

INTERNATIONAL  
STANDARD

ISO/IEC  
23009-1

Third edition  
2019-08

---

---

---

**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*



Reference number  
ISO/IEC 23009-1:2019(E)

© ISO/IEC 2019



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2019

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  
Fax: +41 22 749 09 47  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

	Page
<b>Foreword</b>	<b>vii</b>
<b>Introduction</b>	<b>ix</b>
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>1</b>
<b>3 Terms, definitions, symbols and abbreviated terms</b>	<b>2</b>
3.1 Terms and definitions	2
3.2 Symbols and abbreviated terms	6
3.3 Conventions	8
<b>4 Overview</b>	<b>8</b>
4.1 System description	8
4.2 DASH Client model	9
4.3 DASH data model overview	10
4.4 Protocols	13
4.5 Media stream and Representation properties	14
4.5.1 Switching and Random Access Support	14
4.5.2 Media stream access points	14
4.5.3 Non-overlapping Segments and Subsegments	15
4.5.4 Bitstream concatenation	16
4.6 Brands	16
4.7 Schemes	16
<b>5 Media Presentation</b>	<b>19</b>
5.1 General	19
5.2 Media Presentation Description	19
5.2.1 General	19
5.2.2 Schema	20
5.2.3 Elements and Attributes added in revisions and amendments	21
5.3 Hierarchical data model	22
5.3.1 General	22
5.3.2 Period	27
5.3.3 Adaptation Sets	31
5.3.4 Media content component	42
5.3.5 Representation	44
5.3.6 Sub-Representation	52
5.3.7 Common attributes and elements	53
5.3.8 Subsets	59
5.3.9 Segments and Segment information	60
5.3.10 Label and Group Label	78
5.3.11 Preselection	79
5.4 Media Presentation Description updates	82
5.4.1 General	82
5.4.2 MPD Reset	83
5.5 MPD assembly	83
5.5.1 General	83
5.5.2 Syntax and semantics	83
5.5.3 Processing	84
5.6 Base URL Processing	85
5.6.1 Overview	85
5.6.2 Semantics	85
5.6.3 XML syntax	86
5.6.4 Reference resolution	87
5.6.5 Alternative base URLs	87
5.7 Program information	87

5.7.1	Overview .....	87
5.7.2	Semantics .....	87
5.7.3	XML syntax .....	88
5.8	Descriptors .....	88
5.8.1	General .....	88
5.8.2	Semantics of generic descriptor .....	89
5.8.3	XML syntax of generic descriptor .....	90
5.8.4	Specific descriptors .....	90
5.8.5	Specific scheme definitions .....	93
5.9	DASH metrics descriptor .....	101
5.9.1	Overview .....	101
5.9.2	Semantics .....	101
5.9.3	XML syntax .....	102
5.9.4	Metric reporting .....	103
5.10	Events .....	103
5.10.1	Overview .....	103
5.10.2	MPD Events .....	103
5.10.3	Inband Event Signalling .....	106
5.10.4	DASH-specific events .....	109
5.11	MPD Chaining .....	112
5.11.1	General .....	112
5.11.2	Regular Chaining .....	112
5.11.3	Fallback Chaining .....	113
6	<b>Segment formats .....</b>	<b>113</b>
6.1	General .....	113
6.2	Segment types .....	114
6.2.1	General .....	114
6.2.2	Initialization Segment .....	114
6.2.3	Media Segment .....	114
6.2.4	Index Segment .....	116
6.2.5	Bitstream Switching Segment .....	116
6.3	Segment formats for ISO base media file format .....	116
6.3.1	General .....	116
6.3.2	Preliminaries: Refinements of generic concepts .....	116
6.3.3	Initialization Segment format .....	117
6.3.4	Media Segment types .....	117
6.3.5	Self-Initializing Media Segment formats .....	119
6.4	Segment formats for MPEG-2 transport streams .....	119
6.4.1	General .....	119
6.4.2	Preliminaries: Refinements of generic concepts .....	120
6.4.3	Initialization Segment types and formats .....	121
6.4.4	Media Segment types and formats .....	122
6.4.5	Bitstream Switching Segment .....	122
6.4.6	Index Segment .....	123
6.4.7	Boxes used with MPEG-2 TS Index Segments .....	125
7	<b>Combined semantics of MPD and Segment formats .....</b>	<b>125</b>
7.1	Overview .....	125
7.2	General .....	126
7.2.1	Media Presentation timeline .....	126
7.2.2	Segment Index .....	127
7.2.3	Segment alignment .....	127
7.2.4	Subsegment alignment .....	127
7.3	Media Presentation based on the ISO base media file format .....	127
7.3.1	General .....	127
7.3.2	Media presentation timeline .....	128
7.3.3	Authoring Rules for specific MPD attributes .....	128
7.3.4	Sub-Representations .....	129

7.4	7.3.5 Segment Timeline without Segment Index .....	129
	Media Presentation based on MPEG-2 TS .....	129
	7.4.1 General .....	129
	7.4.2 Media presentation timeline .....	130
	7.4.3 Authoring rules for specific MPD attributes .....	130
	7.4.4 Sub-Representations .....	131
<b>8</b>	<b>Profiles .....</b>	<b>131</b>
8.1	Definition .....	131
8.2	Full profile .....	133
	8.2.1 General .....	133
	8.2.2 Media Presentation Description constraints .....	133
	8.2.3 Segment format constraints .....	133
8.3	ISO Base media file format On Demand profile .....	133
	8.3.1 General .....	133
	8.3.2 Media Presentation Description constraints .....	133
	8.3.3 Segment format constraints .....	134
8.4	ISO Base media file format live profile .....	134
	8.4.1 General .....	134
	8.4.2 Media Presentation Description constraints .....	135
	8.4.3 Segment format constraints .....	135
8.5	ISO Base media file format main profile .....	136
	8.5.1 General .....	136
	8.5.2 Media Presentation Description constraints .....	136
	8.5.3 Segment format constraints .....	136
8.6	MPEG-2 TS main profile .....	137
	8.6.1 General .....	137
	8.6.2 Media Presentation Description constraints .....	137
	8.6.3 Segment format constraints .....	137
	8.6.4 Comments and recommendations .....	137
8.7	MPEG-2 TS simple profile .....	138
	8.7.1 General .....	138
	8.7.2 Media Presentation Description constraints .....	138
	8.7.3 Segment format constraints .....	138
	8.7.4 Recommendations .....	138
8.8	ISO Base media file format extended live profile .....	139
	8.8.1 General .....	139
	8.8.2 Media Presentation Description constraints .....	139
	8.8.3 Segment format constraints .....	140
	8.8.4 Inband Events .....	140
8.9	ISO Base media file format extended On Demand profile .....	140
	8.9.1 General .....	140
	8.9.2 Media Presentation Description constraints .....	140
	8.9.3 Segment format constraints .....	141
8.10	ISO Base media file format common profile .....	142
	8.10.1 General .....	142
	8.10.2 Media Presentation Description constraints .....	142
	8.10.3 Segment format constraints .....	142
8.11	ISO Base media file format broadcast TV profile .....	142
	8.11.1 General .....	142
	8.11.2 Media Presentation Description constraints .....	143
	8.11.3 Segment format constraints .....	144
	8.11.4 MPD Updates and Inband Event Streams .....	144
<b>Annex A</b> (informative) <b>Example DASH Client behaviour .....</b>	<b>146</b>	
<b>Annex B</b> (normative) <b>MPD schema .....</b>	<b>154</b>	
<b>Annex C</b> (normative) <b>MIME type registration for MPD .....</b>	<b>155</b>	
<b>Annex D</b> (normative) <b>DASH Metrics .....</b>	<b>159</b>	

<b>Annex E (normative) Byte range requests with regular HTTP GET methods</b> .....	<b>165</b>
<b>Annex F (informative) Guidelines for extending DASH with other delivery formats</b> .....	<b>167</b>
<b>Annex G (informative) MPD Examples and MPD Usage</b> .....	<b>168</b>
<b>Annex H (normative) Spatial Relationship Description</b> .....	<b>197</b>
<b>Annex I (normative) Flexible Insertion of URL Parameters</b> .....	<b>208</b>
<b>Annex J (informative) Open GOP resolution change</b> .....	<b>223</b>
<b>Bibliography</b> .....	<b>224</b>

## 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](http://www.iso.org/directives)).

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](http://www.iso.org/patents)) or the IEC list of patent declarations received (see <http://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](http://www.iso.org/iso/foreword.html).

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 third edition cancels and replaces the second edition (ISO/IEC 23009-1:2014), which has been technically revised. It also incorporates the Technical Corrigenda ISO/IEC 23009-1:2014/Cor.1:2015 and ISO/IEC 23009-1:2014/Cor.2:2015 and the Amendments ISO 23009-1:2014/Amd.1:2015, ISO 23009-1:2014/Amd.2:2015 and ISO 23009-1:2014/Amd.3:2016. The main changes compared to the previous edition are as follows:

- signaling a server timing source was added;
- the ability to label different structures with human readable identifiers was included;
- signaling of properly prepared media at Period boundaries for continuous playback across Periods was added;
- the concept of Preselections was added in order to combine different Adaptation Sets into a single decoding and user experience;
- the ability to offer segments which are not starting with SAP types 1 or 2, but still provide consistent random access and switching points on MPD level, including a new profile suitable for Broadcast TV, was added;
- MPD chaining to enable sequencing MPDs, for example to support pre-roll content to live services, was added;
- the ability to separate a long on-Demand Asset in multiple Periods, possibly including Periods with other content such as ads, was added;
- spatial Relationship Description was added to signal the logical combination of multiple Adaptation Sets into a single experience;

- the ability to add URL Parameters to HTTP requests in flexible and interoperable manner was included.

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

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](http://www.iso.org/members.html).

## Introduction

Dynamic adaptive streaming over HTTP (DASH) is intended to support a media-streaming model for delivery of media content in which control lies exclusively 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 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:

Electronics & Telecommunications Research Institute  
161, Gajeong-dong, Yuseong-gu, Daejeon, Korea  
Intellectual Property Management Team, ETRI

EMBLAZE LTD.  
9 1-lamenofim Street  
Herzeliya Pituach 46725, Israel P.O.Box 2216

Sony Corporation  
1-7-1 Knonan, Minato-Kyu, Tokyo, 108-0075

Intel Corporation  
2200 Mission College Blvd., MS: RNB-4-150 Santa Clara, CA 95054

Koninklijke KPN N.V.  
Maanplein 55, 2516 CK, The Hague, The Netherlands

Sharp Corporation  
2613-1 Ichinomoto-cho, Tenri-shi, Nara Prefecture 632-8567, Japan

Huawei Technologies Co., Ltd

Administration Building, Huawei Technologies Co., Ltd Bantian Longgang District  
Shenzhen 518129, China

Samsung Electronics Co., Ltd  
Maetan dong 129, Samsung-ro, Yeongtong-gu, Suwon-si,  
Gyeonggi-do 443-742, Korea Republic

Fraunhofer-Gesellschaft  
Hansastr. 27c  
80686 München

Google Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043

Nokia Technologies Oy  
Joensuunkatu 7E, FIN-24100 Salo, Finland

Qualcomm Incorporated  
5775 Morehouse Drive  
San Diego, CA 92121

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. 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:—<sup>1)</sup>, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 23001-8, *Information technology — MPEG systems technologies — Part 8: Coding-independent code points*

IETF RFC 1738, *Uniform Resource Locators*

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

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

IETF RFC 3986:2005, *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 6381:2011, *The ‘Codecs’ and ‘Profiles’ Parameters for “Bucket” Media Types*

IETF RFC 6838:2013, *Media Type Specifications and Registration Procedures*

IETF RFC 7231:2014, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

1) Under preparation. Stage at the time of publication: ISO/IEC/DIS 14496-12:2017. This document is technically identical to ISO/IEC 15944-12 (withdrawn).

IETF RFC 7233:2014, *Hypertext Transfer Protocol (HTTP/1.1): Range Requests*

IETF RFC 8141:2017, *URN Syntax*

HTML 4.01 Specification, W3C Recommendation, 24 December 1999

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 following terms and definitions apply.

