If the video is rendered in full-screen mode, the video resolution usually coincides with the characteristics of the full physical display. If the video is rendered in a smaller subwindow, the characteristics of the actual video window shown are logged.

If known by the DASH Client, the physical screen width and the horizontal field-of-view are logged.

The metric is logged at the start of each QoE reporting period, and whenever the characteristics change during the session (for instance if the device is rotated from horizontal to vertical orientation, or if the subwindow size is changed).

Table D.6 defines the device information metrics. If an individual metric cannot be logged, its value is set to 0 (zero).

Table D.6 – Device information

Key			Type	Description
DeviceInformationList			List	A list of device information objects.
	Entry		Object	A single object containing new device information.
		start	Real-Time	Timestamp when the device information was logged.
		mstart	Media-Time	The presentation time at which the device information was logged.
		videoWidth	Integer	The width of the displayed video, in screen pixels (not encoded video pixels).
		videoHeight	Integer	The height of the displayed video, in screen pixels (not encoded video pixels)
		screenWidth	Integer	The total width of the screen, in screen pixels
		screenHeight	Integer	The total height of the screen, in screen pixels
		pixelWidth	Real	The width of a screen pixel, in millimetres
		pixelHeight	Real	The height of a screen pixel, in millimetres
		fieldOfView	Real	The actual or estimated horizontal angle subtended at the eye by the screen, measured in degrees.

Annex E (normative)

Byte range requests with regular HTTP GET methods

E.1 Background

There are deployment environments where HTTP partial GET is not supported, or results in the return of the entire, rather than partial target. This represents a problem for DASH Clients. It is expected that these problems gradually disappear, but until this will be the case, a method is provided to not exclude DASH Clients operating in these environments and service providers wanting to support such clients are excluded from using this DASH standard to deploy media streaming services using the formats defined in this document.

To address these requirements, the **BaseURL@byteRange** attribute may be present. If present, it provides indication that resources offered in the MPD that are requested by a HTTP partial GET (e.g. Segments for which HTTP-URLs contain byte ranges or Subsegments) may also be requested using a regular HTTP GET and mapping the information that is otherwise added in the **Range** header in case of a HTTP partial GET into the request URI of a regular HTTP GET request. It is expected that DASH Clients only use this method if HTTP partial GET requests fail. If DASH Clients only have this alternative to request segments or Subsegments, then it is expected that they request single units of segments or Subsegments.

E.2 Construction rule

The **BaseURL@byteRange** attribute represents a template that may be used to construct a URL requesting a byte range (a “byte range URL”) from a resource, given the original URL of the resource and the required byte range. The result of issuing a GET request to this byte range URL without including the HTTP Range header should be identical to the result of requesting the original URL with the byte range specified in the HTTP Range header.

The **BaseURL@byteRange** contains a template string that contains one or more of the identifiers as listed in Table E.1. The string shall contain identifiers `$first$` and `$last$` as specified in Table E.1.

The byte range URL shall be constructed from the template string by substituting the identifiers specified in the first column of Table E.1 with the values specified in the second column of Table E.1. If the `$query$` identifier is not present in the template and the `query` portion of the original URL as defined in IETF RFC 3986 is not empty, then the string “?” `query` shall be appended to the constructed URL.

If the template string contains unrecognized identifiers, then the result of the URL construction is unspecified. In this case, it is expected that the DASH Client ignores the entire containing **ByteRange** element and the processing of the MPD continues as if this **ByteRange** element was not present.

Strings outside identifiers shall only contain characters that permit to form a valid HTTP-URL according to IETF RFC 3986.

Table E.1 — Identifiers for Byte Range Templates

\$<Identifier>\$	Substitution parameter
\$ \$	Is an escape sequence, i.e. "\$\$" is replaced with a single "\$"
\$base\$	The identifier shall be substituted by the <code>scheme</code> <code>:"</code> <code>hier-part</code> of the original URL as defined in IETF RFC 3986.
\$query\$	The identifier shall be substituted by the <code>query</code> part of the original URL as defined in IETF RFC 3986. If the <code>query</code> part of the original URL is empty then inclusion of this identifier in the template shall cause removal of the separator character immediately preceding the <code>\$query\$</code> identifier in the template string if that character is not the "?" character, or, otherwise, the separator character immediately following the <code>\$query\$</code> identifier if present.
\$first\$	The identifier shall be substituted by the byte offset of the first byte in a range and shall be identical to the value of <code>'first-byte-pos'</code> of <code>'byte-range-spec'</code> of IETF RFC 7231:2014, 2.1, if this request would be executed using a partial GET request.
\$last\$	The identifier is substituted by the byte offset of the last byte in the range; that is, the byte positions specified are inclusive. It shall be identical to the value of <code>'last-byte-pos'</code> of <code>'byte-range-spec'</code> of IETF RFC 7231:2014, 2.1, if this request would be executed using a partial GET request.

E.3 Examples

NOTE The URL examples below are intentionally not valid, resolvable URLs and are just examples.

Original URL	<code>http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926</code>
Byte Range	<code>1876-23456</code>
BaseURL@byteRange	<code>\$base\$/range/\$first\$-\$last\$</code>
Byte range URL	<code>http://cdn.example.com/movies/134532/audio/en/aac64.mp4/range/1876-23456?token=8787r08f2gf087g28gf926</code>

Original URL	<code>http://cdn.example.com/movies/134532/audio/en/aac64.mp4</code>
Byte Range	<code>1876-23456</code>
BaseURL@byteRange	<code>\$base\$?\$query\$&range=\$first\$-\$last\$</code>
Byte range URL	<code>http://cdn.example.com/movies/134532/audio/en/aac64.mp4?range=1876-23456</code>

Original URL	http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926
Byte Range	1876-23456
BaseURL@byteRange	\$base\$?\$query\$&range=\$first\$-\$last\$
Byte range URL	http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926&range=1876-23456

Annex F (informative)

Guidelines for extending DASH with other delivery formats

F.1 Adding delivery formats to DASH

In order to support use with DASH, a delivery format should have the property that decoding and playback of any portion of the media can be achieved using a subset of the media that is only a constant amount larger than the portion of the media to be played.

For example, a delivery format following this property is one for which the media is stored as a header followed by a sequence of small blocks, with the property that any block can be decoded and played out given only that block and the header. The definition of these blocks and the mapping to the Subsegments in this document are encouraged. A Subsegment may be defined as a contiguous time interval of a Segment and a contiguous byte range of a Segment for which no overlap in both dimensions with any other Subsegment in the Segment exists.

Additionally, it is desirable that the delivery format supports some kind of “index” which enables the byte range within the Segment corresponding to any given time range to be efficiently discovered. A suitable unit is the indexing of Subsegments. It should be possible to discover the position in the Segment of the index without downloading the whole Segment. The position of the index may also be advertised in the MPD or the index may be provided as a separate Index Segment. The Segment Index ('*sid*x') or Subsegment Index ('*ss**ix*'), both defined in ISO/IEC 14496-12, may serve as a starting point and/or may be directly applied to any other media format.

F.2 Media Presentation authoring rules

A specification for how to use a media container format with DASH should include:

- Definition of the MIME type for the Representation as a concatenation of Segments.
- Description of either a self-initializing Media Segment or the combination of an Initialization Segment and a Media Segment format.

In addition, the specification may further define:

- Index Segments;
- Bitstream switching segments;
- Interpretation of a stream access point (SAP), potentially different types as defined in subclause 4.5.2 in the context of the media container format. (The SAP types are fully *defined* in ISO/IEC 14496-12 and should not be re-defined, but the *interpretation* of those definitions in media-container-specific language may be necessary);
- Container-format-specific semantics for the @bitstreamSwitching, @segmentAlignment and @subsegmentAlignment. These should align with the definitions in this document.

Representation attributes present in the MPD may also be repeated in the media itself, e.g. in an Initialization Segment or a Media Segment. The media content should be provided such that no mismatch between these two values occurs. If it does, the value in the media itself is expected to take precedence over values expressed in the MPD, especially when used in the media decoding process.

Annex G (informative)

MPD Examples and MPD Usage

All examples documented in this annex are provided at <https://standards.iso.org/iso-iec/23009/-1/ed-5/en> with the following file naming convention:

example-G<subsection>[<order in subsection>].mpd

G.1 Example MPD for ISO Base media file format On Demand profile

This subclause provides a simple example for a static presentation with self-initializing Media Segments, multiple languages, subtitles, content protection and multiple base URLs. This MPD document describes content available from two sources (cdn1 and cdn2) that has audio available in English or French at rates of 64 kbits and 32 kbits and subtitles in German. Six versions of the video are provided at bitrates between 256 kbit/s and 2 Mbit/s in different spatial resolutions. Content protection is applied.

The Media Presentation complies with the ISO Base media file format On Demand profile as defined in subclause 8.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT1.2S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

  <Period>
    <!-- English Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="en" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Representation id="1" bandwidth="64000">
        <BaseURL>7657412348.mp4</BaseURL>
      </Representation>
      <Representation id="2" bandwidth="32000">
        <BaseURL>3463646346.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- French Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40.2" lang="fr"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
      <Representation id="3" bandwidth="64000">
        <BaseURL>3463275477.mp4</BaseURL>
      </Representation>
      <Representation id="4" bandwidth="32000">
        <BaseURL>5685763463.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
  <!-- Timed text -->

```



```

    <AdaptationSet mimeType="application/ttml+xml" lang="de">
      <Role schemeIdUri="urn:mpeg:dash:role" value="subtitle"/>
      <Representation id="5" bandwidth="256">
        <BaseURL>796735657.xml</BaseURL>
      </Representation>
    </AdaptationSet>
    <!-- Video -->
    <AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228" subsegmentAlignment="true"
subsegmentStartsWithSAP="2">
      <ContentProtection schemeIdUri="urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
      <Representation id="6" bandwidth="256000" width="320" height="240">
        <BaseURL>8563456473.mp4</BaseURL>
      </Representation>
      <Representation id="7" bandwidth="512000" width="320" height="240">
        <BaseURL>56363634.mp4</BaseURL>
      </Representation>
      <Representation id="8" bandwidth="1024000" width="640" height="480">
        <BaseURL>562465736.mp4</BaseURL>
      </Representation>
      <Representation id="9" bandwidth="1384000" width="640" height="480">
        <BaseURL>41325645.mp4</BaseURL>
      </Representation>
      <Representation id="A" bandwidth="1536000" width="1280" height="720">
        <BaseURL>89045625.mp4</BaseURL>
      </Representation>
      <Representation id="B" bandwidth="2048000" width="1280" height="720">
        <BaseURL>23536745734.mp4</BaseURL>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>

```

G.2 Example for ISO Base media file format Live profile

This subclause provides a simple example for a dynamic presentation, with multiple languages, multiple base URLs, multiple video bitrates, and segments about two seconds in length for low latency from live programming. At the time this MPD was fetched, 432 Segments of the dynamic presentation were available so the wall clock time has been approximately 2011-12-25T12:44:24 UTC. All the video Segments are aligned and start with a Stream Access Point. All the audio Segments are aligned so language switching can be done with the non-language sound (e.g. music) seamlessly.

In this MPD, assuming that the first **BaseURL** element and the video Representation with id "v1" is selected, and template results in `http://cdn1.example.com/video/500000/$Time$.mp4v`, the segment list starting at number 0 results in

```

http://cdn1.example.com/video/500000/0.mp4v
http://cdn1.example.com/video/500000/180180.mp4v
http://cdn1.example.com/video/500000/360360.mp4v
http://cdn1.example.com/video/500000/540540.mp4v
http://cdn1.example.com/video/500000/720720.mp4v
...

```

The Media Presentation conforms to the ISO Base media file format Live profile in subclause 8.4.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="dynamic"
  minimumUpdatePeriod="PT2S"

```

```

timeShiftBufferDepth="PT30M"
availabilityStartTime="2014-10-17T17:17:05Z"
minBufferTime="PT4S"
profiles="urn:mpeg:dash:profile:isoff-live:2011"
publishTime="2014-10-17T17:17:05Z">

<BaseURL>http://cdn1.example.com/</BaseURL>
<BaseURL>http://cdn2.example.com/</BaseURL>

<Period id="1">
  <!-- Video -->
  <AdaptationSet
    mimeType="video/mp4"
    codecs="avc1.4D401F"
    frameRate="30000/1001"
    segmentAlignment="true"
    startWithSAP="1">
    <BaseURL>video/</BaseURL>
    <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
media="$Bandwidth%/$Time$.mp4v">
      <SegmentTimeline>
        <S t="0" d="180180" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="v0" width="320" height="240" bandwidth="250000"/>
    <Representation id="v1" width="640" height="480" bandwidth="500000"/>
    <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
  </AdaptationSet>
  <!-- English Audio -->
  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="en" segmentAlignment="0"
startWithSAP="1">
    <SegmentTemplate timescale="48000" initialization="audio/en/init.mp4a"
media="audio/en/$Time$.mp4a">
      <SegmentTimeline>
        <S t="0" d="96000" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="a0" bandwidth="64000" />
  </AdaptationSet>
  <!-- French Audio -->
  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="fr" segmentAlignment="0"
startWithSAP="1">
    <SegmentTemplate timescale="48000" initialization="audio/fr/init.mp4a"
media="audio/fr/$Time$.mp4a">
      <SegmentTimeline>
        <S t="0" d="96000" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="b0" bandwidth="64000" />
  </AdaptationSet>
</Period>
</MPD>

```

G.3 Example for MPEG-2 TS Simple profile

This subclause introduces a simple example for a static presentation, with multiple languages, multiple base URLs, multiple video bitrates, and segments about four seconds in length.

In this MPD, assuming that the first BaseURL element and the Representation with id "1400kbps" is selected, and template results in `http://cdn1.example.com/SomeMovie_1400kbps_${Number%05$.ts}`, the segment list starting at number 0 results in

```

http://cdn1.example.com/SomeMovie_1400kbps_00001.ts
http://cdn1.example.com/SomeMovie_1400kbps_00002.ts
http://cdn1.example.com/SomeMovie_1400kbps_00003.ts
http://cdn1.example.com/SomeMovie_1400kbps_00004.ts
http://cdn1.example.com/SomeMovie_1400kbps_00005.ts
...

```

The Media Presentation conforms to the profile in 8.7.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT6158S"
  availabilityStartTime="2011-05-10T06:16:42"
  minBufferTime="PT1.4S"
  profiles="urn:mpeg:dash:profile:mp2t-simple:2011"
  maxSegmentDuration="PT4S">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

  <Period id="42" duration="PT6158S">
    <AdaptationSet
      mimeType="video/mp2t"
      codecs="avc1.4D401F,mp4a"
      frameRate="24000/1001"
      segmentAlignment="true"
      subsegmentAlignment="true"
      bitstreamSwitching="true"
      startWithSAP="2"
      subsegmentStartsWithSAP="2">
      <ContentComponent contentType="video" id="481"/>
      <ContentComponent contentType="audio" id="482" lang="en"/>
      <ContentComponent contentType="audio" id="483" lang="es"/>
      <BaseURL>SomeMovie/</BaseURL>
      <SegmentTemplate
        media="$RepresentationID$_$Number%05d$.ts"
        index="$RepresentationID$.idx"
        initialization="$RepresentationID$-init.ts"
        bitstreamSwitching="$RepresentationID$-bssw.ts"
        duration="4"/>
      <Representation id="720kbps" bandwidth="792000" width="640" height="368"/>
      <Representation id="1130kbps" bandwidth="1243000" width="704" height="400"/>
      <Representation id="1400kbps" bandwidth="1540000" width="960" height="544"/>
      <Representation id="2100kbps" bandwidth="2310000" width="1120" height="640"/>
      <Representation id="2700kbps" bandwidth="2970000" width="1280" height="720"/>
      <Representation id="3400kbps" bandwidth="3740000" width="1280" height="720"/>
    </AdaptationSet>
  </Period>
</MPD>

```

G.4 Example for multiple stereo views

This subclause introduces a simple example for a stereo video presentation from three cameras in one line where one stereo view is from the left-hand two cameras and the second is from the right-hand two cameras.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT10.00S"
  profiles="urn:mpeg:dash:profile:isoff-main:2011">

  <BaseURL>http://www.example.com/</BaseURL>

  <!-- In this Period there are 3 views: coming from three lined up cameras: C1-C2-C3.
        C1+C2 and C2+C3 each form a stereo pair but C1+C3 does not.
        C2 is taken as the base view for MVC while C1 and C3 are enhancement views -->
  <Period start="PT0.00S" duration="PT2000.00S">
    <SegmentList>
      <Initialization sourceURL="seg-m-init.mp4"/>
    </SegmentList>
    <AdaptationSet mimeType="video/mp4" codecs="avc1.640828">
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="l1 r0"/>
      <Representation id="C2" bandwidth="128000">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C2view-1.mp4"/>
          <SegmentURL media="seg-m1-C2view-2.mp4"/>
          <SegmentURL media="seg-m1-C2view-3.mp4"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
    <!-- The following Adaptation set contains a Representation functionally identical
          to the Representation in the previous Adaptation set. Therefore, these both
          have the same Representation@id. This is done for compatibility to 2D receivers
          that do not understand the schemeIdUri of the Role Descriptor and may ignore the
          Adaptation set -->
    <AdaptationSet mimeType="video/mp4" codecs="avc1.640828">
      <Representation id="C2" bandwidth="128000">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C2view-1.mp4"/>
          <SegmentURL media="seg-m1-C2view-2.mp4"/>
          <SegmentURL media="seg-m1-C2view-3.mp4"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
    <AdaptationSet mimeType="video/mp4" codecs="mvc1.760028">
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="10"/>
      <Representation id="C1" dependencyId="C2" bandwidth="192000">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C1view-1.mp4"/>
          <SegmentURL media="seg-m1-C1view-2.mp4"/>
          <SegmentURL media="seg-m1-C1view-3.mp4"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
    <AdaptationSet mimeType="video/mp4" codecs="mvc1.760028">
      <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="r1"/>
      <Representation id="C3" dependencyId="C2" bandwidth="192000">
        <SegmentList duration="10">
          <SegmentURL media="seg-m1-C3view-1.mp4"/>

```

```

        <SegmentURL media="seg-m1-C3view-2.mp4"/>
        <SegmentURL media="seg-m1-C3view-3.mp4"/>
    </SegmentList>
</Representation>
</AdaptationSet>
</Period>

<!-- In this Period there are only 2 views: C1+C2 form a stereo pair;
      C2 is the base view for MVC and C1 is the enhancement view -->
<Period duration="PT1256.00S">
    <SegmentList>
        <Initialization sourceURL="seg-m-init-2.mp4"/>
    </SegmentList>
    <AdaptationSet mimeType="video/mp4" codecs="avc1.640828">
        <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="r0"/>
        <Representation id="C2" bandwidth="128000">
            <SegmentList duration="10">
                <SegmentURL media="seg-m1-C2view-201.mp4"/>
                <SegmentURL media="seg-m1-C2view-202.mp4"/>
            </SegmentList>
        </Representation>
    </AdaptationSet>
    <AdaptationSet mimeType="video/mp4" codecs="mvc1.760028">
        <Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="10"/>
        <Representation id="C1" dependencyId="C2" bandwidth="192000">
            <SegmentList duration="10">
                <SegmentURL media="seg-m1-C1view-201.mp4"/>
                <SegmentURL media="seg-m1-C1view-202.mp4"/>
            </SegmentList>
        </Representation>
    </AdaptationSet>
</Period>
</MPD>

```

G.5 Example for SVC alternative streams

This simple example introduces a piece of SVC content split into three Representations with each additional bitrate depending on the previous ones.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="static"
    mediaPresentationDuration="PT3256S"
    minBufferTime="PT1.2S"
    profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

    <BaseURL>http://cdn1.example.com/</BaseURL>
    <BaseURL>http://cdn2.example.com/</BaseURL>

    <!-- In this Period the SVC stream is split into three representations -->
    <Period>
        <AdaptationSet
            subsegmentAlignment="true"
            subsegmentStartsWithSAP="2"
            minBandwidth="512000"
            maxBandwidth="1024000"
            width="640"
            height="480"
            frameRate="30"
            lang="en">

```

```

<!-- Independent Representation -->
<Representation
  mimeType="video/mp4"
  codecs="avc1.4D401E,mp4a.40"
  id="tag5"
  bandwidth="512000">
  <BaseURL>video-512k.mp4</BaseURL>
  <SegmentBase indexRange="0-4332"/>
</Representation>
<!-- Representation dependent on above -->
<Representation
  mimeType="video/mp4"
  codecs="avc2.56401E"
  id="tag6"
  dependencyId="tag5"
  bandwidth="768000">
  <BaseURL>video-768k.mp4</BaseURL>
  <SegmentBase indexRange="0-3752"/>
</Representation>
<!-- Representation dependent on both above -->
<Representation
  mimeType="video/mp4"
  codecs="avc2.56401E"
  id="tag7"
  dependencyId="tag5 tag6"
  bandwidth="1024000">
  <BaseURL>video-1024k.mp4</BaseURL>
  <SegmentBase indexRange="0-3752"/>
</Representation>
</AdaptationSet>
</Period>
</MPD>

```

G.6 Example for trick play support

This subclause introduces a simple example for using Sub-Representations to support layered coding.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT1.2S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

  <!-- Period with a multiplexed stream with subrepresentations
  described for use with fast forward -->
  <Period>
    <AdaptationSet
      mimeType="video/mp4"  codecs="avc2.4D401E,avc1.4D401E,mp4a.40"
      width="640" height="480" frameRate="30" lang="en"
      subsegmentAlignment="true" subsegmentStartsWithSAP="2">
        <ContentComponent id="0" contentType="video"/>
        <ContentComponent id="1" contentType="audio"/>
        <Representation id="tag0" bandwidth="512000">
          <BaseURL>video-512k.mp4</BaseURL>
          <SubRepresentation level="0" contentComponent="0" bandwidth="128000"
            codecs="avc1.4D401E" maxPlayoutRate="4"/>

```

```

        <SubRepresentation level="1" dependencyLevel="0" contentComponent="0"
bandwidth="320000" codecs="avc2.4D401E"/>
        <SubRepresentation level="2" contentComponent="1" bandwidth="64000"
codecs="mp4a.40"/>
        <SegmentBase indexRange="7632-7632" />
    </Representation>
</AdaptationSet>
</Period>
</MPD>

```

G.7 Example for content protected by multiple schemes

In the example below, *example.com* is a provider of CDN services and also a hosting service for movie service providers *MoviesSP*. The English audio and the video tracks are encrypted and licensed by *MoviesSP*. However, the French audio track is encrypted and licensed by a different service provider.

A hypothetical DRM standardization organization has registered a Scheme Type 'ZZZZ' with MP4REG and documented how scheme specific licensing information is stored entirely within the content so there is no additional information provided in the **ContentProtection** element. Since the scheme type is registered and the rules for its use are documented, the "urn:mpeg:dash:mp4protection:2011" is used for the @schemeIdUri and "ZZZZ" is the assigned @value.

In addition, a second DRM scheme is used that comes from a DRM vendor who has published documentation of their system that declares that they use the DASH ContentProtection element with a @schemeIdUri attribute value "<http://example.net/052011/drm>". (This DRM vendor owns the domain *example.net* as of May, 2011.) Documentation for this scheme states that there must always be two URLs in the ContentProtection element that are placed in elements defined in the <http://example.net/052011/drm> namespace. The **License** element contains a license token and the **Content** element contains a content token. Regardless of which service provider uses the protection product from this DRM vendor, these rules must always be followed.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:drm="http://example.net/052011/drm"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT10.00S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
  <BaseURL>http://cdn.example.com/movie23453235</BaseURL>

  <Period>
    <!-- Audio protected with a specified license -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="en"
      subsegmentStartsWithSAP="1"
      subsegmentAlignment="true">
      <ContentProtection schemeIdUri="http://example.net/052011/drm">

<drm:License>http://MoviesSP.example.com/protect?license=kljkl sdfiowek</drm:License>

<drm:Content>http://MoviesSP.example.com/protect?content=oyfYvpo8yFyvyo8f</drm:Content>
      </ContentProtection>
      <Representation id="1" bandwidth="64000">
        <BaseURL>audio/en/64.mp4</BaseURL>

```

```

        </Representation>
    </AdaptationSet>
    <!-- Audio protected with embedded information defined by 'ZZZZ' -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="fr"
        subsegmentStartsWithSAP="1"
        subsegmentAlignment="true">
        <ContentProtection schemeIdUri="urn:mpeg:dash:mp4protection:2011" value="ZZZZ"/>
        <Representation id="3" bandwidth="64000">
            <BaseURL>audio/fr/64.mp4</BaseURL>
        </Representation>
    </AdaptationSet>
    <!-- Timed text in the clear -->
    <AdaptationSet mimeType="application/ttml+xml" lang="de">
        <Representation id="5" bandwidth="256">
            <BaseURL>subtitles/de.xml</BaseURL>
        </Representation>
    </AdaptationSet>
    <!-- Video protected with a specified license -->
    <AdaptationSet mimeType="video/mp4" codecs="avc1" subsegmentAlignment="true"
subsegmentStartsWithSAP="2">
        <ContentProtection schemeIdUri="http://example.net/052011/drm">

<drm:License>http://MoviesSP.example.com/protect?license=jfjhwlsdkfiowkl</drm:License>
<drm:Content>http://MoviesSP.example.com/protect?content=mslkfjsfiowelkfl</drm:Content>
        </ContentProtection>
        <BaseURL>video/</BaseURL>
        <Representation id="6" bandwidth="256000" width="320" height="240">
            <BaseURL>video256.mp4</BaseURL>
        </Representation>
        <Representation id="7" bandwidth="512000" width="320" height="240">
            <BaseURL>video512.mp4</BaseURL>
        </Representation>
        <Representation id="8" bandwidth="1024000" width="640" height="480">
            <BaseURL>video1024.mp4</BaseURL>
        </Representation>
    </AdaptationSet>
</Period>
</MPD>

```

G.8 Example for usage of Role descriptor

In the following MPD example, "supplementary" audio Representations with ids "31" or "32" can be presented together with "main" video Representation with id "11" or "12" since **Viewpoint** descriptors are equivalent, i.e. the @schemeIdUri and the @value are equivalent. Similarly, the "supplementary" audio Representation with ids "41" or "42" can be presented together with **"alternate"** video Representations with ids "21" and "22".

NOTE The MPD is not complete and only provides a description of the concept.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Attention: this is not a complete MPD and thus will not validate against the MPD Schema -->
<MPD
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="static"
    mediaPresentationDuration="PT3256S"
    minBufferTime="PT10.00S"
    profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
<Period>
    <AdaptationSet mimeType="video/mp4" group="1">

```



```

        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <Viewpoint schemeIdUri="urn:mpeg:dash:viewpoint:2011" value="vp1"/>
        <Representation id="11" bandwidth="1024000"><!-- ... --></Representation>
        <Representation id="12" bandwidth="512000"><!-- ... --></Representation>
        <!-- ... -->
    </AdaptationSet>

    <AdaptationSet mimeType="video/mp4" group="1">
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <Viewpoint schemeIdUri="urn:mpeg:dash:viewpoint:2011" value="vp2"/>
        <Representation id="11" bandwidth="1024000"><!-- ... --></Representation>
        <Representation id="12" bandwidth="512000"><!-- ... --></Representation>
        <!-- ... -->
    </AdaptationSet>

    <AdaptationSet mimeType="audio/mp4" group="1">
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
        <Viewpoint schemeIdUri="urn:mpeg:dash:viewpoint:2011" value="vp1"/>
        <Representation id="11" bandwidth="1024000"><!-- ... --></Representation>
        <Representation id="12" bandwidth="512000"><!-- ... --></Representation>
        <!-- ... -->
    </AdaptationSet>

    <AdaptationSet mimeType="audio/mp4" group="1">
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
        <Viewpoint schemeIdUri="urn:mpeg:dash:viewpoint:2011" value="vp2"/>
        <Representation id="11" bandwidth="1024000"><!-- ... --></Representation>
        <Representation id="12" bandwidth="512000"><!-- ... --></Representation>
        <!-- ... -->
    </AdaptationSet>

</Period>
<!-- ... -->
</MPD>

```

G.9 Example for usage of Event Messaging

In the following MPD example, two types of Events are added. In the Event stream with @schemeIdUri="urn:org:example:xscte3", time-synchronous events are added to the program. The events have a presentation time and a presentation duration. In addition, one Representation carries the MPD validity expiry information in an Inband event stream and an ad break information for ad insertion (assuming ANSI/SCTE 35 fields are used to represent the ad break parameters).

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="dynamic"
  minimumUpdatePeriod="PT2S"
  timeShiftBufferDepth="PT30M"
  availabilityStartTime="2011-12-25T12:30:00"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
  publishTime="2011-12-25T12:30:00">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

```

```

<Period id="1">
  <EventStream schemeIdUri="urn:uuid:XYZY" timescale="1000" value="call">
    <Event presentationTime="0" duration="10000" id="0" messageData="+ 1 800 10101010"/>
    <Event presentationTime="20000" duration="10000" id="1" messageData="+ 1 800
10101011"/>
    <Event presentationTime="40000" duration="10000" id="2" messageData="+ 1 800
10101012"/>
    <Event presentationTime="60000" duration="10000" id="3" messageData="+ 1 800
10101013"/>
  </EventStream>
  <!-- Video -->
  <AdaptationSet
    mimeType="video/mp4"
    codecs="avc1.4D401F"
    frameRate="30000/1001"
    segmentAlignment="true"
    startWithSAP="1">
    <BaseURL>video/</BaseURL>
    <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
media="$Bandwidth%/$Time$.mp4v">
      <SegmentTimeline>
        <S t="0" d="180180" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="v0" width="320" height="240" bandwidth="250000"/>
    <Representation id="v1" width="640" height="480" bandwidth="500000"/>
    <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
  </AdaptationSet>
  <!-- English Audio -->
  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="en" segmentAlignment="0"
startWithSAP="1">
    <SegmentTemplate timescale="48000" initialization="audio/en/init.mp4a"
media="audio/en/$Time$.mp4a">
      <SegmentTimeline>
        <S t="0" d="96000" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="a0" bandwidth="64000">
      <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012"
value="1"></InbandEventStream>
      <InbandEventStream schemeIdUri="urn:org:example:event"
value="avail"></InbandEventStream>
    </Representation>
  </AdaptationSet>
  <!-- French Audio -->
  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40" lang="fr" segmentAlignment="0"
startWithSAP="1">
    <SegmentTemplate timescale="48000" initialization="audio/fr/init.mp4a"
media="audio/fr/$Time$.mp4a">
      <SegmentTimeline>
        <S t="0" d="96000" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="b0" bandwidth="64000" />
  </AdaptationSet>
</Period>
</MPD>

```

The 'emsg' box may contain information as follows:

scheme_id_uri	= "urn:org:example:xscte35"
value	= 0x0602
timescale	= 1000
presentation_time_delta	= 8000

```

event_duration          = 0xFFFFFFFF
id                      = 12356789
message_data[] =
    <metadata label="Ad Insertion Trigger"
      type="http://www.example.com/schemas/xscte35"
      namespace="xscte35"
      xmlns:xscte35="http://www.example.com/schemas/xscte35">
      <xscte35:signal>
        <id name="segmentEventId" value="1" />
        <time name="start" value="10" />
        <time name="end" value="20" />
      </xscte35:signal>
    </metadata>

```

G.10 Example for MPD Adaptation Set Linking

This example shows how to describe MPD Adaptation Set Linking scheme as defined in subclause 5.8.5.9. In this example, main video is the server-based mosaic channel as described in ISO/IEC TR 23009-3.

The screen position in main video and MPD linking for each mosaic video are described by SRD scheme and MPD Adaptation Set Linking scheme in the separate Adaptation Set(s).

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" type="dynamic"
  availabilityStartTime="2015-05-30T09:30:10Z" minimumUpdatePeriod="PT10S" minBufferTime="PT1S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011" publishTime="2015-05-30T09:30:10Z">
  <ProgramInformation>
    <Title>Example of a DASH Media Presentation Description using Spatial Relationships
    Description to indicate tiles of a video</Title>
  </ProgramInformation>
  <Period id="1">
    <!-- Mosaic Video -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
    value="0,0,0,2,2,2,2"/>
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:sai:2014" value="1"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <Representation id="1" mimeType="video/mp4" codecs="avc1.42c01e"
    width="640" height="360" bandwidth="226597" startWithSAP="1">
        <BaseURL>full_video_small.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="837-988"/>
      </Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.42c01f"
    width="1280" height="720" bandwidth="553833" startWithSAP="1">
        <BaseURL>full_video_hd.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="838-989"/>
      </Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc1.42c033"
    width="3840" height="2160" bandwidth="1055223" startWithSAP="1">
        <BaseURL>full_video_4k.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="839-990"/>
      </Representation>
    </AdaptationSet>
    <!-- Tile 1/Service1 -->
    <EmptyAdaptationSet>
      <EssentialProperty schemeIdUri="urn:mpeg:dash:mpd-as-linking:2015"
    value="http://example.com/service1/my.mpd#period=1&as=video"/>
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
    value="0,0,0,1,1,2,2"/>
    </EmptyAdaptationSet>
  </Period>
</MPD>

```

```

        <!--Tile /Service 2 -->
        <EmptyAdaptationSet>
            <EssentialProperty schemeIdUri="urn:mpeg:dash:mpd-as-linking:2015"
value="http://example.com/service2/my.mpd#period=1&as=video timeOffset=70000"/>
            <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
value="0,1,0,1,1"/>
        </EmptyAdaptationSet>
        <!--Tile 3/Service 3 -->
        <EmptyAdaptationSet>
            <EssentialProperty schemeIdUri="urn:mpeg:dash:mpd-as-linking:2015"
value="http://example.com/service3/my.mpd#period=1&as=video timeOffset=100000"/>
            <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
value="0,0,1,1,1"/>
        </EmptyAdaptationSet>
        <!--Tile 4/Service 4 -->
        <EmptyAdaptationSet>
            <EssentialProperty schemeIdUri="urn:mpeg:dash:mpd-as-linking:2015"
value="http://example.com/service4/my.mpd#period=1&as=video timeOffset=120000"/>
            <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
value="0,1,1,1,1"/>
        </EmptyAdaptationSet>
    </Period>
</MPD>