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

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://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* (3.1.39) that is accessible at its assigned *HTTP-URL* (3.1.18) 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* (3.1.39) that if present contains essential data to switch to the *Representation* (3.1.38) it is assigned to

#### 3.1.8

##### **bundle**

set of media components which can be consumed jointly by a single decoder instance

**3.1.9****complementary Representation**

*Representation* ([3.1.38](#)) which complements at least one *dependent Representation* ([3.1.13](#))

**3.1.10****continuous media**

media with an inherent notion of time

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

**3.1.11****DASH metric**

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

**3.1.12****data URL**

URL with a fixed scheme “data”

**3.1.13****dependent Representation**

*Representation* ([3.1.38](#)) for which *Segments* ([3.1.39](#)) from its *complementary Representations* ([3.1.9](#)) are necessary for presentation and/or decoding of the contained *media content components* ([3.1.22](#))

**3.1.14****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* ([3.1.49](#)) for a *media stream* ([3.1.29](#))

**3.1.15****event**

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

**3.1.16****event stream**

sequence of related *events* ([3.1.15](#))

**3.1.17****group**

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

**3.1.18****HTTP-URL**

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

**3.1.19****Index Segment**

*Segment* ([3.1.39](#)) that primarily contains indexing information for *Media Segments* ([3.1.28](#))

**3.1.20****Initialization Segment**

*Segment* ([3.1.39](#)) containing metadata that is necessary to present the *media streams* ([3.1.29](#)) encapsulated in *Media Segments* ([3.1.28](#))

**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](#))

EXAMPLE      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* ([3.1.39](#)) that complies with media format in use and enables playback when combined with zero or more preceding Segments and an *Initialization Segment* ([3.1.20](#)) (if any)

**3.1.29**

**media stream**

encoded version of a *media content component* ([3.1.22](#))

**3.1.30**

**Media Subsegment**

*Subsegment* ([3.1.49](#)) that only contains media data but no *Segment Index* ([3.1.43](#))

**3.1.31**

**message**

part of an *event* ([3.1.15](#)) 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* ([3.1.25](#)), where a contiguous sequence of all Periods constitutes the Media Presentation

**3.1.35**

**Preselection**

subset of media component in a *bundle* ([3.1.8](#)) that are expected 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* (3.1.18) 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****Segment**

unit of data associated with an *HTTP-URL* (3.1.18) and optionally a byte range that are specified by an *MPD* (3.1.26), or with a *data URL* (3.1.12)

**3.1.40****Segment availability start time**

latest time instant in *wall-clock time* (3.1.51) at which a *Segment* (3.1.39) becomes an *available Segment* (3.1.6)

**3.1.41****adjusted Segment availability start time**

time instant in *wall-clock time* (3.1.51) at which a *Segment* (3.1.39) becomes an *available Segment* (3.1.6)

**3.1.42****Segment availability end time**

time instant in *wall-clock time* (3.1.51) at which a *Segment* (3.1.39) ceases to be an *available Segment* (3.1.6)

**3.1.43****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)

**3.1.44****Segment Sequence**

sequence of *Segments* (3.1.39) that are sharing a common number address

**3.1.45****Spatial Object**

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

**3.1.46****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* (3.1.20), if any]

**3.1.47****sub-asset**

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

**3.1.48****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)

**3.1.49****Subsegment**

unit within *Media Segments* ([3.1.28](#)) that is indexed by a *Segment Index* ([3.1.43](#))

**3.1.50****valid Segment URL**

*HTTP-URL* ([3.1.18](#)) that is promised to reference a *Segment* ([3.1.39](#)) during its Segment availability period

**3.1.51****wall-clock time**

time as stated by UTC

## 3.2 Symbols and abbreviated terms

For the purposes of this document, the following symbols and abbreviated terms apply.

ABNF	Augmented Backus-Naur Form
AVC	advanced video coding
CAT	conditional access table
CDN	content delivery network
DASH	dynamic adaptive streaming over HTTP
DM	DASH Metrics
DRM	digital rights management
ECM	entitlement control message
EMM	entitlement management message
EPT	earliest presentation time
GDR	gradual decoder refresh
GOP	group of pictures
GPS	global positioning system
HEVC	high efficiency video coding
HTML	hypertext markup language
HTTP	hypertext transfer protocol
IAB	interactive advertising bureau
IDR	instantaneous decoding refresh
IP	internet protocol
ISOBMFF	ISO base media file format
MBT	minimum buffer time
MPD	Media Presentation Description
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
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. @everyImportantAttribute. 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.

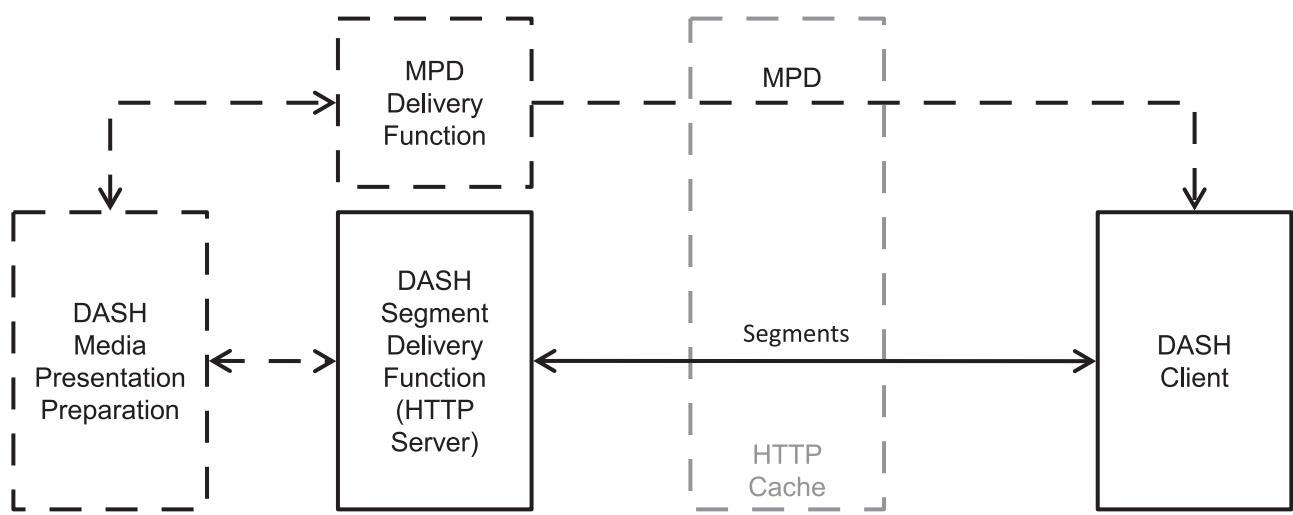
This document primarily defines two formats:

- 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 client 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.

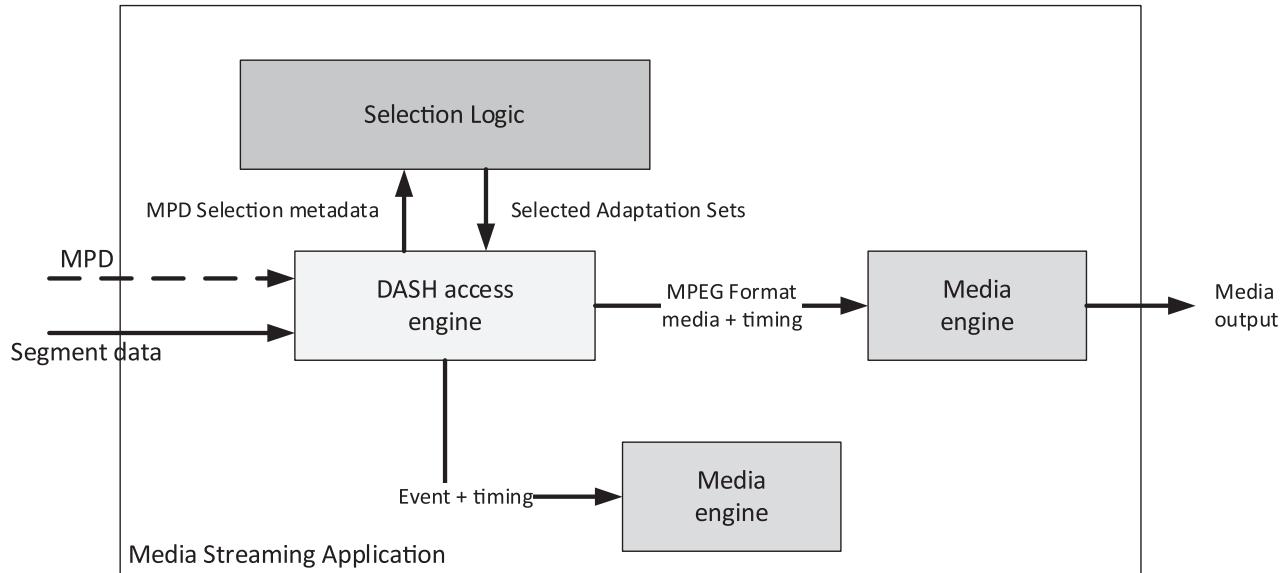


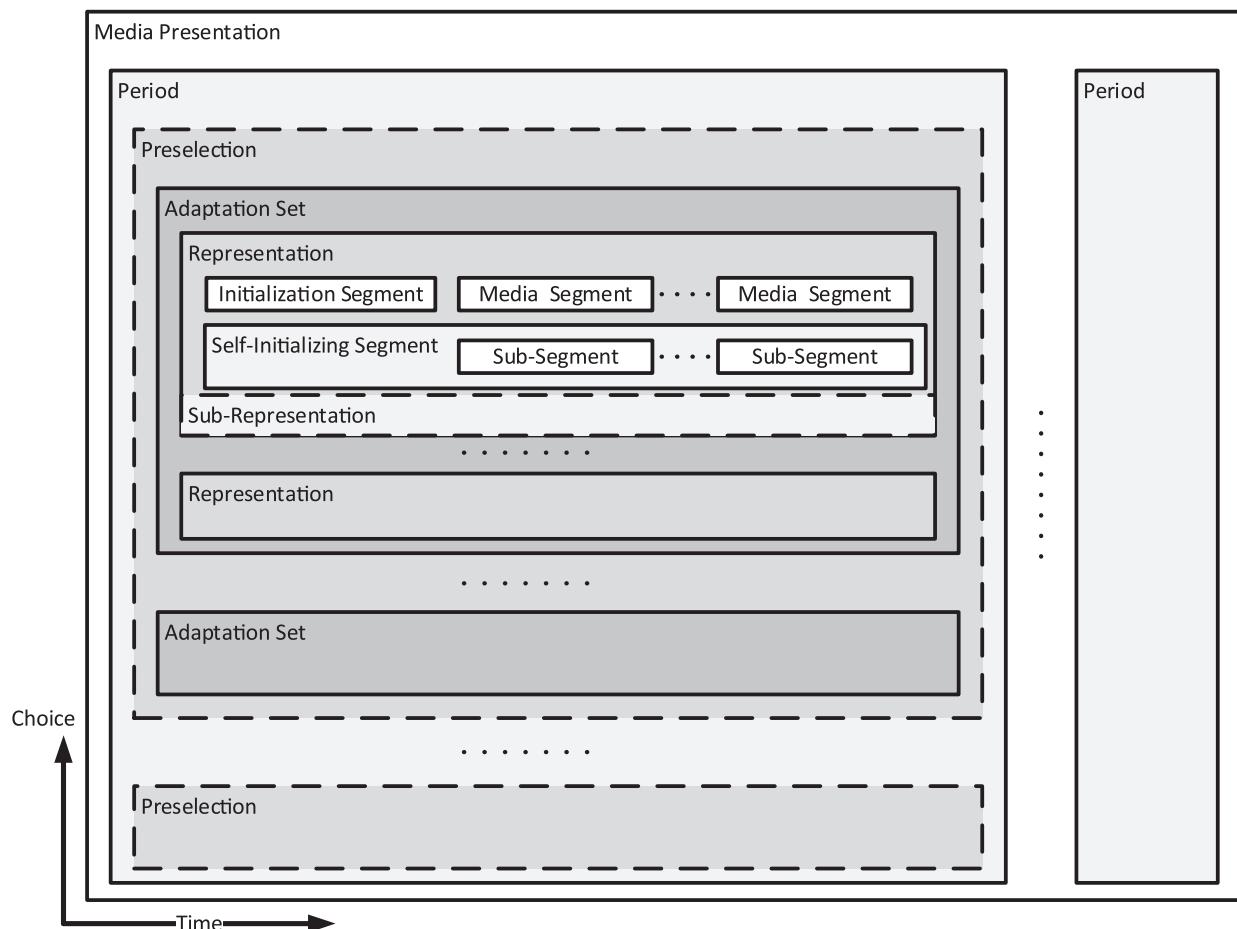
Figure 2 — DASH Client model

#### 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 describing 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 horizontal 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 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. Clients may ignore Representations that rely on codecs or other rendering technologies they do not support or that are otherwise unsuitable.