```

G.11 Remote Element Entity

```

<?xml version="1.0" encoding="utf-8"?>
<!--Single content continuing at the period boundary.-->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" minBufferTime="PT1.500000S"
type="static" mediaPresentationDuration="PT704S" profiles="urn:mpeg:dash:profile:isoff-
live:2011">
    <Period id="0" duration="PT250S">
        <AssetIdentifier schemeIdUri="urn:org:dashif:asset-id:2013"
value="md:cid:EIDR:10.5240%2f0EFB-02CD-126E-8092-1E49-W"/>
        <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720"
maxFrameRate="24" par="16:9">
            <Representation id="1" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="980104">
                <SegmentTemplate timescale="12288" presentationTimeOffset="1024"
duration="24576" media="BBB_720_1M_video_$Number$.mp4" startNumber="1"
initialization="BBB_720_1M_video_init.mp4"/>
            </Representation>
            <Representation id="2" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1950145">
                <SegmentTemplate timescale="12288" presentationTimeOffset="1024"
duration="24576" media="BBB_720_2M_video_$Number$.mp4" startNumber="1"
initialization="BBB_720_2M_video_init.mp4"/>
            </Representation>
            <Representation id="3" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3893089">
                <SegmentTemplate timescale="12288" presentationTimeOffset="1024"
duration="24576" media="BBB_720_4M_video_$Number$.mp4" startNumber="1"
initialization="BBB_720_4M_video_init.mp4"/>
            </Representation>
        </AdaptationSet>
        <AdaptationSet segmentAlignment="true">
            <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.29"
audioSamplingRate="48000" startWithSAP="1" bandwidth="33434">
                <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
            </Representation>
        </AdaptationSet>
    </Period>
</MPD>

```

```

        <SegmentTemplate timescale="48000" duration="94175"
media="BBB_32k_${Number$.mp4" startNumber="1" initialization="BBB_32k_init.mp4"/>
        </Representation>
    </AdaptationSet>
</Period>
<Period xlink:href="example_G11_remote.period.xml" xlink:actuate="onRequest"
xmlns:xlink="http://www.w3.org/1999/xlink"/>
    <Period id="2" duration="PT344S">
        <AssetIdentifier schemeIdUri="urn:org:dashif:asset-id:2013"
value="md:cid:EIDR:10.5240%2f0EFB-02CD-126E-8092-1E49-W"/>
        <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720"
maxFrameRate="24" par="16:9">
            <Representation id="1" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="980104">
                <SegmentTemplate timescale="12288" presentationTimeOffset="3073024"
duration="24576" media="BBB_720_1M_video_${Number$.mp4" startNumber="126"
initialization="BBB_720_1M_video_init.mp4"/>
            </Representation>
            <Representation id="2" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1950145">
                <SegmentTemplate timescale="12288" presentationTimeOffset="3073024"
duration="24576" media="BBB_720_2M_video_${Number$.mp4" startNumber="126"
initialization="BBB_720_2M_video_init.mp4"/>
            </Representation>
            <Representation id="3" mimeType="video/mp4" codecs="avc1.4d401f"
width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3893089">
                <SegmentTemplate timescale="12288" presentationTimeOffset="3073024"
duration="24576" media="BBB_720_4M_video_${Number$.mp4" startNumber="126"
initialization="BBB_720_4M_video_init.mp4"/>
            </Representation>
        </AdaptationSet>
        <AdaptationSet segmentAlignment="true">
            <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.29"
audioSamplingRate="48000" startWithSAP="1" bandwidth="33434">
                <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
            <SegmentTemplate timescale="48000" presentationTimeOffset="11964416"
duration="94175" media="BBB_32k_${Number$.mp4" startNumber="126"
initialization="BBB_32k_init.mp4"/>
            </Representation>
        </AdaptationSet>
    </Period>
</MPD>

```

Content of remote period element:

```

<?xml version="1.0" encoding="UTF-8"?>
<Period xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
id="1" duration="PT110S" start="PT250S">
    <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="24"
par="16:9">
        <Representation id="1" mimeType="video/mp4" codecs="avc1.4d401f" width="1280" height="720"
frameRate="24" sar="1:1" startWithSAP="1" bandwidth="927434">
            <SegmentTemplate timescale="12288" presentationTimeOffset="1024" duration="61440"
media="ED_720_1M_MPEG2_video_${Number$.mp4" startNumber="1"
initialization="ED_720_1M_MPEG2_video_init.mp4" />
        </Representation>
        <Representation id="2" mimeType="video/mp4" codecs="avc1.4d401f" width="1280" height="720"
frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1865663">

```

```

    <SegmentTemplate timescale="12288" presentationTimeOffset="1024" duration="61440"
media="ED_720_2M_MPEG2_video_${Number$.mp4" startNumber="1"
initialization="ED_720_2M_MPEG2_video_init.mp4" />
  </Representation>
  <Representation id="3" mimeType="video/mp4" codecs="avc1.4d401f" width="1280" height="720"
frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3750115">
    <SegmentTemplate timescale="12288" presentationTimeOffset="1024" duration="61440"
media="ED_720_4M_MPEG2_video_${Number$.mp4" startNumber="1"
initialization="ED_720_4M_MPEG2_video_init.mp4" />
  </Representation>
</AdaptationSet>
<AdaptationSet segmentAlignment="true">
  <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.29" audioSamplingRate="48000"
startWithSAP="1" bandwidth="33026">
    <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
    <SegmentTemplate timescale="48000" duration="239615" media="ED_MPEG2_32k_${Number$.mp4"
startNumber="1" initialization="ED_MPEG2_32k_init.mp4" />
  </Representation>
</AdaptationSet>
</Period>

```

NOTE The example code is accessible here: https://standards.iso.org/iso-iec/23009/-1/ed-5/en/example_G11_remote.period.xml

G.12 Directory Limit Support in Segment Template Based Delivery

In certain cases, if Segments are stored in directories and Segments, then the directory limit may be reached. For example, consider a 9 hour asset with 2 second Segment duration with server directory limit of 1 000 files. This VOD asset will result in 16 200 segments. With a server directory limit of 1 000 files per directory on the server, this would require 17 directories to store the segments.

Segment Templates can only be changed at Period boundaries. However, by using Period continuity, this issue can be solved in order to ensure continuous playout across Periods. Assuming a directory limit of 1 000 files, the following may be done:

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
type="dynamic" minimumUpdatePeriod="PT10S"
timeShiftBufferDepth="PT600S"
minBufferTime="PT2S"
profiles="urn:mpeg:dash:profile:isoff-main:2011"
publishTime="2014-10-17T17:17:05Z"
availabilityStartTime="2014-10-17T17:17:05Z">
  <Period id="1" start="PT0S">
    <BaseURL> http://example.com/1/</BaseURL>
    <SegmentTemplate media=".$RepresentationID$/${Number$.m4s" initialization=".$RepresentationID$-
init.mp4"/>
    <!-- Video -->
    <AdaptationSet id="1" mimeType="video/mp4" codecs="hev1.A1.80.L93.B0" segmentAlignment="true"
startWithSAP="1">
      <SegmentTemplate timescale="25" duration="25"/>
      <Representation id="v2048" bandwidth="2048000"/>
      <Representation id="v1024" bandwidth="1024000"/>
      <Representation id="v512" bandwidth="512000"/>
      <Representation id="v128" bandwidth="128000"/>
    </AdaptationSet>
    <!-- Audio -->
  </Period>

```

```

    <AdaptationSet id="2" mimeType="audio/mp4" codecs="mp4a.40.2" segmentAlignment="true"
    startWithSAP="1" bitstreamSwitching="true">
      <SegmentTemplate timescale="20" duration="20"/>
      <Representation id="a128" bandwidth="128000"/>
      <Representation id="a64" bandwidth="64000"/>
    </AdaptationSet>
  </Period>
  <Period id="2" start="PT1000S">
    <BaseURL> http://example.com/2/</BaseURL>
    <SegmentTemplate media=".$RepresentationID$/$Number$.m4s" initialization="$RepresentationID$-
    init.mp4"/>
    <!-- Video -->
    <AdaptationSet id="1" mimeType="video/mp4" codecs="hev1.A1.80.L93.B0" segmentAlignment="true"
    startWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:period-continuity:2015" value="1"/>
      <SegmentTemplate timescale="25" duration="25" presentationTimeOffset="25000"/>
      <Representation id="v2048" bandwidth="2048000"/>
      <Representation id="v1024" bandwidth="1024000"/>
      <Representation id="v512" bandwidth="512000"/>
      <Representation id="v128" bandwidth="128000"/>
    </AdaptationSet>
    <!-- Audio -->
    <AdaptationSet id="2" mimeType="audio/mp4" codecs="mp4a.40.2" segmentAlignment="true"
    startWithSAP="1" bitstreamSwitching="true">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:period-continuity:2015" value="1"/>
      <SegmentTemplate timescale="20" duration="20" presentationTimeOffset="20000"/>
      <Representation id="a128" bandwidth="128000"/>
      <Representation id="a64" bandwidth="64000"/>
    </AdaptationSet>
  </Period>
</MPD>

```

G.13 Data URLs

In order to minimize the http requests at startup or at Period boundaries, data URLs allow inclusion of small data items as "immediate" data, as if it had been included externally. A server-side example processing for adding for example Initialization Segments is as follows:

- traverse the MPD tree, propagate templated initialization segment URLs (if any) down the MPD levels (Period, Adaptation Set) to the Representation level (only for the first Period).
- resolve the template (if any) potentially using **BaseURL** elements (if any), download the resolved IS and add a **SegmentTemplate** element with a "data:" URL scheme whose content is the base64 encoded IS.
- For example, assume the following MPD part:

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
mediaPresentationDuration="PT3256S"
minBufferTime="PT1.500000S" profiles="urn:mpeg:dash:profile:isoff-ext-live:2014">
  <Period>
    <AdaptationSet startWithSAP="2" segmentAlignment="true" id="1" sar="1:1"
    mimeType="video/mp4">
      <InbandEventStream
      schemeIdUri="tag:rdmedia.bbc.co.uk,2014:events:ballposition" value="1"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <BaseURL>avc3-events/</BaseURL>
    </AdaptationSet>
  </Period>
</MPD>

```



```

        <SegmentTemplate startNumber="1" timescale="1000" duration="3840"
media="$RepresentationID$/$Number%06d$.m4s" initialization="$RepresentationID$/IS.mp4"/>
        <Representation id="960x540p50" codecs="avc3.64001f" height="540"
width="960" frameRate="50" scanType="progressive" bandwidth="2814440"/>
        <Representation id="192x108p6_25" codecs="avc3.42c015" height="108"
width="192" frameRate="25/4" scanType="progressive" bandwidth="31368"/>
    </AdaptationSet>
</Period>
</MPD>

```

This may be modified to move the templated @initialization attribute from the **AdaptationSet.SegmentTemplate** element to each **Representation.SegmentTemplate** element and the information from equivalent ISs is included. The resulting MPD looks as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
mediaPresentationDuration="PT3256S"
minBufferTime="PT1.500000S" profiles="urn:mpeg:dash:profile:isoff-ext-live:2014">
    <Period>
        <AdaptationSet id="1" segmentAlignment="true" sar="1:1" mimeType="video/mp4"
startWithSAP="2" maxPlayoutRate="1">
            <InbandEventStream
schemeIdUri="tag:rdmedia.bbc.co.uk,2014:events:ballposition" value="1"/>
            <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
            <BaseURL>avc3-events/</BaseURL>
            <SegmentTemplate media="$RepresentationID$/$Number%06d$.m4s"
timescale="1000" duration="3840" startNumber="1"/>
            <Representation id="960x540p50" bandwidth="2814440" width="960"
height="540" frameRate="50" codecs="avc3.64001f" maxPlayoutRate="1" scanType="progressive">
                <SegmentTemplate
initialization="data:video/mp4;base64,AAAAHGZ0eXBpc282AAAAWF2YzFpc29tZGFzaAAAAldtb292...AAAAA
AAAAAQc3RjbwAAAAA" />
            </Representation>
            <Representation id="192x108p6_25" bandwidth="31368" width="192"
height="108" frameRate="25" codecs="avc3.42c015" maxPlayoutRate="1" scanType="progressive">
                <SegmentTemplate
initialization="data:video/mp4;base64,AAAAHGZ0eXBpc282AAAAWF2YzFpc29tZGFzaAAAAlNtb292...AAAAA
BBzdHNjAAAAA" />
            </Representation>
        </AdaptationSet>
    </Period>
</MPD>

```

G.14 Example Live MPD with Timing Information

The following MPD shows the use of the **UTCTiming** element and the **LeapSecondInformation** element in a Live service including MPD update events.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD type="dynamic" xmlns="urn:mpeg:dash:schema:mpd:2011"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
profiles="urn:mpeg:dash:profile:isoff-ext-live:2014" minBufferTime="PT1.143S"
maxSegmentDuration="PT3.84S" minimumUpdatePeriod="PT1H" timeShiftBufferDepth="PT2M"
availabilityStartTime="2019-03-24T21:20:00Z" publishTime="2019-03-24T20:44:33.540Z">
    <Period id="first" start="PT0S">
        <AdaptationSet startWithSAP="2" segmentAlignment="true" par="16:9" id="1"
contentType="video" mimeType="video/mp4">
            <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012" value="1"/>

```



```

        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <SegmentTemplate startNumber="404547501"
presentationTimeOffset="310692480000" timescale="200" duration="768"
media="$RepresentationID$/$Number%06d$.m4s" initialization="$RepresentationID$/IS.mp4" />
        <Representation id="1280x720p50" codecs="avc3.640020" height="720"
width="1280" frameRate="50" bandwidth="5447392" scanType="progressive" />
        </AdaptationSet>
        <AdaptationSet startWithSAP="2" segmentAlignment="true" id="6" codecs="mp4a.40.2"
audioSamplingRate="48000" contentType="audio" lang="eng" mimeType="audio/mp4" >
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="6"/>
        <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012" value="1"/>
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <SegmentTemplate startNumber="404547501"
presentationTimeOffset="74566195200000" timescale="48000" duration="184320"
media="$RepresentationID$/$Number%06d$.m4s" initialization="$RepresentationID$/IS.mp4" />
        <Representation id="320kbps-5_1" bandwidth="319520" />
        </AdaptationSet>
    </Period>
    <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014"
value="https://example.com/iso"/>
    <LeapSecondInformation availabilityStartLeapOffset="0" nextAvailabilityStartLeapOffset="1"
nextLeapChangeTime="2020-01-01T00:00:00Z"/>
</MPD>

```

G.15 Example MPD for Preselection Descriptor

This subclause introduces a simple example of using the Preselection Descriptor to define two audio experiences. The first audio experience is the combination of English dialogue with the Music and Effects. The second audio experience is the combination of Spanish dialogue with the Music and Effects.

```

<?xml version="1.0" encoding="utf-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="dynamic"
  availabilityStartTime="2018-12-20T06:04:22Z"
  publishTime="2018-12-20T06:04:22Z"
  minimumUpdatePeriod="PT2S"
  mediaPresentationDuration="PT0H4M9.708S"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">

  <Period id="1" start="PT0S">
    <!-- Video -->
    <AdaptationSet id="1" mimeType="video/mp4" segmentAlignment="true" startWithSAP="1">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <SegmentTemplate timescale="5994" media="video_$Number$.mp4" initialization="video.mp4">
        <SegmentTimeline>
          <S t="0" d="12000" r="154"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="1" bandwidth="3732256" codecs="hev1.1.6.L120.90" width="1920"
height="1080"/>
    </AdaptationSet>
    <!-- Audio (M&E) -->
    <AdaptationSet id="2" mimeType="audio/mp4" codecs="mhm2.0x0C" segmentAlignment="true"
startWithSAP="1">

```

```

<EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" value="1,2 3"/>
<EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016" value="2,2 4"/>
<Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
<SegmentTemplate timescale="48000" media="audio0_$_Number$.mp4" initialization="audio0.mp4">
  <SegmentTimeline>
    <S t="0" d="90112" r="133"/>
  </SegmentTimeline>
</SegmentTemplate>
<Representation id="2" bandwidth="132669" audioSamplingRate="48000"/>
</AdaptationSet>
<!-- Audio (English) -->
<AdaptationSet id="3" mimeType="audio/mp4" lang="en" codecs="mhm2.0x0C"
segmentAlignment="true" startWithSAP="1">
  <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
  <SegmentTemplate timescale="48000" media="audio1_$_Number$.mp4" initialization="audio1.mp4">
    <SegmentTimeline>
      <S t="0" d="90112" r="133"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="3" bandwidth="32494" audioSamplingRate="48000"/>
</AdaptationSet>
<!-- Audio (Spanish) -->
<AdaptationSet id="4" mimeType="audio/mp4" lang="es" codecs="mhm2.0x0C"
segmentAlignment="true" startWithSAP="1">
  <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
  <SegmentTemplate timescale="48000" media="audio2_$_Number$.mp4" initialization="audio2.mp4">
    <SegmentTimeline>
      <S t="0" d="90112" r="133"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="4" bandwidth="32494" audioSamplingRate="48000"/>
</AdaptationSet>
</Period>
</MPD>

```

G.16 Example MPD for Preselection Element

This subclause introduces a simple example of using the Preselection Element to define two audio experiences. The first audio experience is the combination of English dialogue with the Music and Effects. The second audio experience is the combination of Spanish dialogue with the Music and Effects.

```

<?xml version="1.0" encoding="utf-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="dynamic"
  availabilityStartTime="2018-12-20T06:04:22Z"
  publishTime="2018-12-20T06:04:22Z"
  minimumUpdatePeriod="PT2S"
  mediaPresentationDuration="PT0H4M9.708S"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <Period id="1" start="PT0S">
    <!-- Video -->
    <AdaptationSet id="1" mimeType="video/mp4" segmentAlignment="true" startWithSAP="1">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <SegmentTemplate timescale="5994" media="video_$_Number$.mp4" initialization="video.mp4">
        <SegmentTimeline>
          <S t="0" d="12000" r="154"/>
        </SegmentTimeline>
      </SegmentTemplate>
    </AdaptationSet>
  </Period>
</MPD>

```

```

        </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="1" bandwidth="3732256" codecs="hev1.1.6.L120.90" width="1920"
height="1080"/>
</AdaptationSet>
<!-- Audio (M&E) -->
<AdaptationSet id="2" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
    <SegmentTemplate timescale="48000" media="audio0_${Number$.mp4" initialization="audio0.mp4">
        <SegmentTimeline>
            <S t="0" d="90112" r="133"/>
        </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="2" bandwidth="132669" codecs="mhm2.0xC0" audioSamplingRate="48000"/>
</AdaptationSet>
<!-- Audio (English) -->
<AdaptationSet id="3" mimeType="audio/mp4" lang="en" segmentAlignment="true"
startWithSAP="1">
    <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
    <SegmentTemplate timescale="48000" media="audio1_${Number$.mp4" initialization="audio1.mp4">
        <SegmentTimeline>
            <S t="0" d="90112" r="133"/>
        </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="3" bandwidth="32494" codecs="mhm2.0xC0" audioSamplingRate="48000"/>
</AdaptationSet>
<!-- Audio (Spanish) -->
<AdaptationSet id="4" mimeType="audio/mp4" lang="es" segmentAlignment="true"
startWithSAP="1">
    <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
    <SegmentTemplate timescale="48000" media="audio2_${Number$.mp4" initialization="audio2.mp4">
        <SegmentTimeline>
            <S t="0" d="90112" r="133"/>
        </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="4" bandwidth="32825" codecs="mhm2.0xC0" audioSamplingRate="48000"/>
</AdaptationSet>
<!-- Preselections -->
<Preselection id="1" tag="1" lang="en" preselectionComponents="2 3">
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:mpegB:cicp:ChannelConfiguration"
value="2"/>
    <Label id="1">Main English</Label>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
</Preselection>
<Preselection id="2" tag="2" lang="es" preselectionComponents="2 4">
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:mpegB:cicp:ChannelConfiguration"
value="2"/>
    <Label id="1">Main Spanish</Label>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
</Preselection>
</Period>
</MPD>

```

G.17 Example MPD for Preselection Element using ContentComponent

This subclause introduces a simple example of using the Preselection Element to define two audio experiences that are multiplexed at the file-container level. Three audio components are delivered in a single Representation and described by **ContentComponent** elements. The first audio experience is the combination of English dialogue with the Music and Effects. The second audio experience is the combination of Spanish dialogue with the Music and Effects.

```

<?xml version="1.0" encoding="utf-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
type="dynamic"
availabilityStartTime="2018-12-20T06:04:22Z"
publishTime="2018-12-20T06:04:22Z"
minimumUpdatePeriod="PT2S"
mediaPresentationDuration="PT0H4M9.708S"
minBufferTime="PT4S"
profiles="urn:mpeg:dash:profile:isoff-live:2011">
<Period id="1" start="PT0S">
  <!-- Video -->
  <AdaptationSet id="1" mimeType="video/mp4" codecs="hev1.1.6.L120.90" segmentAlignment="true"
startWithSAP="1">
    <SegmentTemplate timescale="5994" media="video_$Number$.mp4" initialization="video.mp4">
      <SegmentTimeline>
        <S t="0" d="12000" r="154"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="1" bandwidth="3732256" width="1920" height="1080"/>
  </AdaptationSet>
  <!-- Audio -->
  <AdaptationSet id="2" mimeType="audio/mp4" codecs="mp4a.40.2,mp4a.40.2,mp4a.40.2"
segmentAlignment="true" startWithSAP="1">
    <EssentialProperty schemeIdUri="urn:mpeg:dash:preselection:2016"/>
    <ContentComponent id="3">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
    </ContentComponent>
    <ContentComponent id="4" lang="en">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
    </ContentComponent>
    <ContentComponent id="5" lang="es">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
    </ContentComponent>
    <SegmentTemplate timescale="48000" media="audio0_$Number$.mp4" initialization="audio0.mp4">
      <SegmentTimeline>
        <S t="0" d="90112" r="133"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="2" bandwidth="132669" audioSamplingRate="48000"/>
  </AdaptationSet>
  <!-- Preselections -->
  <Preselection id="1" tag="1" lang="en" preselectionComponents="3 4">
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:mpegB:cicp:ChannelConfiguration"
value="2"/>
    <Label id="1">Main English</Label>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
  </Preselection>
  <Preselection id="2" tag="2" lang="es" preselectionComponents="3 5">
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:mpegB:cicp:ChannelConfiguration"
value="2"/>
    <Label id="1">Main Spanish</Label>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="dub"/>
  </Preselection>
</Period>
</MPD>

```

G.18 Example MPD for Low Latency Streaming

This subclause introduces a simple example of using the **ServiceDescription** element to control latency for a subset of clients as well as the usage of the Leap Second Information and Producer Reference Time.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD type="dynamic" xmlns="urn:mpeg:dash:schema:mpd:2011"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
profiles="urn:mpeg:dash:profile:isoff-ext-live:2014" minBufferTime="PT1.143S"
maxSegmentDuration="PT3.84S" minimumUpdatePeriod="PT1H" timeShiftBufferDepth="PT2M"
availabilityStartTime="2019-08-06T13:31:00Z" publishTime="2019-08-06T13:31:00Z">
  <ServiceDescription id="0">
    <Scope schemeIdUri="tag:example.com,082019:lowlatencyclients" value="2"/>
    <Latency min="4800" max="34800" target="6800" referenceId="7"/>
    <PlaybackRate min="0.96" max="1.04"/>
  </ServiceDescription>
  <Period id="first" start="PT0S">
    <AdaptationSet startWithSAP="2" segmentAlignment="true" par="16:9" id="1"
contentType="video" mimeType="video/mp4" >
      <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012" value="1" />
      <ProducerReferenceTime id="7" wallClockTime="2019-08-06T13:44:12Z"
presentationTime="158400">
        <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014"
value="https://example.com/iso"/>
      </ProducerReferenceTime>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <SegmentTemplate startNumber="404547501" presentationTimeOffset="310692480000"
timescale="200" duration="768" media="$RepresentationID$/$Number%06d$.m4s"
initialization="$RepresentationID$/IS.mp4" availabilityTimeOffset="2.88"
availabilityTimeComplete="false"/>
        <Representation id="1280x720p50" codecs="avc3.640020" height="720"
width="1280" frameRate="50" bandwidth="5447392" scanType="progressive" />
      </AdaptationSet>
      <AdaptationSet startWithSAP="2" segmentAlignment="true" id="6" codecs="mp4a.40.2"
audioSamplingRate="48000" contentType="audio" lang="eng" mimeType="audio/mp4" >
        <AudioChannelConfiguration
schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="6"/>
        <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012" value="1"/>
        <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
        <SegmentTemplate startNumber="404547501"
presentationTimeOffset="74566195200000" timescale="48000" duration="184320"
media="$RepresentationID$/$Number%06d$.m4s" initialization="$RepresentationID$/IS.mp4"
availabilityTimeOffset="2.88" availabilityTimeComplete="false" />
        <Representation id="320kbps-5_1" bandwidth="319520" />
      </AdaptationSet>
    </Period>
    <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014"
value="https://example.com/iso"/>
    <LeapSecondInformation availabilityStartLeapOffset="37"/>
  </MPD>

```

G.19 DASH Profile for CMAF Content

This subclause introduces a simple example for an MPD for DASH profile for CMAF content.

```

<?xml version="1.0"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT24S"
  availabilityStartTime="2014-10-17T17:17:05Z"
  minBufferTime="PT4S"

```

```

profiles="urn:mpeg:dash:profile:cmaf:2019,urn:mpeg:dash:profile:isoff-live:2011">

<Period id="1">
  <!-- Video -->
  <AdaptationSet
    contentType="video"
    id="1"
    mimeType="video/mp4 profiles='cmfc'"
    codecs="avc1.4D401F"
    maxWidth="1080"
    maxHeight="720"
    maxFrameRate="30"
    segmentProfiles="cmfs cmff"
    segmentAlignment="true"
    startWithSAP="1">
    <SegmentTemplate timescale="30" initialization="$RepresentationID$/0"
media="$RepresentationID$/Number$">
      <SegmentTimeline>
        <S t="0" d="120" r="5"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="video1/1" bandwidth="250000"/>
    <Representation id="video1/2" bandwidth="500000"/>
    <Representation id="video1/3" bandwidth="1000000"/>
  </AdaptationSet>
  <!-- Audio -->
  <AdaptationSet
    contentType="audio"
    id="1"
    mimeType="audio/mp4 profiles='cmfc'"
    codecs="mp4a.40.5"
    segmentProfiles="cmfs cmff"
    segmentAlignment="true"
    startWithSAP="1">
    <SegmentTemplate timescale="48" initialization="$RepresentationID$/0"
media="$RepresentationID$/Number$">
      <SegmentTimeline>
        <S t="0" d="120" r="5"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="audio1/1" bandwidth="2500"/>
    <Representation id="audio1/2" bandwidth="500000"/>
  </AdaptationSet>
</Period>
</MPD>

```

G.20 Resynchronization

An example is provided for the usage of the Resynchronization feature in the context of low-latency service offerings. The **Resync** element is added in two places for two purposes. First of all, it describes for all video Adaptation Set, that a chunked mode is applied.

```
<Resync type="0" dT="500000" dImin="0.03125" dImax="0.09375"/>
```

The resynchronization point is not further specified, i.e. @type=0. The maximum duration of the chunk is identified by the @dT value of "500000" and by the usage of the @timescale value of "1000000", this results in 500ms. The minimum distance in bytes between two Resynchronization Points is provided by @dImin="0.03125" times the bandwidth value of the first Representation

(@bandwidth="500000"), which is 15,625 bytes. The same applies for the maximum difference by multiplying the value of @bandwidth with the value of @dImax, resulting in 46,875 bytes. For the second Representation with @bandwidth="200000", this results in the minimum difference of 6,250 bytes and the maximum difference of 18,750 bytes. For the third Representation, with @bandwidth="300000", this is respectively set to 9,375 bytes and 28,125 bytes.

In addition, for the third Representation, an additional **Resync** element is present as:

```
<Resync type="2" dT="1000000" dImin="0.1" dImax="0.15" marker="true"/>
```

This information signals that at least every second (based on the value of @dT), one Resync Point of type 2 can be found in this Representation, which may for example be used for fast access to the content, or may also be used for fast down-switching or Resynchronization. The minimum and maximum distance are documented according to @dImin and @dImax as 30,000 and 45,000 bytes, respectively. As the @marker flag is set to true, a DASH client may apply the algorithm documented in A.12.3 to find the Resynchronization Point.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-
  MPD.xsd" profiles="urn:dvb:dash:profile:dvb-dash:2014" type="dynamic"
  minimumUpdatePeriod="PT500S" availabilityStartTime="2020-02-19T10:42:02.684Z" publishTime="2020-
  02-19T11:01:42.688Z" minBufferTime="PT2.0S">
  <ProgramInformation>
  </ProgramInformation>
  <ServiceDescription id="0">
    <Latency target="3500" referenceId="0"/>
  </ServiceDescription>
  <Period id="0" start="PT0.0S">
    <AdaptationSet id="0" contentType="video" segmentAlignment="true" frameRate="10/1"
    maxWidth="960" maxHeight="400" par="12:5">
      <ProducerReferenceTime id="0" inband="true" type="encoder"
      wallClockTime="2020-02-19T10:42:02.667Z" presentationTime="0">
        <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014"
        value="http://time.akamai.com"/>
      </ProducerReferenceTime>
      <Resync type="0" dT="500000" dImin="0.03125" dImax="0.09375"/>
      <SegmentTemplate timescale="1000000" duration="8000000"
      availabilityTimeOffset="7.500" availabilityTimeComplete="false" initialization="init-
      stream$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-$Number%05d$.m4s"
      startNumber="1"/>
      <Representation id="0" mimeType="video/mp4" codecs="avc1.640016"
      bandwidth="500000" width="960" height="400" sar="1:1" qualityRanking="5"/>
      <Representation id="1" mimeType="video/mp4" codecs="avc1.640016"
      bandwidth="200000" width="720" height="300" sar="1:1" qualityRanking="2"/>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.640016"
      bandwidth="300000" width="720" height="300" sar="1:1" qualityRanking="1">
        <Resync type="2" dT="1000000" dImin="0.1" dImax="0.15"
        marker="true"/>
      </Representation>
    </AdaptationSet>
    <AdaptationSet id="1" contentType="audio" segmentAlignment="true">
      <AudioChannelConfiguration
      schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      <SegmentTemplate timescale="1000000" duration="1000000"
      initialization="init-stream$RepresentationID$.m4s" media="chunk-stream$RepresentationID$-
      $Number%05d$.m4s" startNumber="1"/>
      <Representation id="3" mimeType="audio/mp4" codecs="mp4a.40.2"
      bandwidth="96000" audioSamplingRate="44100"/>
    </AdaptationSet>
  </Period>
```

```

    <UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xdate:2014"
value="http://time.akamai.com"/>
</MPD>

```

G.21 MPD Patch Document

This following example shows an MPD that includes an MPD Patch location.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MPD
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  id="live-stream"
  type="dynamic"
  minimumUpdatePeriod="PT3.0S"
  publishTime="2020-05-13T05:34:06+00:00"
  availabilityStartTime="2017-05-01T07:00:00+00:00"
  minBufferTime="PT1.0S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">

  <PatchLocation ttl="60"> example_G21_patch.mpp?publishTime=2020-05-
13T05%3A34%3A06%2B00%3A00</PatchLocation>

  <Period id="1588435200" start="PT95725984.571S">
    <AdaptationSet
      id="1"
      contentType="video"
      mimeType="video/mp4"
      maxFrameRate="60"
      maxHeight="1080"
      maxWidth="1920"
      bitstreamSwitching="true"
      segmentAlignment="true">
      <Accessibility schemeIdUri="urn:scte:dash:cc:cea-608:2015" value="CC1=eng"/>
      <SegmentTemplate
        timescale="90000"
        presentationTimeOffset="5491773166"
        initialization="live-stream/$RepresentationID$/init.mp4"
        media="live-stream/$RepresentationID$/Time$.m4s">
        <SegmentTimeline>
          <S d="360360" r="8" t="5491776169"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation
        id="video-0"
        codecs="avc1.4d4015"
        bandwidth="1070676"
        frameRate="30"
        height="270"
        width="480"
        startWithSAP="1"/>
      <Representation
        id="video-1"
        codecs="avc1.4d401e"
        bandwidth="2035716"
        frameRate="30"
        height="432"
        width="768"
        startWithSAP="1"/>
      <Representation
        id="video-2"

```



```

        codecs="avc1.64001f"
        bandwidth="3169038"
        frameRate="30"
        height="576"
        width="1024"
        startWithSAP="1"/>
    <Representation
        id="video-3"
        codecs="avc1.64001f"
        bandwidth="4532135"
        frameRate="30"
        height="720"
        width="1280"
        startWithSAP="1"/>
    <Representation
        id="video-4"
        codecs="avc1.640020"
        bandwidth="6728694"
        frameRate="60"
        height="720"
        width="1280"
        startWithSAP="1"/>
    <Representation
        id="video-5"
        codecs="avc1.64002a"
        bandwidth="10923530"
        frameRate="60"
        height="1080"
        width="1920"
        startWithSAP="1"/>
</AdaptationSet>
<AdaptationSet
    id="2"
    contentType="audio"
    mimeType="audio/mp4"
    bitstreamSwitching="true"
    segmentAlignment="true">
    <SegmentTemplate
        timescale="90000"
        presentationTimeOffset="5491773166"
        initialization="live-stream/$RepresentationID$/init.mp4"
        media="live-stream/$RepresentationID$/Time$.m4s">
        <SegmentTimeline>
            <S d="360960" r="1" t="5491776448"/>
            <S d="359040"/>
            <S d="360960" r="1"/>
            <S d="359040"/>
            <S d="360960" r="2"/>
        </SegmentTimeline>
    </SegmentTemplate>
    <Representation
        id="audio-0"
        codecs="mp4a.40.2"
        bandwidth="96000"
        audioSamplingRate="48000"
        startWithSAP="1">
        <AudioChannelConfiguration
            schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011"
            value="2"/>
    </Representation>
</AdaptationSet>
</Period>
</MPD>

```

An example for an MPD Patch document is provided below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Patch
  xmlns="urn:mpeg:dash:schema:mpd-patch:2020"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd-patch:2020 DASH-MPD-PATCH.xsd"
  mpdId="live-stream"
  originalPublishTime="2020-05-13T05:34:06+00:00"
  publishTime="2020-05-13T05:34:28.601Z">
  <replace sel="/MPD/@publishTime">2020-05-13T05:34:28.601Z</replace>
  <replace sel="/MPD/PatchLocation[0]">
    <PatchLocation ttl="60">live-stream/patch.mpd?publishTime=2020-05-
13T05%3A34%3A28.601Z</PatchLocation>
  </replace>
  <add
sel="/MPD/Period[@id='1588435200']/AdaptationSet[@id='1']/SegmentTemplate/SegmentTimeline">
    <S d="360360" r="6" t="5494659049"/>
  </add>
  <add
sel="/MPD/Period[@id='1588435200']/AdaptationSet[@id='2']/SegmentTemplate/SegmentTimeline">
    <S d="360960" t="5494660288"/>
    <S d="359040"/>
    <S d="360960" r="1"/>
    <S d="359040"/>
    <S d="360960" r="1"/>
  </add>
</Patch>
```

NOTE The example code is accessible here: https://standards.iso.org/iso-iec/23009/-1/ed-5/en/example_G21_patch.mpp

G.22 Failover Content

Upstream failures (e.g. packet loss) may create a situation where the i^{th} Segment is missing in one or more Representation but is present in at least one Representation within the same Adaptation Set.