Within a Representation, the content may be divided in time into **Segments** (see subclause 5.3.9 and Clause 6) 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. Media may also be organized by a single self-initializing Segment which contains then both initialization information as well as media data.

**NOTE** This is not strictly true, since 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 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 whole 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 whole 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 download of '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 for providing good user experience even in case 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.
- Bitstream 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:—, Annex I. ISO/IEC 14496-12:—, Annex I.3 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  $T_{PTP}$ , are identified and defined in ISO/IEC 14496-12:—, Annex I.2.

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 Bitstream concatenation

A sequence of Segments (respectively Subsegments) is a “conforming Segment (respectively Subsegment) sequence” 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	<a href="#">5.10.3.3</a>	Event message box.
msdh	<a href="#">6.3.4.2</a>	Media Segment conforming to the general format type for ISO base media file format.
msix	<a href="#">6.3.4.3</a>	Media Segment conforming to the Indexed Media Segment format type for ISO base media file format.
sims	<a href="#">6.3.4.4</a>	Media Segment conforming to the Sub-Indexed Media Segment format type for ISO base media file format.
dsms	<a href="#">6.3.5.1</a>	Media Segment conforming to the DASH Self-Initializing Media Segment format type for ISO base media file format.
dash	<a href="#">6.3.5.2</a>	ISO base media file format file specifically designed for DASH including movie fragments and Segment Index.
sisx	<a href="#">6.4.6.2</a>	Single Index Segment used to index MPEG-2 TS based Media Segments.
rissx	<a href="#">6.4.6.3</a>	Representation Index Segment used to index MPEG-2 TS based Media Segments.
ssss	<a href="#">6.4.6.4</a>	Subsegment Index Segment used to index MPEG-2 TS based Media Segments.
lmsg	<a href="#">7.3.1</a>	Last Media Segment indicator for ISO base media file format.
dums	<a href="#">8.11.3</a>	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**

<b>Scheme identifier</b>	<b>Clause in this document</b>	<b>Informative description</b>
urn:mpeg:dash:schema:mpd:2011	<a href="#">Annex B</a>	The namespace of the XML schema for the MPD.
urn:mpeg:dash:period-continuity:2015	<a href="#">5.3.2.4</a>	Period continuity signaling.
urn:mpeg:dash:period-connectivity:2015	<a href="#">5.3.2.4</a>	Period connectivity signaling.
urn:mpeg:dash:adaptation-set-switching:2016	<a href="#">5.3.3.5</a>	Scheme Identifier for switching across Adaptation Sets.
urn:mpeg:dash:preselection:2016	<a href="#">5.3.11.2</a>	Preselection descriptor.
urn:mpeg:dash:reset:2016	<a href="#">5.4.2</a>	MPD reset indicator.
urn:mpeg:dash:resolve-to-zero:2013	<a href="#">5.5.3</a>	xlink resolution to zero element.
urn:mpeg:dash:mp4protection:2011	<a href="#">5.8.5.2</a>	Protection schemes identified by the Scheme Type within the Scheme Type Box of the Protection Scheme Information Box of ISO/IEC14496-12.
urn:mpeg:dash:13818:1:CA_descriptor:2011	<a href="#">5.8.5.2</a>	Conditional Access System used for ISO/IEC 13818-1 (MPEG-2 Transport Stream).
urn:mpeg:dash:14496:10:frame_packing_arrangement_type:2011	<a href="#">5.8.5.3</a>	Frame-packing arrangement.
urn:mpeg:dash:13818:1:stereo_video_format_type:2011	<a href="#">5.8.5.3</a>	Frame-packing arrangement.
urn:mpeg:dash:23003:3:audio_channel_configuration:2011	<a href="#">5.8.5.4</a>	Channel configuration. Legacy format for backward-compatibility, it is recommended to use the signalling as defined in ISO/IEC 23001-8 instead.
urn:mpeg:dash:outputChannelPositionList:2012	<a href="#">5.8.5.4</a>	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 the signalling as defined in ISO/IEC 23001-8 instead.
urn:mpeg:dash:role:2011	<a href="#">5.8.5.5</a>	DASH role scheme.
urn:mpeg:dash:stereoid:2011	<a href="#">5.8.5.6</a>	Scheme for multiple views media content description.
urn:mpeg:dash:utc:ntp:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for NTP servers.
urn:mpeg:dash:utc:sntp:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for SNTP servers.
urn:mpeg:dash:utc:http-head:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for HTTP date headers.
urn:mpeg:dash:utc:http-xsdate:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for HTTP server with xsdate format.
urn:mpeg:dash:utc:http-iso:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for HTTP server with ISO timing format.
urn:mpeg:dash:utc:http-ntp:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for HTTP server with NTP format.
urn:mpeg:dash:utc:direct:2014	<a href="#">5.8.5.7</a>	UTC Timing scheme for direct inclusion of time.
urn:mpeg:dash:audio-receiver-mix:2014	<a href="#">5.8.5.8</a>	Scheme identifier for receiver mix.
urn:mpeg:dash:mpd-as-linking:2015	<a href="#">5.8.5.9</a>	MPD Adaptation Set Linking scheme.

**Table 2** (*continued*)

<b>Scheme identifier</b>	<b>Clause in this document</b>	<b>Informative description</b>
urn:mpeg:dash:sai:2015	<a href="#">5.8.5.10</a>	Sub-Asset Scheme Identifier.
urn:mpeg:dash:client-authentication:2015	<a href="#">5.8.5.11</a>	Client Authentication scheme.
urn:mpeg:dash:content-authorization:2015	<a href="#">5.8.5.11</a>	Content Access Authorization scheme.
urn:mpeg:dash:audio-interactivity:2016	<a href="#">5.8.5.12</a>	Scheme to indicate content interactivity.
urn:mpeg:dash:event:2012	<a href="#">5.10.4</a>	DASH event signalling scheme.
urn:mpeg:dash:event:callback:2015	<a href="#">5.10.4.5</a>	DASH call back event.
urn:mpeg:dash:event:ttfn:2016	<a href="#">5.10.4.6</a>	Presentation Termination Event.
urn:mpeg:dash:chaining:2016	<a href="#">5.11.1</a>	MPD chaining descriptor.
urn:mpeg:dash:fallback:2016	<a href="#">5.11.2</a>	MPD fallback descriptor.
urn:mpeg:dash:profile:full:2011	<a href="#">8.2</a>	Identifier for Full profile.
urn:mpeg:dash:profile:isoff-on-demand:2011	<a href="#">8.3</a>	Identifier for ISO Base media file format On Demand profile.
urn:mpeg:dash:profile:isoff-live:2011	<a href="#">8.4</a>	Identifier for ISO Base media file format live profile.
urn:mpeg:dash:profile:isoff-main:2011	<a href="#">8.5</a>	Identifier for ISO Base media file format main profile.
urn:mpeg:dash:profile:mp2t-main:2011	<a href="#">8.6</a>	Identifier for MPEG-2 TS main profile.
urn:mpeg:dash:profile:mp2t-simple:2011	<a href="#">8.7</a>	Identifier for MPEG-2 TS simple profile.
urn:mpeg:dash:profile:isoff-ext-live:2014	<a href="#">8.8</a>	Identifier for ISO Base media file format extended live profile.
urn:mpeg:dash:profile:isoff-ext-on-demand:2014	<a href="#">8.9</a>	Identifier for ISO Base media file format extended On Demand profile.
urn:mpeg:dash:profile:isoff-common:2014	<a href="#">8.10</a>	Identifier for ISO Base media file format common profile.
urn:mpeg:dash:profile:isoff-broadcast:2015	<a href="#">8.11</a>	Identifier for ISO Base media file format broadcast TV profile
urn:mpeg:dash:srd:2014	<a href="#">H.1</a>	Scheme identifier for Spatial Relationship Description. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in <a href="#">Annex H</a> .
urn:mpeg:dash:srd:dynamic:2016	<a href="#">H.1</a>	Scheme to signal dynamic SRD. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in <a href="#">Annex H</a> .
urn:mpeg:dash:urlparam:2014	<a href="#">I.2</a>	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 <a href="#">Annex I</a> .
urn:mpeg:dash:urlparam:2016	<a href="#">I.3</a>	Scheme identifier for indicating usage of the extended parameterization scheme. If signalled in an Adaptation Set, the Adaptation Set shall follow the description in <a href="#">Annex I</a> .
urn:mpeg:dash:resolutionSwitching:2016	<a href="#">Annex J</a>	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 [5.3](#). 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](#).

### 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 the files referenced 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 the files referenced 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 the files referenced 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 referenced 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 [Clause 8](#). Certain profiles as defined in [Clause 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 [5.3](#).

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<sup>[3]</sup> 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<sup>[8]</sup>.

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<sup>[9]</sup> 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 the files referenced in [Annex B](#). In case of inconsistencies, the schema in the files of [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-3/en>. These files include the schema as well as all examples provided in [Annex G](#).

```
<?xml version="1.0"?>
<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 namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>
  <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>
  <!-- MPD: main element -->
  <xs:element name="MPD" type="MPDtype"/>
  <!-- MPD Type -->
  <xs:complexType name="MPDtype">
    ...
  </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 the files referenced 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)

This revision added the following elements and attributes to the schema defined in the files referenced 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 this revision

This revision adds the following elements and attributes to the schema defined in the files referenced 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

## 5.3 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](#).
- 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 [Clause 6](#). 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 [5.3.6](#) 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](#).

The summary of the semantics of the attributes and elements within an **MPD** element are provided in [Table 3](#). 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
<b>MPD</b>		The root element that carries the Media Presentation Description for a Media Presentation.
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are <b>non-bold</b> and preceded with an @ .		

**Table 3 (continued)**

Element or Attribute Name	Use	Description
@id	O	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 <a href="#">Clause 8</a> . 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 DQUOTE characters, i.e. including only the <code>unencodedv</code> or <code>encodedv</code> elements respectively. As profile identifier, the URI defined for the conforming Media Presentation profiles as described in <a href="#">Clause 8</a> shall be used.
@type	OD default: static	specifies the type of the Media Presentation. For static Media Presentations ( <code>@type="static"</code> ), all Segments are available between the <code>@availabilityStartTime</code> and the <code>@availabilityEndTime</code> . For dynamic Media Presentations ( <code>@type="dynamic"</code> ), Segments typically have different availability times. For details, refer to subclause <a href="#">5.3.9.5.3</a> . In addition, the Media Presentation Description may be updated in dynamic Media Presentations, i.e. the <code>@minimumUpdatePeriod</code> may be present. NOTE Static Media Presentations are typically used for On-Demand services, whereas dynamic Media Presentations are used for live services.
@availabilityStartTime	CM shall be present for <code>@type='dynamic'</code>	For <code>@type='dynamic'</code> , 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. For <code>@type="static"</code> 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.
@publishTime	OD shall be present for <code>@type='dynamic'</code>	specifies the wall-clock time when the MPD was generated and published at the origin server. MPDs with a later value of <code>@publishTime</code> shall be an update as defined in subclause <a href="#">5.4</a> to MPDs with earlier <code>@publishTime</code> .
@availabilityEndTime	O	specifies the latest Segment availability end time for any Segment in the Media Presentation. When not present, the value is unknown.
@mediaPresentationDuration	O	specifies the duration of the entire Media Presentation. If the attribute is not present, the duration of the Media Presentation is unknown. This attribute shall be present when neither the attribute <code>MPD@minimumUpdatePeriod</code> nor the <code>Period@duration</code> of the last Period are present.
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 3** (continued)

Element or Attribute Name	Use	Description
@minimumUpdatePeriod	0	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. 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 <a href="#">5.4</a> . If this attribute is not present, it indicates that the MPD does not change. If <b>MPD@type</b> is not 'dynamic', @minimumUpdatePeriod shall not be present. Details on the use of the value of this attribute are specified in subclause <a href="#">5.4</a> .
@minBufferSize	M	specifies a common duration used in the definition of the Representation data rate (see @bandwidth attribute in subclauses <a href="#">5.3.5.2</a> and <a href="#">5.3.5.4</a> ).
@timeShiftBufferDepth	0	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'. When not present, the value is infinite. This value of the attribute is undefined if the type attribute is equal to 'static'.
@suggestedPresentationDelay	0	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 <a href="#">7.2.1</a> . When not specified, then no value is provided and the client is expected to choose a suitable value. When @type is 'static' the value of the attribute is undefined and may be ignored.
@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.
<b>ProgramInformation</b>	0...N	specifies descriptive information about the program. For more details, refer to the description in subclause <a href="#">5.7</a> .
<b>BaseURL</b>	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 <a href="#">5.6</a> .
<b>Location</b>	0...N	specifies a location at which the MPD is available.
<b>Period</b>	1...N	specifies the information of a Period. For more details, refer to the description in subclause <a href="#">5.3.2</a> .
<b>Metrics</b>	0 ... N	specifies the DASH Metrics. For more details, see subclause <a href="#">5.9</a> .
<b>EssentialProperty</b>	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 <a href="#">5.8.4.8</a> .
<b>SupplementalProperty</b>	0 ... N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause <a href="#">5.8.4.9</a> .
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @ .		

**Table 3** (*continued*)

Element or Attribute Name	Use	Description
<b>UTCTiming</b>	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 <a href="#">5.8.4.11</a> .
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @ .		

### 5.3.1.3 XML syntax