For example, Segment i is missing in representation A, but exists in representation B, C, and D. Let's assume that segments $i-1$ and $i+1$ do exist for all 4 Representations (i.e., we have a 1-segment gap). When the client is made aware that Segments $A(i)$, $B(i)$, and $C(i)$ are missing, it requests $D(i)$ directly. Assuming that the player was playing Representation A, the segment sequence would be $A(i-2)$, $A(i-1)$, $D(i)$, $A(i+1)$, $A(i+2)$

Missing Segments are signalled in the MPD using the **FailoverContent** element. The example below shows a single Adaptation Set with a single missing Segment.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="dynamic"
  minimumUpdatePeriod="PT2S"
  timeShiftBufferDepth="PT30M"
  availabilityStartTime="2020-10-17T17:17:05Z"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
```

```

publishTime="2020-10-17T17:17:05Z">

<BaseURL>http://cdn1.example.com/</BaseURL>
<BaseURL>http://cdn2.example.com/</BaseURL>
<Period id="42">
  <!-- Video -->
  <AdaptationSet
    mimeType="video/mp4" codecs="avc1.4D401F" frameRate="30000/1001"
    segmentAlignment="true" startWithSAP="1">

    <SegmentTemplate startNumber="260319075"
      initialization="Travel_HD/$RepresentationID$/header.mp4"
      media="Travel_HD/$RepresentationID$/Number$.mp4"
      timescale="90000"
      presentationTimeOffset="6532028810">
      <SegmentTimeline>
        <S t="6532028810" d="222222" r="0" />
        <S t="6532251032" d="180180" r="420" />
        <S t="6534593372" d="135135" r="0" />
      </SegmentTimeline>
    </SegmentTemplate>

    <Representation id="C" bandwidth="828800" codecs="avc1.4d401e" width="640"
height="360" />
    <Representation id="B" bandwidth="2107200" codecs="avc1.4d401f" width="1280"
height="720" />

    <!-- Representation A is missing a segment -->
    <Representation id="A" bandwidth="3718000" codecs="avc1.640020" width="1280"
height="720">
      <SegmentTemplate>
        <FailoverContent>
          <FCS t="6532251032" d="180180" />
        </FailoverContent>
      </SegmentTemplate>
    </Representation>
  </AdaptationSet>

  <!-- Audio left as an exercise for the reader -->
</Period>
</MPD>

```

Annex H (normative)

Spatial Relationship Description

H.1 Spatial Relationship Description (SRD) scheme

The SRD scheme allows Media Presentation Description authors to express spatial relationships between Spatial Objects. A Spatial Object is represented by either an Adaptation Set or a Sub-Representation. As an example, a spatial relationship may express that a video represents a spatial part of another full-frame video (e.g. a region of interest, or a tile).

The **SupplementalProperty** and/or **EssentialProperty** descriptors with @schemeIdUri equal to "urn:mpeg:dash:srd:2014" and "urn:mpeg:dash:srd:dynamic:2016" may be used to provide spatial relationship information associated to the containing Spatial Object. SRD information shall be contained exclusively in these two MPD elements (**AdaptationSet** and **SubRepresentation**).

To preserve the compatibility with legacy clients, MPD shall use **SupplementalProperty** and **EssentialProperty** in such a way that at least one Representation can be interpreted by legacy clients after discarding the element containing **EssentialProperty**.

NOTE Sub-Representation level SRDs can be used to represent Spatial Objects in one Representation such as HEVC tiling streams. In that case, SRD descriptors can be present at Adaptation Set as well as Sub-Representation levels.

H.2 SRD @value syntax

H.2.1 General

The @value of the **SupplementalProperty** or **EssentialProperty** elements using the SRD scheme is a comma-separated list of values for SRD parameters.

When @value is not present, the SRD does not express any spatial relationship information at all and can be ignored.

H.2.2 Common parameters

The `source_id` parameter provides a unique identifier, within the Period, for the source of the content. It implicitly defines a coordinate system associated to this source. This coordinate system uses an arbitrary origin (0; 0); the x-axis is oriented from left to right and the y-axis from top to bottom. All SRD sharing the same `source_id` value have the same origin and axes orientations. Spatial relationships for Spatial Objects using SRD with different `source_id` values are undefined.

For a given `source_id` value, a reference space is defined, corresponding to the rectangular region encompassing the entire source content, whose top-left corner is at the origin of the coordinate system. The `total_width` and `total_height` values in a SRD provide the size of this reference space expressed in arbitrary units.

NOTE 1 There can be no Spatial Object in the MPD that covers the entire source of the content, e.g. when the entire source content is represented by two separate videos.

MPD authors can express, using the `spatial_set_id` parameter, that some Spatial Objects, within a given `source_id`, have a particular spatial relationship. For instance, an MPD author may group all Adaptation Sets corresponding to tiles at a same resolution level. This way, the `spatial_set_id` parameter may be used by the DASH Client to quickly select spatially related Spatial Objects. When there are two or more groups of full-frame videos which consist of one or more Spatial Objects with the same `total_width` and `total_height` value, different values of `spatial_set_id` may be used to distinguish the groups of full-frame video.

NOTE 2 ISO/IEC TR 23009-3 gives concrete examples showing how to use the `spatial_set_id`.

H.2.3 Specific parameters for static spatial description

For expressing static description within the scope of the Period, the following Scheme Identifier is used "urn:mpeg:dash:srd:2014".

The `object_x` and `object_y` parameters (respectively `object_width` and `object_height`) express 2D positions (respectively 2D sizes) of the associated Spatial Object in the coordinate system associated to the source. The values of the `object_x`, `object_y`, `object_width`, and `total_height` parameters are relative to the values of the `total_width` and `total_height` parameters, as defined above. Positions (`object_x`, `object_y`) and sizes (`object_width`, `object_height`) of SRD sharing the same `source_id` value may be compared after taking into account the size of the reference space, i.e. after the `object_x` and `object_width` values are divided by the `total_width` value and the `object_y` and `object_height` values divided by the `total_height` value of their respective descriptors.

NOTE Different `total_width` and `total_height` values can be used in different descriptors to provide positions and sizes information in different units.

Table H.1 — EssentialProperty@value and/or SupplementalProperty@value attributes for the static SRD scheme

EssentialProperty@value or SupplementalProperty@value parameter	Use	Description
<code>source_id</code>	M	non-negative integer in decimal representation providing the identifier for the source of the content
<code>object_x</code>	M	non-negative integer in decimal representation expressing the horizontal position of the top-left corner of the Spatial Object in arbitrary units
<code>object_y</code>	M	non-negative integer in decimal representation expressing the vertical position of the top-left corner of the Spatial Object in arbitrary units
<code>object_width</code>	M	non-negative integer in decimal representation expressing the width of the Spatial Object in arbitrary units
<code>object_height</code>	M	non-negative integer in decimal representation expressing the height of the Spatial Object in arbitrary units
<code>total_width</code>	O	optional non-negative integer in decimal representation expressing the width of the reference space in arbitrary units. At each Period and for a given <code>source_id</code> value, the following rules apply:

EssentialProperty@value or SupplementalProperty@value parameter	Use	Description
		<ul style="list-style-type: none"> — There shall be at least one descriptor providing a value for the <code>total_width</code> parameter. — If two or more descriptors provide different <code>total_width</code> values, all other descriptors shall explicitly provide the value of <code>total_width</code>. — If the <code>total_width</code> value is provided in only one descriptor, all other descriptors are assumed to use that <code>total_width</code> value. — The value of <code>total_width</code> shall be such that, for each descriptor using this value of <code>total_width</code>, the sum of <code>object_x</code> and <code>object_width</code> is smaller or equal to <code>total_width</code>. <p>When the value <code>total_width</code> is present, the value <code>total_height</code> shall be present.</p>
<code>total_height</code>	0	<p>optional non-negative integer in decimal representation expressing the height of the reference space in arbitrary units.</p> <p>At each Period and for a given <code>source_id</code> value, the following rules apply:</p> <ul style="list-style-type: none"> — There shall be at least one descriptor providing a value for the <code>total_height</code> parameter. — If two or more descriptors provide different <code>total_height</code> values, all other descriptors shall explicitly provide the value of <code>total_height</code>. — If the <code>total_height</code> value is provided in only one descriptor, all other descriptors are assumed to use that <code>total_height</code> value. — The value of <code>total_height</code> shall be such that, for each descriptor using this value of <code>total_height</code>, the sum of <code>object_y</code> and <code>object_height</code> is smaller or equal to <code>total_height</code>. <p>When the value <code>total_height</code> is present, the value <code>total_width</code> shall be present.</p>

EssentialProperty@value or SupplementalProperty@value parameter	Use	Description
spatial_set_id	0	optional non-negative integer in decimal representation providing an identifier for a group of Spatial Object. When not present, the Spatial Object associated to this descriptor does not belong to any spatial set and no spatial set information is given. When the value of spatial_set_id is present, the value of total_width and total_height shall be present.
Key M=Mandatory, 0=Optional		

The syntax for the value field of the static spatial relationship description shall follow the **STATIC-SRD-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```

STATIC-SRD-VALUE = source_id "," object_x "," object_y ","
object_width "," object_height [ "," total_width [ ","
total_height [ "," spatial_set_id ]]]

source_id = DECIMAL_DIGITS
object_x = DECIMAL_DIGITS
object_y = DECIMAL_DIGITS
object_width = DECIMAL_DIGITS
object_height = DECIMAL_DIGITS
total_width = DECIMAL_DIGITS
total_height = DECIMAL_DIGITS
spatial_set_id = DECIMAL_DIGITS

DECIMAL_DIGITS = 1*DIGIT

```

H.2.4 Specific parameters for dynamic spatial description

For expressing dynamic description within the scope of the Period, the following Scheme Identifier is used "urn:mpeg:dash:srd:dynamic:2016".

In the case the Spatial Object moves within the reference space, the coordinates of the Spatial Object are time dependent and thus cannot be expressed as static values in a SRD. As a result, the SRD does not provide directly the coordinates and size as in the static case but instead specifies the @id attribute of the metadata Representation that provides the coordinates and size of the Spatial Object. This @id attribute value is signalled in the coordinate_id parameter.

Examples of such scenarios include director's view, object tracking view, person tracking view in video conference applications, etc. For instance, the MPD author may offer two Spatial Objects, providing a wide angle view and a close-up view of the same scene of a sport event. The close-up view follows the action of the most popular athlete. But to ensure a satisfying Quality of Experience for the end-user, it is essential to describe the position of the close-up view with respect to the wide angle view at any point in time of the media content. This way, the end-user application can seamlessly switch from one video to another providing a smooth zooming in and out transition for the end-user.

Table H.2 — EssentialProperty@value and/or SupplementalProperty@value attributes for the dynamic SRD scheme

EssentialProperty@value or SupplementalProperty@value parameter	Use	Description
source_id	M	non-negative integer in decimal representation providing the identifier for the source of the content
coordinate_id	M	specifies the @id attribute of the Representation that provides the 2D coordinates of the Spatial Object as timed metadata track according to ISO/IEC 23001-10.
spatial_set_id	O	optional non-negative integer in decimal representation providing an identifier for a group of Spatial Object. When not present, the Spatial Object associated to this descriptor does not belong to any spatial set and no spatial set information is given.
Key M=Mandatory, O=Optional		

The coordinates and size of a moving Spatial Object shall be provided by a Representation offering a 2D Cartesian coordinate track '2dcc' as defined in ISO/IEC 23001-10. The @associationId attribute of this metadata Representation shall contain the value of the attribute @id of the Representation containing the moving Spatial Object. In addition, the @associationType attribute of this metadata Representation shall be set to 'cdsc'. See H.3.3 for examples.

The mapping between the '2dcc' sample parameters and the SRD parameters as defined in Table H.3 shall apply in order to determine the coordinates and size of a Spatial Object whose coordinates and size is provided as '2dcc' metadata Representation.

Table H.3 — Mapping between the '2dcc' sample parameters and the SRD parameters

2D Cartesian Coordinates Sample (ISO/IEC 23001-10)	SRD parameters
top_left_x	object_x
top_left_y	object_y
width	object_width
height	object_height
reference_width	total_width
reference_height	total_height

NOTE The 2D Cartesian Coordinates Sample is a generic sample providing the position and size of a rectangle, hence the more generic naming of these parameters compared to the SRD parameter's names in ISO/IEC 23009-1.

The syntax for the value field of the dynamic spatial relationship description shall follow the **DYNAMIC-SRD-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```
DYNAMIC-SRD-VALUE = source_id "," coordinate_id [ ","
spatial_set_id ]
```



```

source_id = DECIMAL_DIGITS
coordinate_id = STRING
spatial_set_id = DECIMAL_DIGITS

DECIMAL_DIGITS = 1*DIGIT
STRING = *VCHAR

```

H.3 Examples

H.3.1 General

All examples documented in this Annex are provided at <https://standards.iso.org/iso-iec/23009/-1/ed-5/> en with the following file naming convention referring to H.3.

<subsection>: example-H<subsection>.mpd

H.3.2 Zoomed video

This subclause provides a simple example of a static presentation with 2 videos, one video representing a zoomed part of the other video. The Media Presentation complies with the ISO Base media file format On Demand profile as defined in subclause 8.3.

```

<?xml version="1.0"?>
<MPD
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT10S"
  minBufferTime="PT1S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <ProgramInformation>
    <Title>Example of a DASH Media Presentation Description using Spatial Relationship Description
    to indicate that a video is a zoomed part of another</Title>
  </ProgramInformation>

  <Period>
    <!-- Panorama Video -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,3,3,3,3"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <Representation id="1" mimeType="video/mp4" codecs="avc1.42c033" width="1920"
      height="1080" bandwidth="1055223" startWithSAP="1">
        <BaseURL> panorama_video.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="839-990"/>
      </Representation>
    </AdaptationSet>
    <!-- Zoomed Video -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,1,1,1,1,3,3"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.42c028" width="1920" height="1080"
      bandwidth="769458" startWithSAP="1">
        <BaseURL> zoomed_video.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="838-989"/>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>

```

H.3.3 Tiled video

This subclause provides a simple example of a static presentation of a video available in different resolutions and of tiles of that video also available in different resolutions. The Media Presentation complies with the ISO Base media file format On Demand profile as defined in subclause 8.3.

```
<?xml version="1.0"?>
<MPD
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT10S"
  minBufferTime="PT1S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <ProgramInformation>
    <Title>Example of a DASH Media Presentation Description using Spatial Relationship Description
to indicate tiles of a video</Title>
  </ProgramInformation>

  <Period>
    <!-- Main Video -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,2,2,2,2"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <Representation id="1" mimeType="video/mp4" codecs="avc1.42c01e" width="640" height="360"
bandwidth="226597" startWithSAP="1">
        <BaseURL> full_video_small.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="837-988"/>
      </Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720"
bandwidth="553833" startWithSAP="1">
        <BaseURL> full_video_hd.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="838-989"/>
      </Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc1.42c033" width="3840"
height="2160" bandwidth="1055223" startWithSAP="1">
        <BaseURL> full_video_4k.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="839-990"/>
      </Representation>
    </AdaptationSet>
    <!-- Tile 1 -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,1,1,2,2"/>
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
      <Representation id="4" mimeType="video/mp4" codecs="avc1.42c00d" width="640" height="360"
bandwidth="218284" startWithSAP="1">
        <BaseURL> tile1_video_small.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="837-988"/>
      </Representation>
      <Representation id="5" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720"
bandwidth="525609" startWithSAP="1">
        <BaseURL> tile1_video_hd.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="838-989"/>
      </Representation>
      <Representation id="6" mimeType="video/mp4" codecs="avc1.42c028" width="1920"
height="1080" bandwidth="769514" startWithSAP="1">
        <BaseURL> tile1_video_fullhd.mp4</BaseURL>
        <SegmentBase indexRangeExact="true" indexRange="839-990"/>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

```

</AdaptationSet>
<!-- Tile 2 -->
<AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,1,0,1,1,2,2"/>
</AdaptationSet>
<!-- Tile 3 -->
<AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,1,1,1,1,2,2"/>
</AdaptationSet>
<!-- Tile 4 -->
<AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,1,1,1,2,2"/>
</AdaptationSet>
</Period>
</MPD>

```

H.3.4 Tiled panorama with moving Region-of-Interest

This subclause provides a simple example of a video moving within a panorama composed of two adjacent videos. The Media Presentation complies with the ISO Base media file format On Demand profile as defined in subclause 8.3.

```

<?xml version="1.0"?>
<MPD
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT10S"
  minBufferTime="PT1S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

  <Period>
    <!-- Tiled Panorama (2 full HD video next to each other) -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
      value="1, 0, 0, 1920, 1080, 3840, 1080, 0"/>
      <Representation id="left_panorama" mimeType="video/mp4" codecs="avc1.42c01e"
      bandwidth="5000000" width="1920" height="1080">
        <BaseURL>left_panorama.mp4</BaseURL>
      </Representation>
    </AdaptationSet>

    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
      value="1, 1920, 0, 1920, 1080, 3840, 1080, 0"/>
      <Representation id="right_panorama" mimeType="video/mp4" codecs="avc1.42c01e"
      bandwidth="5000000" width="1920" height="1080">
        <BaseURL>right_panorama.mp4</BaseURL>
      </Representation>
    </AdaptationSet>

    <!-- Moving Region-of-Interest -->
    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
    subsegmentStartsWithSAP="1">
      <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2016"

```

```
        value="1, roi-coordinates"/>
        <Representation id="zoomed" mimeType="video/mp4" codecs="avc1.42c01e" bandwidth="5000000"
width="1920" height="1080">
            <BaseURL>zoomed_part.mp4</BaseURL>
        </Representation>
    </AdaptationSet>

    <AdaptationSet segmentAlignment="true" subsegmentAlignment="true"
subsegmentStartsWithSAP="1">
        <Representation id="roi-coordinates" associationId="zoomed" associationType="cdsc"
codecs="2dcc" bandwidth="100">
            <BaseURL>roi_coordinates.mp4</BaseURL>
        </Representation>
    </AdaptationSet>

</Period>
</MPD>
```

Annex I (normative)

Flexible Insertion of URL Parameters

I.1 General

This Annex describes how it is possible to configure URL parameters of media segment URLs, in a similar fashion to the URL template mechanism. The core specification of DASH already allows building media segment URLs containing static URL parameters. This Annex aims at providing more flexibility and dynamicity in the way URL parameters are inserted.

Parameters are *instantiated* — a name-value pair is constructed by the client. Then parameters are *output* — an instantiated key-value pair is written to either query parameters or headers, depending on the *output mode*.

The mechanism described in this Annex allows several methods of instantiation:

- "Inheritance" from MPD URL parameters when the MPD is delivered over HTTP, i.e. extraction of one or more key-value pairs from the query string of the URL used to fetch MPD.
- Just-in-time server-side instantiation using the XLink mechanism.
- Header instantiation: given a header name, the parameter value is the value of the header.
- Client-side computation — specific feedback (uniquely identified by URN) is expected from the client side.

There are two output modes described by this Annex:

- Query parameters: parameters are written as key-value pairs in HTTP GET requests issued by the DASH Client.
- HTTP header: parameters are written as a list of key-value pairs in a DASH-specific HTTP header.

Flexible insertion of URL parameters is signalled through the use of **EssentialProperty** or **SupplementalProperty** descriptors, with @schemeIdUri values defined below:

- Baseline segment URL parametrization scheme identified by URN "urn:mpeg:dash:urlparam:2014" and described in I.2. This scheme applies only to requests for media segments.
- Extended parametrization scheme applying to requests for media segments, MPD, XLink and callback events, and is a superset of the baseline scheme. This scheme is identified by URN "urn:mpeg:dash:urlparam:2016" and described in I.3.

I.2 Segment URL parametrization

I.2.1 General

Flexible insertion of URL parameters is signalled through the use of **EssentialProperty** or **SupplementalProperty** descriptors, with @schemeIdUri equal to "urn:mpeg:dash:urlparam:2014".

A child element `up:UrlQueryInfo` shall be present in these descriptors, defined within the "urn:mpeg:dash:schema:urlparam:2014" namespace. The namespace prefix should be "up:".

As defined by this specification, each of these descriptors may be present at the MPD, Adaptation Set or at the Representation level. Only **SupplementalProperty** descriptor may be present at the Period level. When insertion of URL parameters is required for a Period, **EssentialProperty** descriptors shall be inserted in all Adaptation Sets of that Period. At most one descriptor shall be present at each level.

When the insertion of URL parameters requires scheme-dependent computation, one or several additional **EssentialProperty** or **SupplementalProperty** descriptors shall be present. These descriptors shall carry an appropriate @schemeIdUri attribute referencing the scheme to be used, and provide sufficient information to appropriately compute the required URL parameters (see I.2.4.2). Support of these schemes is not in the scope of this specification.

The initial part of the XML schema of the URL Query Information is provided below, including namespace and other definitions. Specific types, elements and attributes are introduced in the remainder of this subclause. The complete normative URL Query Information schema is provided at <https://standards.iso.org/iso-iec/23009/-1/ed-5/en> (DASH-MPD-UP.xsd). In case of inconsistencies, this schema takes precedence both over the XML syntax snippets provided in this clause and all prose text in this document.

Implementors are supported by additional files available at: <https://standards.iso.org/iso-iec/23009/-1/ed-5/en>. These files include the schema as well as all examples provided in this Annex.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="urn:mpeg:dash:schema:urlparam:2014" targetNamespace="urn:mpeg:dash:schema:urlparam:2014"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://www.w3.org/XML/2008/06/xlink.xsd"/>
  <xs:element name="UrlQueryInfo" type="UrlQueryInfoType"/>
  <xs:complexType name="UrlQueryInfoType">
    ...
  </xs:complexType>
</xs:schema>
```

I.2.2 URL Query Information

I.2.2.1 Overview

The **UrlQueryInfo** element describes how to build a URL query string, which is used in the media segments URLs building process.

This query string can come from one of the three sources below:

- The URL of the MPD when the @useMPDUrlQuery is set;
- The @queryString attribute when present;

— The `@queryString` attribute, after any XLink resolution in case `@xlink:href` is present.

The `@queryTemplate` attribute describes which URL parameters contained in the query string are used in the media segment URL building process, as well as the order of these parameters.

The semantics of the attributes and elements for the URL Query Information provided in I.2.2.2, Table I.1. The XML syntax of the URL Query Information is provided in I.2.2.3.

I.2.2.2 Semantics

Table I.1 — Semantics of `UrlQueryInfo` element

Element or Attribute Name	Use	Description
<code>UrlQueryInfo</code>		provides URL query string information
<code>@queryTemplate</code>	0 (string)	<p>provides URL parameters template information</p> <p>This string shall contain one or more <code>\$<ParamIdentifier>\$</code> template identifiers, as listed in Table I.2. These template identifiers are to be replaced to build a query string (see I.2.3). If <code>\$<ParamIdentifier>\$</code> is not in Table I.2, it will be replaced with an empty string. If the template has an opening <code>\$</code> without a matching closing <code>\$</code>, the result is undefined, and the client will act as if it did not understand the EssentialProperty's scheme.</p> <p>When selection of URL parameters is enabled through the use of <code>\$query:param\$</code> template identifiers, URL parameters shall be defined as name=value pairs separated by <code>&</code>, as defined by W3C HTML 4.01 Specification (section on Forms#Form submission).</p>
<code>@useMPDUrlQuery</code>	0 (bool) default: false	<p>indicates that the URL parameters of the MPD URL are used in the construction of media segment URLs.</p> <p>This attribute may only be present when the MPD is delivered over HTTP, and defaults to "false" when the MPD is not delivered over HTTP.</p> <p>If <code>@queryString</code> is present and the value of this attribute is "true", concatenation of MPD parameter string and <code>@queryString</code> shall be used for constructing the query string of media segment URLs.</p> <p>NOTE Simple parameter signalling can be used (<code>"a=X&b=Y"</code>), as well as scheme-dependent signalling (<code>"a=\$urn:XYZ\$b=\$urn:ABC\$"</code>).</p> <p>When scheme-dependent signalling is used, the scheme shall be inserted between two enclosing <code>\$</code> characters. See I.2.3.3 for further details.</p> <p>When <code>UrlQueryInfo</code> element is present at more than one level of the hierarchical data model (e.g. MPD and Period), there shall be at most one <code>UrlQueryInfo</code> element for which <code>@useMPDUrlQuery</code> is true within this hierarchy.</p> <p>See I.2.3.3 for further details.</p>

Element or Attribute Name	Use	Description
@queryString	O (string)	<p>provides a query string to be used in the construction of media segment URLs.</p> <p>NOTE Simple parameter signalling can be used ("a=X&b=Y"), as well as scheme-dependent signalling ("a=\$urn:XYZ\$&b=\$urn:ABC\$").</p> <p>When scheme-dependent signalling is used, the scheme shall be inserted between two enclosing \$ characters. See I.2.3.3 for further details.</p>
@xlink:href	O	specifies a reference to a remote <code>UrlQueryInfo</code> element
@xlink:actuate	OD default: onRequest	<p>specifies the processing instructions, which can be either "onLoad" or "onRequest".</p> <p>This attribute shall not be present if the @xlink:href attribute is not present.</p>
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using <code>xlink:href</code> . If linking is used, then all attributes are "optional" and <minOccurs=0>. Elements are bold ; attributes are non-bold and preceded with an @.		

Table I.2 — Parameter identifiers

\$<ParamIdentifier>\$	Substitution parameter
\$\$	Is an escape sequence, i.e. "\$\$" is replaced with a single "\$".
\$querypart\$	<p>This identifier is substituted with the query part of the computed query string (referred to as <i>initialQueryString</i> in I.2.3). This identifier shall not appear more than once in the template string, and shall not be mixed with other identifiers (e.g. \$query:<param>\$ below).</p> <p>The query part starts after the "?" sign and lasts until the "#" sign</p>
\$query:<param>\$	<p>This identifier is substituted with the value of the <param> parameter if this parameter is present in the query part of the computed query string (referred to as <i>initialQueryString</i> in I.2.3). If same <param> value appears more than once in the query string, the last value will be used. If <param> is not present, the empty string will be used.</p> <p>When this parameter selection is used, URL parameters shall be inserted in the query part of the URL as name=value pairs separated by the "&" symbol, according to W3C HTML 4.01 specification (section on Forms#Form submission)</p>

I.2.2.3 XML Syntax

```
<xs:element name="UrlQueryInfo" type="UrlQueryInfoType"/>
<xs:complexType name="UrlQueryInfoType">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="queryTemplate" type="xs:string"/>
  <xs:attribute name="useMPDUrlQuery" type="xs:boolean" default="false"/>
  <xs:attribute name="queryString" type="xs:string"/>
  <xs:attribute ref="xlink:href"/>
  <xs:attribute ref="xlink:actuate" default="onRequest"/>
  <xs:attribute ref="xlink:type" fixed="simple"/>
  <xs:attribute ref="xlink:show" fixed="embed"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

I.2.3 Modified template-based segment URL construction, according to `UrlQueryInfo` element

I.2.3.1 General

When signaled through an appropriate descriptor, containing a `UrlQueryInfo` element, the following media segment URL building process is performed.

The process is defined in the following steps:

- 1) If this `UrlQueryInfo` element is a remote element, it is dereferenced. This process is defined in 5.5.
- 2) Initial query string (referred to as *initialQueryString*) is derived. This process is described in I.2.3.2.
- 3) Final query string (referred to as *finalQueryString*) is computed, according to `@queryTemplate` and *initialQueryString*. This process is described in I.2.3.3.
- 4) Final query string (*finalQueryString*) is processed to build media segment URLs. This process is described in I.2.3.4.

I.2.3.2 Computation of an initial query string (*initialQueryString*)

The initial query string *initialQueryString* is constructed by concatenating the query strings, if present and available, coming from the MPD URL (if `@useMPDUrlQuery` is set to "true") and `@queryString` (possibly after dereferencing). If `@useMPDUrlQuery` is set to "true" and `@queryString` if present, *initialQueryString* shall be a concatenation of query string from the MPD and `@queryString` string, in this order.

When multiple strings are appended together, an "&" symbol shall be inserted at the start of the second and following strings to be appended.

I.2.3.3 Computation of a final query string (*finalQueryString*)

A final query string *finalQueryString* is then computed by substituting URL parameters templates present in `@queryTemplate` by their values provided in *initialQueryString*, according to Table I.2.

When two or more occurrences of URL query descriptors exist within an MPD, the *finalQueryString* string used at the Representation level is a concatenation of the corresponding URL query strings of the occurrences in their orders of appearance in the MPD hierarchy. The query coming from the MPD URL is appended first. Thus, for each representation the concatenation shall be computed as a concatenation of Representation-level query string with (in this order) AdaptationSet-level string, Period-level query string and, lastly, MPD-level query string.

Simple parameter signalling may be used (`@queryString="a=X&b=Y"`), as well as scheme-dependent signalling (`@queryString="a=$urn:XYZ$b=$urn:ABC$"`). In the latter case, the client needs to be aware of the provided schemes, and has to compute appropriate values for them. Further, in this case, additional **EssentialProperty** or **SupplementalProperty** descriptors, at the same level as the query descriptor, shall be present to reflect that scheme-dependent signalling is used and required to be supported by the client. These descriptors shall have the `@schemeIdUri` attribute set to the same value as used in the URL parameter insertion description (i.e. to "urn:XYZ" or "urn:ABC" in the above example). Support of these specific schemes is out of the scope of this specification.

A straightforward implementation of the process would do the following:

- 1) Create a parameter table out of concatenated `initialQueryString`. The latter is an ' & ' - separated list of `<param>=<value>` strings, and each `<param>=<value>` string is converted into a single entry in the parameter table. E.g. for a string "param0=42" in `initialQueryString` we will have `parameter["param0"]=42`. If a string "param0=42" is followed by a string "param0=43" later in `initialQueryString`, then `parameter["param0"]=43`. If `<param>` is a URN, then `<value>` is computed by the client (and is an empty string otherwise).
- 2) Search for the "\$query:" substring in the `@queryTemplate` attribute. For each appearance of this substring, the characters till the first '\$' character are the parameter name (`<param>` in our notation). Substitute the complete `$query<param>$` string (including the opening and the closing '\$' characters) with `parameter[<param>]`. E.g. given `@queryTemplate="p0=$query:param0"`, and given `parameter["param0"]=43` the result would be `finalQueryString="p0=43"`.

I.2.3.4 Modified media segment URLs building process

The computed final query string *finalQueryString* is then concatenated to media Segment URLs.

If the original media segment URL does not contain any query (as defined in IETF RFC 3986), an "?" character shall be inserted accordingly between the original media segment URL and the *finalQueryString* when performing this concatenation.

If the original media segment URL already contains a query (as defined in IETF RFC 3986), an "&" character shall be inserted between the original media segment URL and the *finalQueryString* when performing this concatenation.

When Annex E is used together with flexible insertion of URL query parameters, processing of URL query parameters descriptors shall occur first, followed by Annex E byte range requests building process.

I.2.4 Examples

I.2.4.1 Example 1

Here, the intent is to re-use the URL parameters of the MPD URL in the media segments URLs.

Assuming DASH MPD is accessible through:
<http://www.example.com/dash/urlparam1.mpd?token=1234&ip=1.2.3.4>

- 1) Computation of an initial query string

initialQueryString="token=1234&ip=1.2.3.4"

- 2) Computation of a final query

finalQueryString="token=1234&ip=1.2.3.4"

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd urn:mpeg:dash:schema:urlparam:2014
  DASH-MPD-UP.xsd"

  type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
  <Period>
    <AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
    maxWidth="1280" maxHeight="720" maxFrameRate="25" par="16:9">
      <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
      xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
        <up:UrlQueryInfo queryTemplate="$querypart$" useMPDUrlQuery="true"/>
      </EssentialProperty>
      <SegmentTemplate duration="2" startNumber="1"
      media="video_${Number}_${Bandwidth}$bps.mp4"/>
      <Representation id="v0" codecs="avc3.4d401f" width="1280" height="720"
      frameRate="25" sar="1:1" bandwidth="3000000"/>
      <Representation id="v1" codecs="avc3.4d401f" width="640" height="360"
      frameRate="25" sar="1:1" bandwidth="1500000"/>
    </AdaptationSet>
  </Period>
</MPD>

```

3) Modified media segment URLs building process

http://www.example.com/dash/video_1_3000000bps.mp4?token=1234&ip=1.2.3.4

http://www.example.com/dash/video_2_3000000bps.mp4?token=1234&ip=1.2.3.4

http://www.example.com/dash/video_3_3000000bps.mp4?token=1234&ip=1.2.3.4

1.2.4.2 Example 2

Here, the intent is to dynamically compute some URL parameters before adding them to the media segments URLs.

Assuming DASH MPD is accessible through <http://www.example.com/dash/urlparam2.mpd>, and that <http://www.example.com/dash/xlinked.mpd> contains the following **UrlQueryInfo** element:

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd urn:mpeg:dash:schema:urlparam:2014
  DASH-MPD-UP.xsd" type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
  <Period>
    <AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1"
    maxWidth="1280" maxHeight="720" maxFrameRate="25" par="16:9">
      <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
      xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
        <up:UrlQueryInfo
        xlink:href="http://www.example.com/dash/xlinked.mpd" xlink:actuate="onRequest"
        xmlns:xlink="http://www.w3.org/1999/xlink"/>
      </EssentialProperty>
      <SegmentTemplate duration="2" startNumber="1"
      media="video_${Number}_${Bandwidth}$bps.mp4"/>
      <Representation id="v0" codecs="avc3.4d401f" width="1280" height="720"
      frameRate="25" sar="1:1" bandwidth="3000000"/>
    </AdaptationSet>
  </Period>
</MPD>

```

```

        <Representation id="v1" codecs="avc3.4d401f" width="640" height="360"
frameRate="25" sar="1:1" bandwidth="1500000"/>
    </AdaptationSet>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
        <up:UrlQueryInfo xmlns:up="urn:mpeg:dash:schema:urlparam:2014"
queryTemplate="$querypart$" queryString="param=justintimecomputedvalue"/>
    </SupplementalProperty>
</Period>
</MPD>

```

- 1) Computation of an initial query string (computed on request, according to xlink:actuate):

initialQueryString="param=justintimecomputedvalue"

- 2) Computation of a final query string

finalQueryString="param=justintimecomputedvalue"

- 3) Modified media segment URLs building process

http://www.example.com/dash/video_1_3000000bps.mp4?param=justintimecomputedvalue

http://www.example.com/dash/video_2_3000000bps.mp4?param=justintimecomputedvalue

http://www.example.com/dash/video_3_3000000bps.mp4?param=justintimecomputedvalue

I.2.4.3 Example 3

Here the intent is asking the client for some feedback through URL parameters (GPS location here).