```

<!-- MPD Type -->
<xs:complexType name="MPDtype">
  <xs:sequence>
    <xs:element name="ProgramInformation" type="ProgramInformationType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="BaseURL" type="BaseURLType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Location" type="xs:anyURI" 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:any namespace="#other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string"/>
  <xs:attribute name="profiles" type="xs:string" 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>

<!-- Presentation Type enumeration -->
<xs:simpleType name="PresentationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="static"/>
    <xs:enumeration value="dynamic"/>
  </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
<b>Period</b>		specifies the information of a Period.
<code>@xlink:href</code>	0	specifies a reference to a remote element entity that is either empty or contains one or multiple top-level elements of type <b>Period</b> .
<code>@xlink:actuate</code>	OD default: <code>onRequest</code>	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>@id</code>	0	specifies an identifier for this Period. The identifier shall be unique within the scope of the Media Presentation. If the <b>MPD</b> <code>@type</code> 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.
<code>@start</code>	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 <i>PeriodStart</i> , together with the value of the <b>MPD</b> <code>@availabilityStartTime</code> enables to derive the Segment availability times for dynamic media presentations. For details, refer to subclause 5.3.9.
<code>@duration</code>	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.
<code>@bitstreamSwitching</code>	OD Default: <code>false</code>	When set to 'true', this is equivalent as if the <b>AdaptationSet</b> <code>@bitstreamSwitching</code> for each Adaptation Set contained in this Period is set to 'true'. In this case, the <b>AdaptationSet</b> <code>@bitstreamSwitching</code> attribute shall not be set to 'false' for any Adaptation Set in this Period.
<b>BaseURL</b>	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.
<b>SegmentBase</b>	0...1	specifies default Segment Base information. Information in this element is overridden by information in <b>AdaptationSet</b> . <b>SegmentBase</b> and <b>Representation</b> . <b>SegmentBase</b> , if present. For more details, see subclause 5.3.9.
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 4 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>SegmentList</b>	0...1	specifies default Segment List information. Information in this element is overridden by information in <b>AdaptationSet.SegmentList</b> and <b>Representation.SegmentList</b> , if present. For more details, see subclause <a href="#">5.3.9</a> .
<b>SegmentTemplate</b>	0...1	specifies default Segment Template information. Information in this element is overridden by information in <b>AdaptationSet.SegmentTemplate</b> and <b>Representation.SegmentTemplate</b> , if present. For more details, see subclause <a href="#">5.3.9</a> .
<b>AssetIdentifier</b>	0...1	specifies that this Period belongs to a certain asset. For more details, see subclause <a href="#">5.8.5.7</a> .
<b>EventStream</b>	0...N	specifies an event stream. For more details, see subclause <a href="#">5.10.2</a> .
<b>AdaptationSet</b>	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 subclause <a href="#">5.3.3</a> .
<b>Subset</b>	0...N	specifies a Subset. For more details, see subclause <a href="#">5.3.8</a> .
<b>SupplementalProperty</b>	0...N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause <a href="#">5.8.4.9</a> .
<b>EmptyAdaptationSet</b>	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 <a href="#">5.8.5.8</a> .
<b>GroupLabel</b>	0 ... N	A summary label for a group of Labels. For more details, refer to subclause <a href="#">5.3.10</a> .
<b>Preselection</b>	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 <a href="#">5.3.11</a> .
<b>Key</b>		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 <b>bold</b> ; attributes are non-bold and preceded with an @.

### 5.3.2.3 XML syntax

```

<!-- Period -->
<xs:complexType name="PeriodType">
  <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="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 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 sequence 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](#).

### 5.3.3 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 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](#). 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
<code>AdaptationSet</code>		Adaptation Set description.
<code>@xlink:href</code>	O	specifies a reference to a remote element entity that shall contain exactly one element of type <code>AdaptationSet</code> .
<code>@xlink:actuate</code>	OD default: 'onRequest'	specifies the processing instructions, which can be either "onLoad" or "onRequest".
<code>@id</code>	O	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.
<code>@group</code>	O	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 <a href="#">5.3.3.1</a> .
<code>CommonAttributesElements</code>	—	specifies the common attributes and elements (attributes and elements from base type <code>RepresentationBaseType</code> ). For details, see subclause <a href="#">5.3.7</a> .
<code>@lang</code>	O	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.
<code>@contentType</code>	O	specifies the media content component type for this Adaptation Set. A value of the top-level Content-type 'type' value as defined in IETF RFC 6838:2013, Clause 4 shall be taken.  If not present, the media content component type may be defined for each media component or it may be unknown.
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 5** (continued)

Element or Attribute Name	Use	Description
@par	O	<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 * sarx) : (@height * sary), with sarx the first number in @sar and sary 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	O	<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	O	<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	O	<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	O	<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	O	<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>
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 5** (continued)

Element or Attribute Name	Use	Description
@maxHeight	O	specifies the maximum @height value in all Representations in this Adaptation Set. This value has the same units as the @height attribute. If not present, the value is unknown.
@minFrameRate	O	specifies the minimum @framerate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute. If not present, the value is unknown.
@maxFrameRate	O	specifies the maximum @framerate value in all Representations in this Adaptation Set. This value is encoded in the same format as the @frameRate attribute. If not present, the value is unknown.
@segmentAlignment	OD default: <i>false</i>	When not set to 'false', this specifies that for any two Representations, X and Y, within the same Adaptation Set, the <i>m</i> -th Segment of X and the <i>n</i> -th Segment of Y are non-overlapping (as defined in subclause 4.5.3) whenever <i>m</i> is not equal to <i>n</i> . For Adaptation Sets containing Representations with multiple media content components, this attribute value shall be either 'true' or 'false'. For Adaptation Sets containing Representations with a single media content component, when two <b>AdaptationSet</b> elements within a Period share the same integer value for this attribute, then for any two Representations, X and Y, within the union of the two Adaptation Sets, the <i>m</i> -th Segment of X and the <i>n</i> -th Segment of Y are non-overlapping (as defined in subclause 4.5.3) whenever <i>m</i> is not equal to <i>n</i> .
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 5** (continued)

Element or Attribute Name	Use	Description
@bitstreamSwitching	O	<p>When this flag is set to 'true', the following applies:</p> <ul style="list-style-type: none"> <li>— All Representations in the Adaptation Set shall have the same number <math>M</math> of Media Segments;</li> <li>— Let <math>R_1, R_2, \dots, R_N</math> be all the Representations within the Adaptation Set.</li> <li>— Let <ul style="list-style-type: none"> <li>— <math>S_{i,j}</math>, for <math>j &gt; 0</math>, be the <math>j^{th}</math> Media Segment in the <math>i^{th}</math> Representation (i.e. <math>R_i</math>)</li> <li>— if present, let <math>S_{i,0}</math> be the Initialization Segment in the <math>i^{th}</math> Representation, and</li> <li>— if present, let <math>B_i</math> be the Bitstream Switching Segment in the <math>i^{th}</math> Representation.</li> </ul> </li> <li>— The sequence of <ul style="list-style-type: none"> <li>— any Initialization Segment, if present, in the Adaptation Set, with,</li> <li>— if Bitstream Switching Segments are present, <ul style="list-style-type: none"> <li><math>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}</math></li> </ul> </li> <li>— else <ul style="list-style-type: none"> <li><math>S_{i(1),1}, S_{i(2),2}, \dots, S_{i(k),k}, \dots, S_{i(M),M}</math>,</li> </ul> </li> </ul> </li> <li>— wherein any <math>i(k)</math> for all <math>k</math> values in the range of 1 to <math>M</math>, respectively, is an integer value in the range of 1 to <math>N</math>,</li> <li>— results in a "conforming Segment sequence" as defined in subclause <a href="#">4.5.4</a> with the media format as specified in the @mimeType attribute.</li> </ul> <p>More detailed rules may be defined for specific media formats.</p> <p>For more details, refer to subclause <a href="#">5.3.3.4</a>.</p> <p>NOTE When this attribute is set to true, the DASH Client can continue decoding without re-initialization. When this attribute is set to FALSE, 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>
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 5 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@subsegmentAlignment	OD default: false	If the @subsegmentAlignment for an Adaptation Set is set to other than 'false', all following conditions shall be satisfied: <ul style="list-style-type: none"> <li>— 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).</li> <li>— For any two Representations, X and Y, within the same Adaptation Set, the <math>m</math>-th Subsegment of X and the <math>n</math>-th Subsegment of Y are non-overlapping (as defined in subclause 4.5.3) whenever <math>m</math> is not equal to <math>n</math>.</li> <li>— For Adaptation Sets containing Representations with a single media content component, when two <b>AdaptationSet</b> elements within a Period share the same integer value for this attribute, then for any two Representations, X and Y, within the union of the two Adaptation Sets, the <math>m</math>-th Subsegment of X and the <math>n</math>-th Subsegment of Y are non-overlapping (as defined in subclause 4.5.3) whenever <math>m</math> is not equal to <math>n</math>.</li> </ul>
@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.
<b>Accessibility</b>	0 ... N	specifies information about accessibility scheme For more details, refer to subclauses 5.8.1 and 5.8.4.3.
<b>Role</b>	0 ... N	specifies information on role annotation scheme For more details, refer to subclauses 5.8.1 and 5.8.4.2.
<b>Rating</b>	0 ... N	specifies information on rating scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.4.
<b>Viewpoint</b>	0 ... N	specifies information on viewpoint annotation scheme. For more details, refer to subclauses 5.8.1 and 5.8.4.5.
<b>ContentComponent</b>	0...N	specifies the properties of one media content component contained in this Adaptation Set. For more details, refer to subclause 5.3.4.
<b>Key</b>		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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.

**Table 5** (continued)

Element or Attribute Name	Use	Description
<b>BaseURL</b>	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 <a href="#">5.6</a> .
<b>SegmentBase</b>	0...1	specifies default Segment Base information. Information in this element is overridden by information in the <b>Representation.SegmentBase</b> , if present. For more details, see subclause <a href="#">5.3.9</a> .
<b>SegmentList</b>	0...1	specifies default Segment List information. Information in this element is overridden by information in the <b>Representation.SegmentList</b> , if present. For more details, see subclause <a href="#">5.3.9</a> .
<b>SegmentTemplate</b>	0...1	specifies default Segment Template information. Information in this element is overridden by information in the <b>Representation.SegmentTemplate</b> , if present. For more details, see subclause <a href="#">5.3.9</a> .
<b>Representation</b>	0 ... N	specifies a Representation. 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 <b>AdaptationSet</b> element. For more details, refer to subclause <a href="#">5.3.5</a> .
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

### 5.3.3.3 XML syntax

```

<!-- Adaptation Set -->
<xs:complexType name="AdaptationSetType">
    <xs:complexContent>
        <xs:extension base="Representation BaseType">
            <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 name="id" type="xs:unsignedInt"/>
            <xs:attribute name="group" type="xs:unsignedInt"/>
            <xs:attribute name="lang" type="xs:language"/>
            <xs:attribute name="contentType" type="IETF RFC 6838 ContentType Type"/>
            <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:extension>
    </xs:complexContent>
</xs:complexType>

```

```

        <xs:attribute name="subsegmentStartsWithSAP" type="SAPType" default="0"/>
        <xs:attribute name="bitstreamSwitching" type="xs:boolean"/>
    </xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- Ratio Type for sar and par -->
<xs:simpleType name="RatioType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]*:[0-9]*"/>
    </xs:restriction>
</xs:simpleType>

<!-- Type for Frame Rate -->
<xs:simpleType name="FrameRateType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]*[0-9] ([/[0-9]*[0-9])?"/>
    </xs:restriction>
</xs:simpleType>

<!-- Type for IETF RFC6838 Content Type -->
<xs:simpleType name="IETFRFC6838ContentTypeType">
    <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>
</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**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>switching</b>		Switching logic description for the associated Representation
@interval	M	<p>specifies the interval between two switching points in the scale of the @timescale on Representation level. Any Segment for which the earliest presentation 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 switch-to opportunity, i.e. it enables to switch to this Representation with the switching 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 switching will be described.</p>
@type	OD default: 'media'	specifies the switching strategy for the switch points identified in by the @interval attribute. Switching strategies are defined in <a href="#">Table 7</a> .