Assuming DASH MPD is accessible through [http://www.example.com/dash/urlparam3.mpd?pd=\\$urn:example:gps\\$](http://www.example.com/dash/urlparam3.mpd?pd=$urn:example:gps$), and that "urn:example:gps" informs the client that it should insert its GPS coordinates.

A **SupplementalProperty** descriptor with @schemeIdUri equal to "urn:example:gps" is added to the MPD to indicate that the computation of URL parameters depends on an externally defined scheme.

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd urn:mpeg:dash:schema:urlparam:2014
DASH-MPD-UP.xsd"

type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
    <Period>
        <AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1" maxWidth="1280"
maxHeight="720" maxFrameRate="25" par="16:9">
            <SupplementalProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
                <up:UrlQueryInfo queryTemplate="$querypart$" useMPDUrlQuery="true"/>
            </SupplementalProperty>
            <SupplementalProperty schemeIdUri="urn:example:gps"/>
        </AdaptationSet>
    </Period>
</MPD>

```

```

<SegmentTemplate duration="2" startNumber="1" media="video_${Number}_${Bandwidth}$bps.mp4">
</SegmentTemplate>
<Representation id="v0" codecs="avc3.4d401f" width="1280" height="720" frameRate="25"
sar="1:1" bandwidth="3000000"/>
<Representation id="v1" codecs="avc3.4d401f" width="640" height="360" frameRate="25" sar="1:1"
bandwidth="1500000"/>
</AdaptationSet>
</Period>
</MPD>

```

- 1) Computation of an initial query string:

initialQueryString="pd=\$urn:example:gps\$"

- 2) Computation of a final query

finalQueryString="pd=\$urn:example::gps\$"

- 3) Modified media segment URLs building process

http://www.example.com/dash/video_1_3000000bps.mp4?pd=computedGPSlocation

http://www.example.com/dash/video_2_3000000bps.mp4?pd=computedGPSlocation

http://www.example.com/dash/video_3_3000000bps.mp4?pd=computedGPSlocation

In this particular example, the client needs to be aware of the process to compute and provide GPS location. How to achieve this is not part of DASH specification.

I.2.4.4 Example 4

Same example as example 1, except that only one single URL parameter is used.

Assuming DASH MPD is accessible through:
<http://www.example.com/dash/urlparam4.mpd?token=1234&ip=1.2.3.4>

```

<?xml version="1.0" encoding="UTF-8"?>
<MPD
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd urn:mpeg:dash:schema:urlparam:2014
DASH-MPD-UP.xsd"

type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
  <Period>
    <AdaptationSet mimeType="video/mp4" segmentAlignment="true" startWithSAP="1" maxWidth="1280"
maxHeight="720" maxFrameRate="25" par="16:9">
      <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
        <up:UrlQueryInfo queryTemplate="token=$query:token$" useMPDUrlQuery="true"/>
      </EssentialProperty>
      <SegmentTemplate duration="2" startNumber="1" media="video_${Number}_${Bandwidth}$bps.mp4">
      </SegmentTemplate>
      <Representation id="v0" codecs="avc3.4d401f" width="1280" height="720" frameRate="25"
sar="1:1" bandwidth="3000000"/>

```

```
<Representation id="v1" codecs="avc3.4d401f" width="640" height="360" frameRate="25" sar="1:1"
bandwidth="1500000"/>
</AdaptationSet>
</Period>
</MPD>
```

1) Computation of an initial query string

initialQueryString="token=1234&ip=1.2.3.4"

2) Computation of a final query string

finalQueryString="token=1234"

3) Modified media segment URLs building process

`http://www.example.com/dash/video_1_3000000bps.mp4?token=1234`

`http://www.example.com/dash/video_2_3000000bps.mp4?token=1234`

`http://www.example.com/dash/video_3_3000000bps.mp4?token=1234`

I.3 Extended HTTP GET request parametrization

I.3.1 General

Extended parametrization scheme provides a superset of the capabilities of the baseline segment-only scheme described in I.2 above. This scheme is identified by URN "urn:mpeg:dash:urlparam:2016".

The scheme extends the baseline scheme described in I.2 in the following ways:

- Parameters may be selectively embedded in MPD, XLink and callback requests, in addition to segment requests;
- Parameters may be instantiated from designated HTTP headers (from HTTP response);
- Parameters defined using the **ExtHttpHeaderParameter** element will be embedded into a custom HTTP header and not into the URL.

The scheme uses two elements, **ExtUrlQueryInfo** and **ExtHttpHeaderInfo**, both of which are extensions of **UrlQueryInfoType** type defined in I.2.

The extended scheme is a functional modification of the one defined in I.2. For simplicity purposes the complete extended mechanism is provided in I.3.4.

I.3.2 Semantics

Table I.3 — Semantics of **ExtendedUrlInfoType** element

Element or Attribute Name	Use	Description
ExtendedUrlInfoType		provides information for derivation of parameter string. This is an extension of UrlQueryInfoType element defined in Table I.1.
@includeInRequests	OD (default: "segment")	<p>specifies which HTTP GET requests shall carry parameters. Value is a white spaced concatenated list of the following keys:</p> <ol style="list-style-type: none"> 1) "segment" (all segment requests), 2) "xlink" (all XLink resolution requests), 3) "mpd" (all MPD requests), 4) "callback" (all requests triggered by DASH callback events), 5) "chaining" (requests for chained-to MPDs, 6) "fallback" (requests for the alternative MPDs. <p>Default value is "segment", i.e. parameters will be only sent with segment requests</p> <p>NOTE Depending on the actual element used, parameter output goes either to query parameters (for ExtUrlQueryInfo) or HTTP headers (for ExtHttpHeaderInfo)</p>

Element or Attribute Name	Use	Description
@headerParamSource	OD (default: "segment")	<p>specifies HTTP responses from which HTTP header values, identified by the template <i>\$header:header-name\$</i>, should be extracted from. Value is a white spaced concatenated list of the following keys :</p> <ol style="list-style-type: none"> 1) "segment" (all segment requests), 2) "xlink" (all XLink resolution requests), 3) "mpd" (all MPD requests), 4) "callback" (all requests triggered by DASH callback events). <p>Default value: empty string (no header parameters inspected)</p> <p>If this attribute is present then: (a) @queryTemplate attribute shall be present and shall contain the <i>\$header:<header-name>\$</i> identifier, and (b) neither @useMPDUrlQuery nor @queryString attribute shall be present.</p>
@sameOriginOnly	OD	<p>specifies that parameters must only be sent to the same origin they were instantiated from. In case of HTTP headers as source, the origin is defined as the origin of the HTTP request identified by the attribute @headerParamSource. In case the parameters are instantiated from the MPD or from the MPD URL, the origin is defined in both case by the MPD URL.</p> <p>Two origins are the same as defined by IETF RFC 6454, i.e. same scheme/host/port triple (see 5. Comparing Origins)</p> <p>Default value: false (no origin restrictions)</p>
<p>Key</p> <p>For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory.</p> <p>For elements: <minOccurs>...<maxOccurs> (N=unbounded)</p> <p>The conditions only hold without using <i>xlink:href</i>. If linking is used, then all attributes are "optional" and <minOccurs=0>.</p> <p>Elements are bold; attributes are non-bold and preceded with an @.</p>		

The following identifiers are defined, in addition to the ones defined in Table I.2

Table I.4 — Parameter identifiers

\$<ParamIdentifier>\$	Substitution parameter
<i>\$header:<header-name>\$</i>	This identifier is substituted with the latest received value of the header-name HTTP header in the HTTP responses indicated by the @headerParamSource attribute.

I.3.3 XML Syntax

```
<xs:element name="ExtUrlQueryInfo" type="ExtendedUrlInfoType"/>
<xs:element name="ExtHttpHeaderInfo" type="ExtendedUrlInfoType"/>
<xs:complexType name="ExtendedUrlInfoType">
  <xs:complexContent>
    <xs:extension base="UrlQueryInfoType">
      <xs:attribute name="includeInRequests" type="xs:string" default="segment"/>
      <xs:attribute name="headerParamSource" type="xs:string" default="segment"/>
      <xs:attribute name="sameOriginOnly" type="xs:boolean" default="false"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

I.3.4 Extended parameter generation

I.3.4.1 Theory of operation

HTTP(S) requests issued by a DASH Client may belong to one of the following *types*:

- MPD request to the MPD URL (e.g. as specified in **MPD.Location**).
- XLink dereferencing request to URL specified in the @xlink:href attribute.
- Segment request.
- Callback request to URL specified in a DASH callback event (inband or MPD).

Parameter generation based on elements of type **ExtendedUrlInfoType** is defined in the following steps:

- 1) If this **ExtendedUrlInfoType** element is a remote element, it is dereferenced. This process is defined in subclause 5.5.
- 2) Initial query string (referred to as *initialQueryString*) is derived. This process is described in I.3.4.2.
- 3) Final query string (referred to as *finalQueryString*) is computed, according to @queryTemplate and *initialQueryString*. This process is described in I.3.4.3.
- 4) Final query string (*finalQueryString*) is processed to build HTTP requests. This process is described in I.3.5.

I.3.4.2 Computation of an initial query string (*initialQueryString*)

The initial query string *initialQueryString* is constructed by concatenating the query strings for the same type of request. One of the following ways of *initialQueryString* construction shall be used:

- 1) If @useMPDUrlQuery is "true", the query part of MPD URL shall be used as *initialQueryString*. If @queryString is also present, *initialQueryString* is a concatenation of the query part of MPD URL and @queryString, in this order.
- 2) If @headerParamSource is present and non-empty, *initialQueryString* shall consist of <header-name>=<value> pairs where <header-name> is the HTTP response header given in the @queryTemplate by the \$header:<header-name>\$ substitution parameter. The inspected responses are of the types appearing in @headerParamSource.
- 3) Otherwise the value of *initialQueryString* is given by @queryString.

When multiple strings are appended together, an "&" symbol shall be inserted at the start of the second and following strings to be appended.

I.3.4.3 Computation of a final query string (*finalQueryString*)

A final query string *finalQueryString* is then computed by substituting URL parameter identifiers present in *@queryTemplate* by their values provided in *initialQueryString*, according to Tables I.2 and I.4. The process itself is equivalent to the one in described in I.2.3.2, with a notable addition of the *\$header:<header-name>\$* identifier.

When two or more occurrences of element of type **ExtendedUrlInfoType** exist within an MPD, the *finalQueryString* used at each level is a concatenation of the corresponding at the current and all upper levels of the hierarchical data model. For example, let us assume that a **ExtUrlQueryInfo** element is present at both Period and MPD levels, the Period has the *@xlink:href* attribute and both the **MPD.ExtUrlQueryInfo@includeInRequests** and **Period.ExtUrlQueryInfo@includeInRequests** contain the string "xlink". In this case, *initialQueryString* for the dereferencing will be a concatenation of *finalQueryString* from **MPD.ExtUrlQueryInfo** and **Period.ExtUrlQueryInfo** in this order.

A straightforward implementation of the process would do the following:

- 1) Create a parameter table out of concatenated *initialQueryString* as defined in step 1 in I.2.3.2.
- 2) Replace the "\$query:<param>" substrings with *parameter[<header-name>]* as described in step 2 in I.2.3.2.
- 3) Replace *\$header:<header-name>\$* substring with *parameter[<header-name>]*.

I.3.5 Extended Parameter Output

In case **ExtUrlQueryInfo** element is used, the computed final query string *finalQueryString* is concatenated to media Segment URLs (same process as in I.2.3.3). Percent encoding (per IETF RFC 3986) may be needed as a last step of URL construction

In case **ExtHttpHeaderInfo** element is used, the output shall appear in a custom HTTP request header in the following way:

- 1) Each "&" (ampersand) character in *finalQueryString* shall be replaced with a ", " (comma followed by space) string.
- 2) The resulting string is written as a value of a custom HTTP header `MPEG-DASH-Param`.

Annex J (informative)

Open GOP resolution change

A scheme is defined to be used with a Supplemental Property Descriptor as "urn:mpeg:dash:resolutionSwitching:2016".

It indicates which Representations allow for a seamless resolution switching at the start of any Segment starting with a SAP type in the range of 1 to 3, inclusive.

If present, the descriptor shall only be placed on Adaptation Set or Representation level in the MPD hierarchy.

@value of the supplemental property descriptor is a white space separated list of two values as specified in Table J.1.

Table J.1 — SupplementalProperty@value attributes resolutionSwitching:2016

SupplementalProperty @value parameter	Use	Description
switchableTo	M	specifies all Representations with representation switch points specified with RandomAccess with @type equal to "open", with Media Segment starting with SAP of type 3, or with Media Subsegment starting with SAP of type 3 for which the Representation contains additional media stream to switch to, as a comma-separated list of values of @id attributes of these Representations.
switchingMimeType	M	specifies, as a comma-separated list of values, the MIME type of the concatenation of the Initialization Segment, if present, and some consecutive Media (Sub)Segments in the Representation and some consecutive Media (Sub)Segments that start with a representation switch point as defined above and belong to the Representation pointed by the respective list item within switchableTo. NOTE The Representation contains a media stream conforming to @mimeType and additionally one or more media streams conforming to switchingMimeType that are only intended for switching to another representation, as indicated by switchableTo.
Key M=Mandatory, O=Optional		

The resolution switching descriptor shall not be present unless all access units in a segment N with presentation time within $[T_{EPT}, T_{DEC})$ is constrained in such a way that they only depend on access units of segment N or segment N-1.

NOTE If a Representation is changed at segment N, where this descriptor is present, it might be necessary to decode an additional media stream during segment N-1 different to the one conforming to the @codecs attribute indicated at the "switch-from" representation, which presence is indicated by the presence of switchingMimeType, in order to be able to decode all access units preceding the first SAP (i.e. in the interval $[T_{EPT}, T_{DEC})$) of segment N of the "switch-to" Representation.

Annex K **(normative)**

DASH Service Description

K.1 General

This annex defines the ISO/IEC 23009-1 DASH Service Description. In the DASH model as described in this document, the DASH Client has significant control over the algorithms and user perception for a DASH service. The DASH Client may for example decide on the applied rate adaptation algorithm, the buffer strategy, the buffer duration and the resulting latency and channel access times. However, by leaving all decisions to the client, this may result inconsistent behaviour as different client implementations may for example chose different strategies and therefore, as an example, one may observe significantly different latencies for the same service on different clients.

A DASH Client may be embedded in an application that controls the usage of the client for the proper playback of the service. DASH Client libraries and applications may provide APIs to control the playback. However, there are weaknesses to such an approach as the application and DASH Client then use proprietary APIs.

Hence, this annex defines the following:

- a service description reference model providing an overview of the normative aspects defined in this annex (in K.2);
- the semantics of the service description and the associated parameters to describe the service (in K.3). This is defined as an abstract set of APIs that can be used by the application, regardless of how the application received this information;
- the mapping of the semantics to DASH MPD (in K.4) as one option to provide the service parameters. This permits that application-independent DASH services can be provided in a consistent manner;
- some example client implementation usage guidelines on how these parameters may be used (in K.5);

The usage of the service description information may be left to client implementations, but it may also be the case that application standards formulate stronger requirements on the client in order to fulfil such service parameters. Service Descriptions can also be scoped for specific clients, for example those implementing specific rules, or clients in specific environments.

The normative aspects of this annex include the definition of specific service parameters in K.3 as well as the carriage of these parameters in the MPD in K.4.

K.2 DASH Client model with Service Description

Figure K.1 shows an extended client model that includes the ability to provide explicit service description information to the DASH access client. The information might originate from the service provider and can be delivered by application-defined signalling, can be generated in the application or can be delivered by the MPD as defined in K.4.

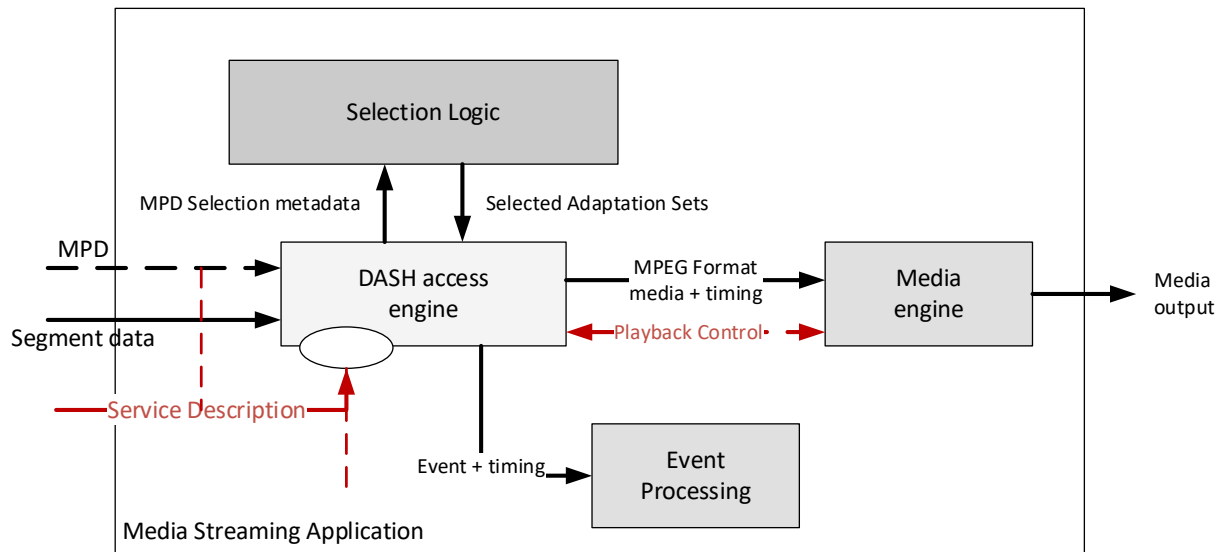


Figure K.1 Extended Client Model with service description (shown in red)

K.3 Semantics of DASH Service Description

K.3.1 General

This subclause provides the general service description definitions.

The semantics follow the principle of the metrics definition using an abstract syntax. Items in this abstract syntax have one of the following primitive types (*Integer*, *Real*, *Boolean*, *Enum*, *String*) or one of the following compound types:

- *Objects*: an unordered sequence of (*key*, *value*) pairs, where the key always has string type and is unique within the sequence.
- *List*: an ordered list of items.

Where lists are defined the name '*entry*' is used to define the format of each entry, but since lists contain unnamed entries this name would not appear in any concrete syntax.

Each service description is defined as a named list of entries that logically contains the service description information. When and how this information is delivered and how it is updated is out of scope of this clause.

K.3.2 Service Latency

Table K.1 defines the service description parameters for the service latency.

Table K.1 — Service Latency

Parameter			Type	Description
ReferenceID			Integer	Defines the ID of the Producer Reference Time element in the MPD against which the latency is measured, if more than one is present.
TargetLatency			Integer	<p>The service provider's preferred presentation latency in milliseconds compared to the producer reference time. Indicates a content provider's desire for the content to be presented as close to the indicated latency as is possible given the player's capabilities and observations.</p> <p>NOTE This attribute can express a latency that is only achievable by low-latency players under favourable network conditions.</p>
MaximumLatency			Integer	The service provider's indication about the maximum presentation latency in milliseconds. Indicates a content provider's desire for the content not to be presented if the latency exceeds the maximum latency.
MinimumLatency			Integer	The service provider's indication about minimum presentation latency in milliseconds for example to avoid inconsistencies with second screen applications, overlays, etc.
	QualityType		URI	Defines the interpretation of the value of Quality. If not present, the quality is considered in linear scale.
	<i>Entry</i>		Object	A list of latency and quality pairs.
		Latency	Integer	A latency value for the service in milliseconds.
		Quality	Integer	The quality of the service at the above latency with 0 being the lowest and 100 being the highest.

K.3.3 Playback Rate

Table K.2 defines the service description parameters for non-nominal playback rate. The keys in Table K.2 shall be used to refer to the non-nominal playback rate as defined in Table K.2.

Table K.2 — Playback Rate

Key	Type	Description
MaxPlaybackRate	Real	The maximum playback rate that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed.
MinPlaybackRate	Real	The minimum playback rate that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed.

K.3.4 Operating Quality

Table K.3 defines the service description parameters for operational quality. The keys in Table K.3 shall be used to refer to the operating quality as defined in Table K.3.

Table K.3 — Operating Quality

Key	Type	Description
MediaType	string	Defines the media type for which these quality parameters apply. They type can be <ul style="list-style-type: none"> — video: applies to video — audio: applies to audio — any: applies to any media type
MinQualityRanking	Integer	Minimum quality ranking value desired through this presentation for the above MediaType. This is the smallest desired value for the @qualityRanking attribute for the selected Representations to be played by the DASH Client during regular playback.
MaxQualityRanking	Integer	Maximum quality ranking value desired through this presentation for the above MediaType. This is the largest desired value for the @qualityRanking attribute for the Representations to be played by the DASH Client during regular playback.
TargetQualityRanking	Integer	Quality ranking value desired through this presentation for the above MediaType. This is the target value for the @qualityRanking attribute for the Representations to be played by the DASH Client during regular playback.
QualityRankingType	URL	Defines the interpretation of the value of @qualityRanking attribute. If not present, the quality is considered in linear scale.
MaxQualityDifference	Integer	Maximum quality difference value recommended by the content author for the presentation for the above MediaType. This is the recommended maximum difference between @qualityRanking attributes for Representations being played concurrently. This is typically applicable for Representations that are picked from an Adaptation Sets within one Preselection.

K.3.5 Operating Bandwidth

Table K.4 defines the service description parameters for operating bandwidth. The keys in Table K.4 shall be used to refer to the operating bandwidth as defined in Table K.4.

Table K.4 — Operating Bandwidth

Key	Type	Description
MediaType	string	Defines the media type for which the bandwidth parameters apply. They type can be <ul style="list-style-type: none"> — video: applies to video — audio: applies to audio — any: applies to any media type individually — all: applies to aggregation of all media types
MinBandwidth	Integer	Minimum bandwidth value desired through this presentation for the above MediaType. This is the smallest desired aggregated value for the @bandwidth attribute for all Representations of this media type to be played concurrently by the DASH Client during regular playback.
MaxBandwidth	Integer	Maximum bandwidth value desired through this presentation for the above MediaType. This is the largest desired aggregated value for the @bandwidth attribute for all Representations of this media type to be played concurrently by the DASH Client during regular playback.
TargetBandwidth	Integer	Target bandwidth value desired through this presentation for the above MediaType. This is the largest desired aggregated value for the @bandwidth attribute for all Representations of this media type to be played concurrently by the DASH Client during regular playback.

K.4 MPD Carriage of DASH Service Description

K.4.1 Description

This clause provides the mapping of the service description in to the MPD. The service description may be scoped to a specific client or to an operational mode of the client. This can be expressed by using a Scope descriptor that is defined by the application.

It is the service provider's responsibility to ensure that the service description is not contradicting in itself and can be used by clients, for example minimum, target and maximum latency should be in non-decreasing order. Service descriptions with inconsistent or unachievable settings are expected to be ignored by the client.

The semantics of the attributes and elements within a **ServiceDescription** element are provided in subclause K.4.2.1, Table K.5. The XML syntax of the **ServiceDescription** element is provided in subclause K.4.3.

K.4.2 Semantics

K.4.2.1 General

Table K.5 — Semantics of **ServiceDescription element**

Element or Attribute Name	Use	Description
ServiceDescription		Service Description
@id	M	specifies a unique identifier for this Service Description. The attribute shall be a unique unsigned integer value amongst ServiceDescription elements in the scope of the MPD.
Scope	0 ... N	specifies the scope of the Service Description. If present, this Service Description only targets DASH Clients identified by this Scope descriptor. DASH Clients not in scope, i.e. not recognizing any of the scope descriptor elements, are expected to ignore this service description. If no Scope element is present the Service Description applies to all clients.
Latency	0 ... N	specifies the latency targets for the service. The details are provided in subclause K.4.2.2, Table K.6.
PlaybackRate	0 ... N	specifies the playback rate targets for the service. The details are provided in subclause K.4.2.3, Table K.7.
OperatingQuality	0 ... N	specifies the operating quality targets for the service. The details are provided in subclause K.4.2.4, Table K.8.
OperatingBandwidth	0 ... N	specifies the operating quality targets for the service. The details are provided in subclause K.4.2.5, Table K.9.
Legend: For attributes: M=mandatory, O=optional, OD=ptional with default value, CM= conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using xlink:href. If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @, List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

K.4.2.2 Latency

Table K.6 — Semantics of **Latency element**

Element or Attribute Name	Use	Description
Latency		see subclause K.3.2.
@referenceId	CM	See <i>referenceID</i> in Table K.1
@target	O	See <i>TargetLatency</i> in Table K.1
@max	O	See <i>MaximumLatency</i> in Table K.1
@min	O	See <i>MinimumLatency</i> in Table K.1

Element or Attribute Name	Use	Description
QualityLatency	0 ... N	A comma-separated list of white space separated latency and quality pair values according to Table K.1.
@type	0	Defines the <i>QualityType</i> according to Table K.1
Legend: For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using <code>xlink:href</code> . If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @, List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

K.4.2.3 Playback Rate

Table K.7 — Semantics of **PlaybackRate** element

Element or Attribute Name	Use	Description
PlaybackRate		see subclause K.3.2.
@max	0	See <i>maxPlaybackRate</i> in Table K.2
@min	0	See <i>minPlaybackRate</i> in Table K.2
Legend: For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using <code>xlink:href</code> . If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @, List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

K.4.2.4 Operating Quality

Table K.8 — Semantics of **OperatingQuality** element

Element or Attribute Name	Use	Description
OperatingQuality		see subclause K.3.2.
@mediaType	OD Default: 'any'	See <i>MediaType</i> in Table K.3
@min	0	See <i>MinQualityRanking</i> in Table K.3
@max	0	See <i>MaxQualityRanking</i> in Table K.3
@target	0	See <i>TargetQualityRanking</i> in Table K.3
@type	0	See <i>QualityRankingType</i> in Table K.3
@maxDifference	0	See <i>MaxQualityDifference</i> in Table K.3
Legend: For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using <code>xlink:href</code> . If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @, List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

K.4.2.5 Operating Bandwidth

Table K.9 — Semantics of OperatingBandwidth element

Element or Attribute Name	Use	Description
OperatingBandwidth		see subclause K.3.2.
@mediaType	OD Default: 'all'	See MediaType in Table K.4
@min	0	See MinBandwidth in Table K.4
@max	0	See MaxBandwidth in Table K.4
@target	0	See TargetBandwidth in Table K.4
Legend: For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed. For elements: <minOccurs>...<maxOccurs> (N=unbounded) The conditions only hold without using xlink:href. If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are bold ; attributes are non-bold and preceded with an @, List of elements and attributes is in <i>italics bold</i> referring to those taken from the Base type that has been extended by this type.		

K.4.3 Syntax

K.4.3.1 General

```

<xs:complexType name="ServiceDescriptionType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Service Description
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Scope" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Latency" type="LatencyType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PlaybackRate" type="PlaybackRateType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OperatingQuality" type="OperatingQualityType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="OperatingBandwidth" type="OperatingBandwidthType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:unsignedInt"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

K.4.3.2 Latency

```

<xs:complexType name="LatencyType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Service Description Latency
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="QualityLatency" type="UIntPairsWithIDType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="referenceId" type="xs:unsignedInt"/>
  <xs:attribute name="target" type="xs:unsignedInt"/>
  <xs:attribute name="max" type="xs:unsignedInt"/>
  <xs:attribute name="min" type="xs:unsignedInt"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="UIntPairsWithIDType">

```

```

<xs:annotation>
  <xs:documentation xml:lang="en">
    UInt Pairs With ID
  </xs:documentation>
</xs:annotation>
<xs:simpleContent>
  <xs:extension base="UIntVectorType">
    <xs:attribute name="type" type="xs:anyURI"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:extension>
</xs:simpleContent>
</xs:complexType>

```

K.4.3.3 Playback Rate

```

<xs:complexType name="PlaybackRateType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Service Description Playback Rate
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="max" type="xs:double"/>
  <xs:attribute name="min" type="xs:double"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

K.4.3.4 Operating Quality

```

<xs:complexType name="OperatingQualityType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Service Description Operating Quality
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="mediaType" default="any">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="video"/>
        <xs:enumeration value="audio"/>
        <xs:enumeration value="any"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="min" type="xs:unsignedInt"/>
  <xs:attribute name="max" type="xs:unsignedInt"/>
  <xs:attribute name="target" type="xs:unsignedInt"/>
  <xs:attribute name="type" type="xs:anyURI"/>
  <xs:attribute name="maxDifference" type="xs:unsignedInt"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

```

K.4.3.5 Operating Bandwidth

```

<xs:complexType name="OperatingBandwidthType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Service Description Operating Bandwidth
    </xs:documentation>
  </xs:annotation>
  <xs:attribute name="mediaType" default="all">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="video"/>
        <xs:enumeration value="audio"/>
        <xs:enumeration value="any"/>
        <xs:enumeration value="all"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

```

```
</xs:attribute>
<xs:attribute name="min" type="xs:unsignedInt"/>
<xs:attribute name="max" type="xs:unsignedInt"/>
<xs:attribute name="target" type="xs:unsignedInt"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

K.5 Example Client Usage

This clause provides some information on how a DASH Client may use the information in the service description.

If the DASH Client is informed of a target latency by the `TargetLatency` parameter and it identifies that the Media Presentation carries the producer reference time with considered type, the DASH Client attempts to present the sample with a specific media time at a wall-clock time that is the sum of provided wall-clock time and the target latency.

The DASH Client may monitor the playback continuously and may adapt the playback rate in the boundaries permitted by the playback rate parameters to catchup to the latency.

The DASH Client may adjust its playback to maintain quality and bandwidth used between the minimum and maximum quality and minimum and maximum bandwidth parameters, if provided.

Bibliography

- [1] ISO/IEC 23001-7, *Information technology — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files*
- [2] 3GPP TS 26.234, *Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs*
- [3] 3GPP TS 26.244, *Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)*
- [4] IETF RFC 1952, *GZIP file format specification version 4.3*
- [5] IETF RFC 2231, *MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations*
- [6] IETF RFC 2405, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*
- [7] IETF RFC 2818, *HTTP Over TLS*
- [8] IETF RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*
- [9] ISO/IEC 13818-2, *Information technology — Generic coding of moving pictures and associated audio information: Video*
- [10] "XML Encryption Syntax and Processing Version 1.1", W3C Recommendation 11 April 2013, <http://www.w3.org/TR/xmlenc-core/>
- [11] "XML Signature Syntax and Processing (Second Edition)", W3C Recommendation 10 June 2008, <http://www.w3.org/TR/xmldsig-core/>
- [12] IEEE 1003.1-2008, *IEEE Standard for Information Technology – Portable Operating System Interface (POSIX), Base Specifications, Issue 7*
- [13] Open Authentication Technical Committee (OATC) Online Multimedia Authorization Protocol (OMAP)
- [14] Open Standard for Authorization (OAuth) 2.0
- [15] OASIS Security Assertion Markup Language (SAML)
- [16] 3GPP Generic Authentication Architecture (GAA)
- [17] 3GPP Generic Bootstrapping Architecture (GBA)
- [18] VIDEO AD SERVING TEMPLATE I.A.B. (VAST) v3.0, July 2012, https://www.iab.com/wp-content/uploads/2015/06/VASTv3_0.pdf
- [19] ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*
- [20] ISO/IEC TR 23009-3, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 3: Implementation guidelines*

- [21] IETF RFC 6265, *HTTP State Management Mechanism*
- [22] IETF RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*
- [23] IETF RFC 7232, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*
- [24] IETF RFC 7234, *Hypertext Transfer Protocol (HTTP/1.1): Caching*
- [25] IETF RFC 7235, *Hypertext Transfer Protocol (HTTP/1.1): Authentication*
- [26] DASH Interoperability Forum, *Content Protection IOP*,
https://dashif.org/identifiers/content_protection/
- [27] W3C HTML 5, aka WhatWG HTML Living Standard, <https://html.spec.whatwg.org/multipage/>
- [28] Digital Content Protection LLC, *HDCP Interface Independent Adaptation Specification Revision 2.3*
- [29] IETF RFC 1738, *Uniform Resource Locators*
- [30] United States Code Title 47 CFR 79.103, Electronic Code of Federal Regulations: Closed caption decoder requirements for apparatus, https://www.ecfr.gov/cgi-bin/text-idx?node=se47.4.79_1103
- [31] SMPTE ST 2067-2, SMPTE Standard - Interoperable Master Format - Core Constraints