**Table 7 — Switching strategies**

<b>Type</b>	<b>Description</b>
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 sequence" as defined in <a href="#">4.5.4</a> with the media format as specified in the @mimeType 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 signaled in the Initialization Segment.</p>

The XML schema snippet is as follows:

```

<!-- Switching -->
<xs:complexType name="SwitchingType">
  <xs:attribute name="interval" type="xs:unsignedInt" use="required"/>
  <xs:attribute name="type" type="SwitchingTypeType"/>
  <xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>
<!--Switching Type type enumeration -->
<xs:simpleType name="SwitchingTypeType">
  <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. 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.

If the content author signals the ability of Adaptation Set switching and `switching` element is provided, the signaled 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 signaled with one Adaptation Set, but not with both as for example one Adaptation Set signaling 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](#). 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
<b>ContentComponent</b>		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 ISOBMFF segments and PID in MPEG-2 TS segments) described by this <b>ContentComponent</b> 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 <a href="#">Table 5</a> for @lang attribute.
@contentType	0	Same semantics as in <a href="#">Table 5</a> for @contentType attribute.
@par	0	Same semantics as in <a href="#">Table 5</a> for @par attribute.
@tag	0	specifies the tag of the Content Component which may be used for selection purposes towards the decoder.
<b>Accessibility</b>	0 ... N	Same semantics as in <a href="#">Table 5</a> for <b>Accessibility</b> element.
<b>Role</b>	0 ... N	Same semantics as in <a href="#">Table 5</a> for <b>Role</b> element.
<b>Rating</b>	0 ... N	Same semantics as in <a href="#">Table 5</a> for <b>Rating</b> element.
<b>Viewpoint</b>	0 ... N	Same semantics as in <a href="#">Table 5</a> for <b>Viewpoint</b> element.
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

### 5.3.4.3 XML syntax

```
<!-- Content Component -->
<xs:complexType name="ContentComponentType">
  <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="xs:string"/>
  <xs:attribute name="par" type="RatioType"/>
  <xs:attribute name="tag" type="TagType"/>
  <xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>
```

### 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 sequence 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 sequence 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 Yi, 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. 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
<code>Representation</code>		This element contains a description of a Representation.
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 9** (continued)

Element or Attribute Name	Use	Description
@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 <a href="#">5.3.9.4.4</a>, 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 <math>\text{@minBufferTime} * \text{@bandwidth}</math> bits have been received (i.e. at time <math>\text{@minBufferTime}</math> 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 <a href="#">5.3.5.4</a>.</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 <b>Representation@id</b> attributes.

**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 @; list of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type.

**Table 9** (continued)

Element or Attribute Name	Use	Description
@associationType	O	<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 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	O	<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"> <li>— The media streams have the same number of stream access points of type 1 to 3.</li> <li>— The values of <math>T_{SAP}</math>, <math>T_{DEC}</math>, <math>T_{EPT}</math>, and <math>T_{PTF}</math> of the <math>i</math>-th SAP of type 1 to 3 in one media stream are identical to the values of <math>T_{SAP}</math>, <math>T_{DEC}</math>, <math>T_{EPT}</math>, and <math>T_{PTF}</math>, respectively, of the <math>i</math>-th SAP of type 1 to 3 in the other media streams for any value of <math>i</math> from 1 to the number of SAPs of type 1 to 3 in any of the media streams.</li> <li>— A media stream formed by concatenating the media stream of a first Representation until <math>I_{SAU}</math> (exclusive) of the <math>i</math>-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 <math>I_{SAU}</math> (inclusive) of the <math>i</math>-th SAP of type 1 to 3 conforms to the specification in which the media stream format is specified for any value of <math>i</math> 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.</li> </ul>
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

**Table 9** (continued)

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>
<b>CommonAttributesElements</b>	—	Common Attributes and Elements (attributes and elements from base type <b>RepresentationBaseType</b> ). For more details, see subclause <a href="#">5.3.7</a> .
<b>BaseURL</b>	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 <a href="#">5.6</a> .
<b>SubRepresentation</b>	0 ... N	specifies information about a Sub-Representation that is embedded in the containing Representation. For more details, see subclause <a href="#">5.3.6</a> .
<b>SegmentBase</b>	0...1	specifies default Segment Base information. For more details, see subclause <a href="#">5.3.9</a> .
<b>SegmentList</b>	0 ... 1	specifies the Segment List information. For more details, see subclause <a href="#">5.3.9</a> .
<b>SegmentTemplate</b>	0 ... 1	specifies the Segment Template information. For more details, see subclause <a href="#">3.9</a> .
<b>Key</b>		
For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @; list of elements and attributes is in <b><i>italics bold</i></b> referring to those taken from the Base type that has been extended by this type.		

### 5.3.5.3 XML syntax

```

<!-- Representation -->
<xs:complexType name="RepresentationType">
  <xs:complexContent>
    <xs:extension base="Representation BaseType">
      <xs:sequence>
        <xs:element name="BaseURL" type="BaseURLType" 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>

<!-- String without white spaces -->
<xs:simpleType name="StringNoWhiteSpaceType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[^\\r\\n\\t \\p{Z}] */>
  </xs:restriction>
</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 signaled Segment onwards. Random Access point may be signaled 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
<code>RandomAccess</code>		Random Access Information
<code>@interval</code>	M	<p>specifies the position of the random access points in the Representations. The information is specified in the scale of the <code>@timescale</code> on Representation level. Any Segment for which the MPD start time minus the <code>@t</code> value of the <code>s</code> element describing the segment is an integer multiple of the product of <code>@timescale</code> and <code>@interval</code> is a random access opportunity, i.e. it enables randomly access to this Representation with the random access 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 random access will be described.</p>
<code>@type</code>	OD default: "closed"	<p>specifies the random access strategy for the random access points in by the <code>@interval</code> attribute.</p> <p>The value shall use a type present in <a href="#">Table 11</a>.</p> <p>If the value of the type is unknown, the DASH Client is expected to ignore the containing Random Access element.</p>
<code>@minBufferSize</code>	O	<p>specifies a common duration used in the definition of the Representation data rate (see <code>@bandwidth</code> attribute in subclauses <a href="#">5.3.5.2</a> and <a href="#">5.3.5.4</a>).</p> <p>If not present, then the value of the MPD level is inherited.</p>

**Table 10 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@bandwidth	0	<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 <code>@minBufferTime * @bandwidth</code> bits have been received (i.e. at time <code>@minBufferTime</code> 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 <a href="#">5.3.5.4</a>.</p> <p>If not present, the value of the Representation is inherited.</p>

**Table 11 — Random Access Strategies**

<b>Type</b>	<b>Informative description</b>
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:

```

<!-- Random Access -->
<xs:complexType name="RandomAccessType">
    <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>
<!-- Random Access Type type enumeration -->
<xs:simpleType name="RandomAccessTypeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="closed"/>
        <xs:enumeration value="open"/>
        <xs:enumeration value="gradual"/>
    </xs:restriction>
</xs:simpleType>

```

### 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 12](#) 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 12 — Semantics of **SubRepresentation** element**

Element or Attribute Name	Use	Description
<b>SubRepresentation</b>		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 <a href="#">6.3.2.4</a> shall be available for each Media Segment in the containing Representation. If not present, then the <b>SubRepresentation</b> 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.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold preceded with an @. List of elements and attributes is in <b>italics bold</b> referring to those taken from the Base type that has been extended by this type.		

**Table 12** (continued)

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	O	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.
<b>CommonAttributesElements</b>	-	Common Attributes and Elements (attributes and elements from base type <b>RepresentationBaseType</b> ). 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 **italics bold** referring to those taken from the Base type that has been extended by this type.

### 5.3.6.3 XML syntax

```
<!-- SubRepresentation -->
<xs:complexType name="SubRepresentationType">
  <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 13](#) 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 13](#) in subclause 5.3.7.2, the syntax is provided in subclause 5.3.7.3.

The 'Use' column in [Table 13](#) 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 13 — Common Adaptation Set, Representation and Sub-Representation attributes and elements**

Element or Attribute Name	Use (see sub-clause <a href="#">5.3.7.1</a> )	Description
<b><i>Common attributes and elements</i></b>		
@profiles	0	<p>specifies the profiles which the associated Representation(s) conform to of the list of Media Presentation profiles as described in <a href="#">Clause 8</a>. 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 subclause <a href="#">5.3.1.2</a> 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" signaling, 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>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>..<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 13 (continued)**

<b>Element or Attribute Name</b>	<b>Use (see sub- clause 5.3.7.1)</b>	<b>Description</b>
@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 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 <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. As profile identifier, the brand identifier for the Segment as defined in <a href="#">Clause 6</a> shall be used.</p> <p>If not present on any level, the value may be deducted from the value of the @profiles attribute.</p>
@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>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>..<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 13** (*continued*)

<b>Element or Attribute Name</b>	<b>Use (see sub- clause <a href="#">5.3.7.1</a>)</b>	<b>Description</b>
@maximumSAPPeriod	0	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. If not present on any level, the value is unknown.
@startWithSAP	0	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. 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. If not present on any level, the value is unknown.
@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 ' <code>true</code> ', 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 ' <code>false</code> ', 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.

**Key**

For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory

For elements: &lt;minOccurs&gt;..&lt;maxOccurs&gt; (N=unbounded)

Elements are **bold**; attributes are non-bold and preceded with an @.

**Table 13** (continued)

Element or Attribute Name	Use (see sub- clause <a href="#">5.3.7.1</a> )	Description
<b>FramePacking</b>	0 ... N	specifies frame-packing arrangement information of the video media component type. When no <b>FramePacking</b> element is provided for a video component, frame-packing shall not be used for the video media component. For details, see subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.6</a>
<b>AudioChannelConfiguration</b>	0 ... N	specifies the audio channel configuration of the audio media component type. For details, see subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.7</a> .
<b>ContentProtection</b>	0 ... N	specifies information about content protection schemes used for the associated Representations. For details, see subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.1</a> .
<b>EssentialProperty</b>	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 <a href="#">5.8.4.8</a> .
<b>SupplementalProperty</b>	0 ... N	specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing. For details, see subclause <a href="#">5.8.4.9</a> .
<b>InbandEventStream</b>	0 ... N	specifies the presence of an inband event stream in the associated Representations. For details, refer to subclause <a href="#">5.10</a> .
<b>Switching</b>	0 ... N	Specifies a switch-to times and types for the associated Representations. For more details, refer to subclause <a href="#">5.3.3.4</a> . 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 signaling.
<b>RandomAccess</b>	0 ... N	Specifies a random access times and types for the associated Representations. For more details, refer to subclause <a href="#">5.3.5.5</a> . 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 signaling.
<b>GroupLabel</b>	0 ... N	specifies a summary label for a group of Labels. For more details, refer to subclause <a href="#">5.3.10</a> .
<b>Label</b>	0 ... N	specifies a textual description of the element that may be used for annotation and selection purposes. For more details, refer to subclause <a href="#">5.3.10</a> .

**Key**

For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory

For elements: &lt;minOccurs&gt;..&lt;maxOccurs&gt; (N=unbounded)

Elements are **bold**; attributes are non-bold and preceded with an @.

### 5.3.7.3 XML syntax

```

<!-- Representation base (common attributes and elements) -->
<xs:complexType name="RepresentationBaseType">
    <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="DescriptorType" 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="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:any namespace="#other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="profiles" type="xs:string"/>
    <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="xs:string"/>
    <xs:attribute name="maximumSAPPPeriod" 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"/>
    <xs:attribute name="tag" type="TagType"/>
    <xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>

```

**Table 13 (continued)**

```

<!-- Audio Sampling Rate -->
<xss:simpleType name="AudioSamplingRateType">
  <xss:restriction base="xss:UIntVector">
    <xss:minLength value="1"/>
    <xss:maxLength value="2"/>
  </xss:restriction>
</xss:simpleType>

<!-- Stream Access Point type enumeration -->
<xss:simpleType name="SAPType">
  <xss:restriction base="xss:unsignedInt">
    <xss:minInclusive value="0"/>
    <xss:maxInclusive value="6"/>
  </xss:restriction>
</xss:simpleType>

<!-- Video Scan type enumeration -->
<xss:simpleType name="VideoScanType">
  <xss:restriction base="xss:string">
    <xss:enumeration value="progressive"/>
    <xss:enumeration value="interlaced"/>
    <xss:enumeration value="unknown"/>
  </xss:restriction>
</xss:simpleType>

<!-- Tag -->
<xss:simpleType name="TagType">
  <xss:restriction base="xss:string"/>
</xss: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 14](#) 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 14 — subset element semantics**

Element or Attribute Name	Use	Description
<code>Subset</code>		specifies a Subset
<code>@contains</code>	M	specifies the Adaptation Sets contained in a Subset by providing a white-space separated list of the <code>@id</code> values of the contained Adaptation Sets.
<code>@id</code>	O	specifies a unique identifier for the Subset.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>..<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

### 5.3.8.3 XML syntax

```
<!-- Subset -->
<xss:complexType name="SubsetType">
  <xss:attribute name="contains" type="UIntVectorType" use="required"/>
  <xss:attribute name="id" type="xs:string"/>
  <xss:anyAttribute namespace="#other" processContents="lax"/>
</xss: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 [Clause 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 the 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 16](#).

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 17](#).

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 15](#) and the multiple Segment base information in [Table 16](#) 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 15 — Semantics of `segmentBase` element and *Segment Base Information* type**

Element or Attribute Name	Use	Description
<b>SegmentBase</b> <b>Segment Base Information</b>		<p>specifies Segment base element.</p> <p>This element also specifies the type for the Segment base information that is the base type for other elements.</p>
@timescale	0	<p>specifies the timescale in units per seconds to be used for the derivation of different real-time duration values in the Segment Information.</p> <p>If not present on any level, it shall be set to 1.</p> <p>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).</p>
@presentationTimeOffset	0	<p>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.</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> <p>If not present on any level, the value of the presentation time offset is 0.</p>
@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>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 15 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@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 <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 value is unknown.</p>
@indexRangeExact	OD default "false"	<p>when set to '<code>true</code>' 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>
@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 <a href="#">5.3.9.5</a>.</p> <p>If not present, no adjusted segment availability start time is defined.</p> <p>NOTE The value of "<code>INF</code>" implies availability of all segments starts at <code>MPD@availabilityStartTime</code>.</p>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 15 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<code>@availabilityTimeComplete</code>	0	<p>specifies if all Segments of all associated Representation are complete at the adjusted availability start time. The attribute shall be ignored if <code>@availabilityTimeOffset</code> 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>
<b>Initialization</b>	0 ... 1	<p>specifies the URL including a possible byte range for the Initialization Segment.</p> <p>For the type definition, refer to <a href="#">Table 17</a>.</p>
<b>RepresentationIndex</b>	0 ... 1	<p>specifies the URL including a possible byte range for the Representation Index Segment.</p> <p>For the type definition, refer to <a href="#">Table 17</a>.</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 16 — Semantics of *MultipleSegmentBaseInformation* type**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>MultipleSegmentBaseInformation</b>		specifies multiple Segment base information.
<code>@duration</code>	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 <code>@timescale</code> attribute associated to the containing Representation.</p> <p>For more details, refer to subclause <a href="#">5.3.9.5.3</a>.</p>
<code>@startNumber</code>	0	<p>specifies the number of the first Media Segment in this Representation in the Period.</p> <p>For details, refer to subclause <a href="#">5.3.9.5.3</a>.</p>
<b>Segment Base Information</b>		specifies Segment base information.
<b>SegmentTimeline</b>	0...1	<p>specifies the timeline of arbitrary Segment durations</p> <p>For more details, see subclause <a href="#">5.3.9.6</a>.</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 16 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>BitstreamSwitching</b>	0 ... 1	specifies the URL including a possible byte range for the Bitstream Switching Segment. For the type definition, refer to <a href="#">Table 17</a> .

**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 elements of type `URLType`**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
Element of type <code>URLType</code>		defines an HTTP-URL
@sourceURL	0	specifies the source URL part and shall be formatted either as an <code>&lt;absolute-URI&gt;</code> according to IETF RFC 3986:2005, subclause 4.3, with a fixed scheme of "http" or "https" or as a <code>&lt;relative-ref&gt;</code> according to IETF RFC 3986:2005, subclause 4.2. If not present, then any <b>BaseUrl</b> element is mapped to the @sourceURL attribute and the range attribute shall be present.
@range	0	specifies the byte range restricting the above HTTP-URL. 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. If not present, the element refers to the entire resource referenced in the @sourceURL attribute.

### 5.3.9.2.3 XML-Syntax

```

<!-- Segment information base -->
<xs:complexType name="SegmentBaseType">
  <xs:sequence>
    <xs:element name="Initialization" type="URLType" minOccurs="0"/>
    <xs:element name="RepresentationIndex" type="URLType" minOccurs="0"/>
    <xs:any namespace="#other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="timescale" type="xs:unsignedInt"/>
  <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="xs:string"/>
  <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>

<!-- Multiple Segment information base -->
<xs:complexType name="MultipleSegmentBaseType">
  <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:attribute name="startNumber" type="xs:unsignedInt"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Segment Info item URL/range -->
<xs:complexType name="URLType">
  <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="xs:string"/>
  <xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>
```

### 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 18](#). The XML syntax of the Segment Information is provided in subclause [5.3.9.3.3](#).

#### 5.3.9.3.2 Semantics

**Table 18 — Semantics of **SegmentList** element**

Element or Attribute Name	Use	Description
<b>SegmentList</b>		specifies Segment information.
@xlink:href	0	specifies a reference to a remote element entity that contains one or multiple elements of type <b>SegmentList</b> .
@xlink:actuate	OD default: "onRequest"	specifies the processing set, can be either "onLoad" or "onRequest"
<b>MultipleSegmentBaseInformation</b>		Multiple Segment base information as defined in subclause <a href="#">5.3.9.2, Table 16</a> .
<b>SegmentURL</b>	0 ... N	specifies a Media Segment URL and a possibly present Index Segment URL
@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 &lt;absolute-URI&gt; according to IETF RFC 3986:2005, sub-clause 4.3, with a fixed scheme of "http" or "https" or as a &lt;relative-ref&gt; according to IETF RFC 3986:2005, subclause 4.2.</p> <p>If not present, then any <b>BaseUrl</b> 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 <b>byte-range-spec</b> 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>
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 18** (continued)

Element or Attribute Name	Use	Description
@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 &lt;absolute-URI&gt; according to IETF RFC 3986:2005, sub-clause 4.3, with a fixed scheme of "http" or "https" or as a &lt;relative-ref&gt; 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 <b>BaseURL</b> element is mapped to the @index attribute and the @indexRange attribute shall be present.</p>
@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>

**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 @.

### 5.3.9.3.3 XML-Syntax

```

<!-- Segment List -->
<xs:complexType name="SegmentListType">
  <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:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Segment URL -->
<xs:complexType name="SegmentURLType">
  <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="xs:string"/>
  <xs:attribute name="index" type="xs:anyURI"/>
  <xs:attribute name="indexRange" type="xs:string"/>
  <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 19](#). The XML syntax of the Segment Information is provided in subclause [5.3.9.4.3](#).

#### 5.3.9.4.2 Semantics

**Table 19 — Semantics of **SegmentTemplate** element**

Element or Attribute Name	Use	Description
<b>SegmentTemplate</b>		specifies Segment template information.

**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 19** (continued)

Element or Attribute Name	Use	Description
<b>MultipleSegmentBaseInformation</b>		Provides the Multiple Segment base information as defined in subclause <a href="#">5.3.9.2</a> .
@media	0	specifies the template to create the Media Segment List. For more details, refer to subclause <a href="#">5.3.9.4.4</a> .
@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 <a href="#">5.3.9.4.4</a> .
@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 <a href="#">5.3.9.4.4</a> .
@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 <a href="#">5.3.9.4.4</a> .

**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

```
<!-- Segment Template -->
<xss:complexType name="SegmentTemplateType">
  <xss:complexContent>
    <xss:extension base="MultipleSegmentBaseType">
      <xss:attribute name="media" type="xs:string"/>
      <xss:attribute name="index" type="xs:string"/>
      <xss:attribute name="initialization" type="xs:string"/>
      <xss:attribute name="bitstreamSwitching" type="xs:string"/>
    </xss:extension>
  </xss:complexContent>
</xss: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 20](#).

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 20](#).

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<sup>[10]</sup> 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 20 — Identifiers for URL templates**

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

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, 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, then the entire Segment may not yet be available at the adjusted segment availability start time.

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 sequence 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` is promised to repeat 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 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 21](#). The XML syntax of the Segment Timeline is provided in subclause [5.3.9.6.3](#).

### 5.3.9.6.2 Semantics

**Table 21 — Semantics of `SegmentTimeline` element**

Element or Attribute Name	Use	Description
<code>SegmentTimeline</code>		specifies the Segment timeline information
<code>s</code>	1 .. N	<p>specifies Segment start time and duration for a contiguous sequence of segments of identical durations, referred to as series in the following.</p> <p><b>NOTE</b> The <code>s</code> elements are ordered in sequence of increasing values of the attribute <code>@t</code>.</p>
<code>@t</code>	0	<p>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 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 <code>s</code> 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 <code>s</code> element, it expresses discontinuities in the timeline.</p> <p>If not present, then the value shall be assumed to be zero for the first <code>s</code> element and for the subsequent <code>s</code> elements, the value shall be assumed to be the sum of the previous <code>s</code> element's earliest presentation time and contiguous duration [i.e. previous <code>s@t + @d * (@r + 1)</code>].</p>
<code>@n</code>	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 <code>s</code> elements plus the <code>@startNumber</code> attribute value, if present. If the value of <code>@n</code> is greater than one plus the previously calculated Segment number, it expresses that one or more prior Segments in the timeline are unavailable.</p>
<code>@d</code>	M	specifies the Segment duration or the duration of a Segment sequence, in units of the value of the <code>@timescale</code> .
<code>@k</code>	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 <a href="#">5.3.9.6.3</a>.</p>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 21 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@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 <b>s</b> element indicates that the duration indicated in @d attribute repeats until the start of the next <b>s</b> element, the end of the Period or until the next MPD update.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

### 5.3.9.6.3 XML syntax

```

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

```

### 5.3.9.6.4 Segment Sequences

Segment sequences in the Segment Timeline may be signaled 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 signaling 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.

## 5.3.10 Label and Group Label

### 5.3.10.1 Overview

Labels provide the ability to annotate data structures in a DASH Media Presentation to provide a description of the context of the element to which label is assigned. Such labels may for example be used by a DASH Client or the associated application in order to provide a selection choice to the user. The label may also be used for simple annotation in another context.

In addition, a `GroupLabel` element may be added on a higher level in order to provide a summary or title of the labels collected in a group. An example may be that this is used in a menu in order to provide a context of the menu of the labels.

All labels with the same `@id` shall be on the same level in the MPD. If a label is present on a lower level with the same `@id` as present on a higher level, the element is expected to be ignored.

The label itself may have a language assigned that describes the language of the label. The label may be used by the DASH Client in order to present the labels according to the user preferences.

The semantics of the label element and the group label element are provided in [Table 22](#) and [Table 23](#), respectively, in subclause [5.3.10.2](#), the XML syntax is provided in subclause [5.3.10.3](#).

### 5.3.10.2 Semantics

**Table 22 — Semantics of `label` element**

Element or Attribute Name	Use	Description
<code>label</code>		The label and the actual text of the label that annotates the element in the DASH Media Presentation

**Table 22 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@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 23 — Semantics of GroupLabel element**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>GroupLabel</b>		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 id shall be unique in the MPD for group labels. The id describes the group label for Labels with the same value of the ID as this the value of this attribute.
@lang	0	Specifies the language of the label. If not present, the language of the label is unknown.

### 5.3.10.3 XML Syntax

```
<!-- Label and Group Label -->
<xs:complexType name="LabelType">
  <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>
```

## 5.3.11 Preselection

### 5.3.11.1 Overview

The concept of Preselection is primarily motivated for the purpose of Next Generation Audio (NGA) codecs in order 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 be extended and be used also for other media types and codecs.

Each Preselection is associated to a bundle. A bundle is a set of media components which may be consumed jointly by a single decoder instance. Elements are addressable and separable components of a bundle and may be selected or deselected dynamically by the application, either directly or indirectly by the use of Preselections. Media components are mapped to Adaptation Sets by either a one-to-one mapping or by the inclusion of multiple media components in a single Adaptation Sets. Furthermore, Representations in one Adaptation Set may contain multiple media components that are multiplexed on elementary stream level or on file container level. In the multiplexing case, each media component is mapped to a Media Content component as defined in subclause 5.3.4. Each media component in the bundle is therefore identified and referenced by the @id of a Media Content component, or, if only a single media component is contained in the Adaptation Set, by the @id of an Adaptation Set.

Each bundle includes a main media component that contains the decoder specific information and bootstraps the decoder. The Adaptation Set that contains the main media component is referred to as

main Adaptation Set. The main media component shall always be included in any Preselection that is associated to a bundle. In addition, each bundle may include one or multiple partial Adaptation Sets. Partial Adaptation Sets may only be processed in combination with the main Adaptation Set.

A Preselection defines a subset of media component in a bundle that are expected to be consumed jointly. A Preselection is identified by a unique tag towards the decoder. Multiple Preselection instances can refer to the same set of streams in a bundle. Only media components of the same bundle can contribute to the decoding and rendering of a Preselection.

In the case of next generation audio, a Preselection is a personalization option that is associated with one or more audio components from one plus additional parameters like gain, spatial location to produce a complete audio experience. A Preselection can be considered the NGA-equivalent of alternative audio tracks containing complete mixes using traditional audio codecs.

A bundle, Preselection, main media component, main Adaptation Set and partial Adaptation Sets may be defined by one of the two means:

- A preselection descriptor is defined in subclause [5.3.11.2](#). Such a descriptor enables simple setups and backward compatibility but may not be suitable for advanced use cases.
- A preselection element as defined in subclauses [5.3.11.3](#) and [5.3.11.4](#). The semantics of the Preselection element is provided in [Table 24](#) in subclause [5.3.11.3](#), the XML syntax is provided in subclause [5.3.11.4](#).

The instantiation of the introduced concepts using both methods is provided in the following.

In both cases, if the Adaptation Set is not including the main Adaptation Set, then the Essential descriptor shall be used together with the `@schemeIdURI` as defined in subclause [5.3.11.2](#).

### **5.3.11.2 Preselection Descriptor**

A scheme is defined to be used with an Essential Descriptor as "`urn:mpeg:dash:preselection:2016`". The value of the Descriptor provides two fields, separated by a comma:

- the tag of the Preselection;
- the id of the contained content components of this Preselection list as white space separated list in processing order. The first id defines the main media component.

If the Adaptation Set contains the main media component, then the Supplemental descriptor may be used to describe contained Preselections in the Adaptation Set.

If the Adaptation Set does not contain the main media component then the Essential Descriptor shall be used.

The bundle is inherently defined by all media components that are included in all Preselections that include the same main media component. Preselections are defined by the metadata that is assigned to each of the media components that are included in the Preselection. This signalling may be simple for basic use cases but is expected to not provide a full coverage for all use cases. Therefore, the Preselection element is introduced in subclause [5.3.11.3](#) to cover more advanced use cases.

### **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 24](#). The selection of Preselections is based on the contained attributes and elements in the Preselection element.

**Table 24 — Semantics of PreSelection element**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>Preselection</b>		
@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 tag defines the main media component.
@lang	0	same semantics as in <a href="#">Table 5</a> for @lang attribute.
<b>Accessibility</b>	0 ... N	specifies information about accessibility scheme. For more details, refer to subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.3</a> .
<b>Role</b>	0 ... N	specifies information on role annotation scheme. For more details, refer to subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.2</a> .
<b>Rating</b>	0 ... N	specifies information on rating scheme. For more details, refer to subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.4</a> .
<b>Viewpoint</b>	0 ... N	specifies information on viewpoint annotation scheme. For more details, refer to subclauses <a href="#">5.8.1</a> and <a href="#">5.8.4.5</a> .
<b>CommonAttributesElements</b>	-	specifies the common attributes and elements (attributes and elements from base type <b>Representation BaseType</b> ). For details, see subclause <a href="#">5.3.7</a> .
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>..<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

#### 5.3.11.4 XML Syntax for Preselection element

```

<!-- Preselection -->
<xs:complexType name="PreselectionType">
  <xs:complexContent>
    <xs:extension base="Representation BaseType">
      <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:extension>
  </xs:complexContent>
</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 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.

**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 [5.3](#) apply to the MPD after dereferencing all remote element entities.

### 5.5.2 Syntax and semantics

[Table 25](#) provides the XLINK attributes that are used in this document and shall be supported accordingly.

**Table 25 — 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> .

**Table 25** (*continued*)

<b>Attribute</b>	<b>Comments and Usage</b>
@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 @xlink:show is optional with fixed setting @xlink:show="embed".</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 <a href="#">5.5.3</a>.</p>
@xlink:actuate	<p>Defines the desired timing of dereferencing a remote element entity from within an MPD as defined in W3C XLINK. The following attribute values are allowed in this document:</p> <ol style="list-style-type: none"> <li>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 @xlink:href with @xlink:actuate set to onLoad.</li> <li>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 Period, dereferencing an Adaptation Set link when it appears to contain Representations that will be needed, and so on.</li> </ol>

The restricted schema for XLINK in the context of the ISO/IEC 23009 series is referred to as "xlink.xsd" in any schema in this document and defined in the files referenced in [Annex B](#).

### 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`.

## 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 26](#). The XML syntax of the Base URL is provided in subclause [5.6.3](#).

### 5.6.2 Semantics

**Table 26 — Semantics of `BaseURL` element**

Element or Attribute Name	Use	Description
<code>BaseURL</code>		A URL that can be used as Base URL. The content of this element is a URI string as described in subclause <a href="#">5.6.4</a> .
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 26** (continued)

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@serviceLocation	0	This attribute specifies a relationship between Base URLs such that <b>BaseURL</b> 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.  If not present, no relationship to any other Base URL is known.
@byteRange	0	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 <a href="#">E.2</a> .  If not present, HTTP partial GET requests may not be converted into regular GET requests.  NOTE Such alternative requests are expected to not be used unless the DASH application requires this. For more details, refer to <a href="#">Annex E</a> .
@availabilityTimeOffset	0	specifies an offset to define the adjusted segment availability time. For semantics, refer to <a href="#">Table 13</a> .  If the value is present in <b>SegmentBase</b> then this attribute should not be present. If present in <b>SegmentBase</b> and <b>BaseURL</b> , the value in <b>BaseURL</b> shall be ignored.
@availabilityTimeComplete	0	specifies if all Segments of all associated Representation are complete at the adjusted availability start time. For semantics, refer to <a href="#">Table 13</a> .  If the value is present in <b>SegmentBase</b> then this attribute should not be present. If present in <b>SegmentBase</b> and <b>BaseURL</b> , the value in <b>BaseURL</b> shall be ignored.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

### 5.6.3 XML syntax

```

<!-- Base URL -->
<xsd:complexType name="BaseURLType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:anyURI">
      <xsd:attribute name="serviceLocation" type="xsd:string"/>
      <xsd:attribute name="byteRange" type="xsd:string"/>
      <xsd:attribute name="availabilityTimeOffset" type="xsd:double"/>
      <xsd:attribute name="availabilityTimeComplete" type="xsd:boolean"/>
      <xsd:anyAttribute namespace="#other" processContents="lax"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd: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](#).

## 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 [Table 27](#) of subclause [5.7.2](#). The XML syntax of `ProgramInformation` element is provided in subclause [5.7.3](#).

### 5.7.2 Semantics

**Table 27 — Program information semantics**

Element or Attribute Name	Use	Description
<code>ProgramInformation</code>		specifies descriptive information about the program
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 27** (*continued*)

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<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.
<b>Title</b>	0 ... 1	specifies the title for the Media Presentation.
<b>Source</b>	0 ... 1	specifies information about the original source (for example content provider) of the Media Presentation.
<b>Copyright</b>	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

```
<!-- Program Information -->
<xss:complexType name="ProgramInformationType">
  <xss:sequence>
    <xss:element name="Title" type="xss:string" minOccurs="0"/>
    <xss:element name="Source" type="xss:string" minOccurs="0"/>
    <xss:element name="Copyright" type="xss:string" minOccurs="0"/>
    <xss:any namespace="#other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xss:sequence>
  <xss:attribute name="lang" type="xss:language"/>
  <xss:attribute name="moreInformationURL" type="xss:anyURI"/>
  <xss:anyAttribute namespace="#other" processContents="lax"/>
</xss: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 [Table 28](#) of subclause [5.8.2](#). 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 28 — 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 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.
<code>@value</code>	O	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>	O	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.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

### 5.8.3 XML syntax of generic descriptor

```
<!-- Descriptor -->
<xs:complexType name="DescriptorType">
  <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

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

The **ContentProtection** descriptors should 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.

When multiple **ContentProtection** elements are present with different SystemID values, each element shall describe a key management and protection scheme that is sufficient to access and present the Representation.

#### 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 23001-8.

This document also defines frame-packing schemes in subclause [5.8.5.6](#). These schemes are maintained for backward-compatibility, but it recommended to use the signalling as defined in ISO/IEC 23001-8.

#### 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 signaling 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.

## 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

The following defines a set of URIs that identify specific content protection schemes, i.e. schemes contained in the `ContentProtection` element:

- 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 of the `@value` attribute 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.

- 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.

- For Representations based on ISO/IEC 14496-12, a content protection scheme using the Protection System Specific Header Box defined in ISO/IEC 23001-7 may be identified in the `ContentProtection` element. In this case, a UUID URN as defined in IETF RFC 4122 indicating the UUID specified in the SystemId field of the Protection System Specific Header Box shall be used. This does not imply that such schemes cannot define alternative URNs, or that all UUID URNs refer to schemes of this type.

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:

<code>MP4-PROTECTION-VALUE</code> = 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"

### 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<sup>[19]</sup>, 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 23001-8.
- 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 23001-8.

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
VIDEO-FRAME-PACKING-TYPE = STEREO-VIDEO-FORMAT-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 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 23001-8. 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 29](#). `Role@value` shall be assigned to Adaptation Sets that contain a media component type to which this role is associated.

**Table 29 — `Role@value` attribute for scheme with a value "`urn:mpeg:dash:role:2011`"**

<code>Role@value</code>	Description
<code>caption</code>	Captions (see NOTE 3 below).

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" may be used to enable assignment of a "kind" value in HTML 5 applications for tracks exposed from a DASH MPD.

**Table 29 (continued)**

<b>Role@value</b>	<b>Description</b>
subtitle	Subtitles (see NOTE 3 below).
Main	Main media component(s) which is/are intended for presentation if no other information is provided.
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).
supplementary	Media content component that is supplementary to a media content component of a different media component type (see NOTE 1 below).
commentary	Experience that contains a commentary (e.g. director's commentary) (typically audio).
dub	Experience that contains an element that is presented in a different language from the original (e.g. dubbed audio, translated captions).
description	Textual or audio media component containing a textual description (intended for audio synthesis) or an audio description describing a visual component.
sign	Visual media component representing a sign-language interpretation of an audio component.
metadata	Media component containing information intended to be processed by application specific elements.
enhanced-audio-intelligibility	Experience containing an element for improved intelligibility of the dialogue.
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.

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" may be used to enable assignment of a "kind" value in 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:

<b>ROLE-VALUE</b> = "caption" / "subtitle" / "main" / "alternate" / "supplementary"
<b>ROLE-VALUE</b> = / "commentary"/ "dub" / "description" / "sign" / "metadata"
<b>ROLE-VALUE</b> = / "enhanced-audio-intelligibility" / "emergency"

### 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 **STEROID-VALUE** as defined in the following ABNF notation according to IETF RFC 5234:

```

STEROID-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 30](#), 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 30 — Different UTC timing Methods**

@schemeIdURI	Description
urn:mpeg:dash:utc:ntp:2014	<p>The identifier indicates that the <b>@value</b> 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 <b>@value</b> attribute is allowed.</p> <p>For exact syntax of the value of the <b>@value</b> attribute, see below.</p>
urn:mpeg:dash:utc:sntp:2014	<p>The identifier indicates that the <b>@value</b> 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 <b>@value</b> attribute, see below.</p>
urn:mpeg:dash:utc:http-head:2014	<p>The identifier indicates that the <b>@value</b> 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 <b>@value</b> 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 <b>@value</b> attribute, see below.</p>

**Table 30 (continued)**

@schemeIdURI	Description
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 time value 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>
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 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 a time 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>

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 IETF RFC 3896 and
; 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  
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<sup>[20]</sup>. 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 signaled 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)<sup>[13]</sup>, Open Standard for Authorization (OAuth) 2.0<sup>[14]</sup>, OASIS Security Assertion Markup Language (SAML)<sup>[15]</sup>, 3GPP Generic Authentication Architecture (GAA)

[16], or 3GPP Generic Bootstrapping Architecture (GBA)[17]. This section describes generic signaling 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 signaled 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 signaling 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 signaled 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 signaling may be used, but the detailed signaling 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.

## 5.9 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 [Table 31](#) of subclause [5.9.2](#). 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 31 — Semantics of **Metrics** element**

Element or Attribute Name	Use	Description
<b>Metrics</b>		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 <a href="#">Annex D</a> , separated by a comma.
<b>Range</b>	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.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 31** (continued)

Element or Attribute Name	Use	Description
@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 <b>MPD@type='dynamic'</b>, the start time is indicated in wall clock time by adding the value of this attribute to the value of the <b>MPD@availabilityStartTime</b> attribute.</p> <p>For services with <b>MPD@type='static'</b>, 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 <b>MPD@availabilityStartTime</b> 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>
<b>Reporting</b>	1 ... N	<p>specifies information about the requested reporting method and formats.</p> <p>For more details, refer to subclause <a href="#">5.9.4</a>.</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

```

<!-- Metrics -->
<xss:complexType name="MetricsType">
  <xss:sequence>
    <xss:element name="Range" type="RangeType" minOccurs="0" maxOccurs="unbounded"/>
    <xss:element name="Reporting" type="DescriptorType" maxOccurs="unbounded"/>
    <xss:any namespace="#other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xss:sequence>
  <xss:attribute name="metrics" type="xs:string" use="required"/>
  <xss:anyAttribute namespace="#other" processContents="lax"/>
</xss:complexType>

<!-- Metrics Range -->
<xss:complexType name="RangeType">
  <xss:attribute name="time" type="xs:duration"/>
  <xss:attribute name="duration" type="xs:duration"/>
  <xss:anyAttribute namespace="#other" processContents="lax"/>
</xss: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 typically has a duration. Events include DASH specific signalling or application-specific events. In the latter case, a proper 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. This enables a DASH Client 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](#).

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 [Table 32](#) of subclause [5.10.2.2](#) and the semantics of the attributes within the `Event` element are provided in [Table 33](#) of subclause [5.10.2.2](#). The XML syntax of `EventStream` and `Event` element is provided in subclause [5.10.2.3](#).

### 5.10.2.2 Semantics

**Table 32 — Event Stream Semantics**

Element or Attribute Name	Use	Description
<code>EventStream</code>		specifies event Stream
<code>@xlink:href</code>	O	specifies a reference to an external <code>EventStream</code> element
<code>@xlink:actuate</code>	OD default: <code>onRequest</code>	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 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. A URL may resolve to an Internet location, and a location that does not 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.
<code>@timescale</code>	O	specifies the timescale in units per seconds to be used for the derivation of different real-time duration values in the <code>Event</code> elements. If not present on any level, it shall be set to 1.
<code>Event</code>	0 ... N	specifies one event. For details see <a href="#">Table 33</a> . Events in Event Streams shall be ordered such that their presentation time is non-decreasing.
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table 33 — Event Semantics**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
<b>Event</b>		specifies an event and contains the message of the event, formatted as a string. The content of this element depends on the event scheme.
@presentationTime	OD default: 0	<p>specifies the presentation time of the event relative to the start of the Period.</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>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 <b>Event</b> 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>
@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 This attribute is an alternative to specifying a complete XML element(s) in the Event. It is useful when an event leans itself to a compact string representation.</p>
<b>Key</b>		
For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory		
For elements: <minOccurs>...<maxOccurs> (N=unbounded)		
Elements are <b>bold</b> ; attributes are non-bold and preceded with an @.		

### 5.10.2.3 XML-Syntax

```

<!-- Event Stream -->
<xs:complexType name="EventStreamType">
  <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 name="schemeIdUri" type="xs:anyURI" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
  <xs:attribute name="timescale" type="xs:unsignedInt"/>
</xs:complexType>

<!-- Event -->
<xs:complexType name="EventType">
  <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="messageData" type="xs:string"/>
  <xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>
```

### 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 that is present in a Representation shall be indicated by an `InbandEventStream` element on Adaptation Set or Representation level. The `InbandEventStream` type is defined in subclause [5.10.2, Table 32](#).

One Representation may contain multiple inband Event streams, each indicated by a separate `InbandEventStream` element.

### 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') also provides signalling that are specific for the DASH operations. The event scheme identifier and the events are defined in subclause [5.10.4](#).

A Media Segment if encapsulated in ISO BMFF may contain one or more event message ('emsg') boxes. If present, any 'emsg' box shall be placed before any 'moof' box.

The carriage of event messages in the MPEG-2 TS is described in subclause [5.10.3.3.5](#).

Event message boxes with scheme identifiers 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 time delta and duration fields within version 0 of this box.
- `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.
- `event_duration` provides the duration of event in media presentation time. In version 0-, the timescale is indicated in the `timescale` field; in version 1, the timescale of the `MovieHeaderBox` is used. The value `0xFFFF` indicates an unknown duration.
- `id`: a field identifying this instance of the message. Messages with equivalent semantics shall have the same value, 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 present and non-false, 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 `AdaptationSet@SegmentAlignment` is an integer larger than 1, the alignment described above applies only to non-overlapping Segments.

NOTE 3 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 34](#).

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 per sec. 5.10.4.3
    if ( 'emsg'.value == 2 )
    {
        string mpd_patch; // MPD patch, per sec. 5.10.4.3
    }
    if ( 'emsg'.value == 3 )
    {
        string mpd; // full MPD, per sec. 5.10.4.4
    }
}
```

**Table 34 — `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 <a href="#">5.10.4.2</a> are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause <a href="#">5.10.4.2</a> 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 <a href="#">5.10.4.3</a> are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause <a href="#">5.10.4.2</a> at least in the last segment with earliest presentation time smaller than the event time. In addition, the message includes an MPD Patch as defined in subclause <a href="#">5.10.4.3</a> in DASHEvent.mpd field within the message_data field.
3	indicates that MPD validity expiration events as defined in subclause <a href="#">5.10.4.3</a> are signalled in the Representation. MPD validity expiration is signalled in the event stream as defined in subclause <a href="#">5.10.4.2</a> at least in the last segment with earliest presentation time smaller than the event time. In addition, the message includes a complete MPD as defined in subclause <a href="#">5.10.4.4</a> 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.

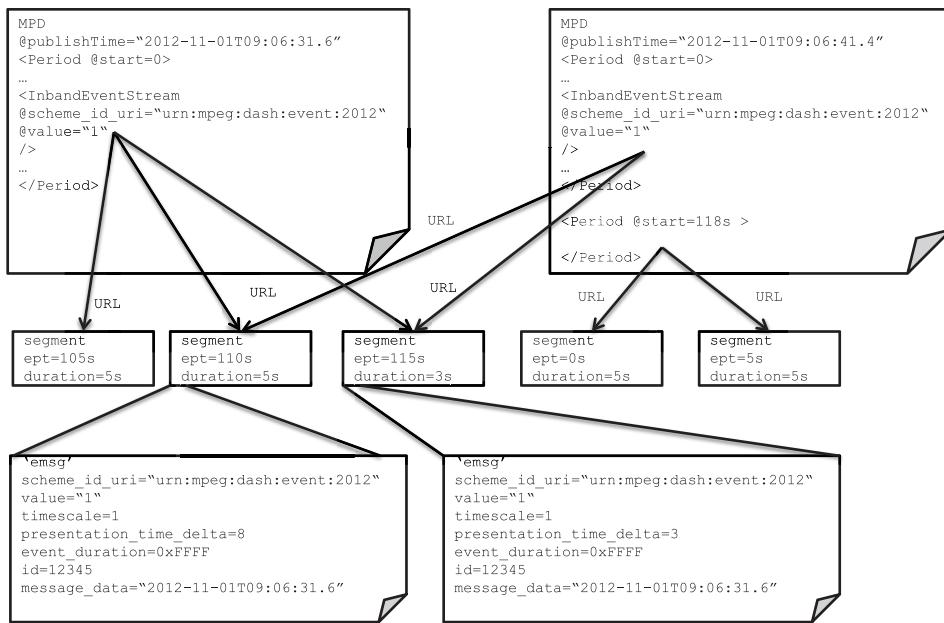
If the `scheme_id_uri` is set to "urn:mpeg:dash:event:2012" and the `value` is set to 1, then the fields in the event message box shall document the following:

- 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`) 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, Media Presentation ends at the event time. If 0xFFFF, the media presentation duration is unknown. In the case in which both `presentation_time_delta` and `event_duration` are zero, then the Media Presentation is ended.

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 `presentation_time_delta` if the earliest presentation time is different across segments.

[Figure 4](#) 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 4 — Example for MPD validity expiration to signal new Period**

#### 5.10.4.3 MPD Patch

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 XML patch compliant to XML Patch Operations framework, as defined in IETF RFC 5261.

The result of the patch application shall be parse-tree identical before any XLink resolution to the MPD that would have been retrieved at event time.

XPath selectors shall at least include a check on `MPD@publishTime`, and the last patch operation should change the value of `MPD@publishTime`.

#### 5.10.4.4 MPD Update Event

For DASH events with value 3, a complete MPD shall be included 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<sup>[18]</sup>.

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 35](#) defines the message data and the expected actions for different `@value` values when the DASH callback event is signaled as an Inband Event.

**Table 35 — Message data and expected actions for DASH call back inband event**

value	message_data[]	Action
1	Valid HTTP/HTTPS URL	An HTTP GET request is expected to be issued to a URL contained in <code>message_data[]</code> . The URL shall be a NULL-terminated string. HTTP response shall either not be provided or be provided such that it can be discarded.

##### 5.10.4.5.3 MPD event

[Table 36](#) defines the relevant parameters for a call back event signaled in the MPD.

**Table 36 — Relevant parameters for a call back event signaled in the MPD**

Attribute	Value
<code>EventStream@schemeIdUri</code>	" <code>urn:mpeg:dash:event:callback:2015</code> "
<code>EventStream@value</code>	1
<code>Event@messageData</code>	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 37](#).

**Table 37 — interpretation of `value` field of a Presentation Termination event**

<code>value</code>	<b>Description</b>
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 signaled 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 38](#). If `@value` only contains the first parameter, no whitespace is needed.

**Table 38 — Semantics of `value` for MPD chaining**

<b>Value parameters</b>	<b>Use</b>	<b>Description</b>
<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 <a href="#">C.4</a>.</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 signaled 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.

## 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 document 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 `first_offset` field and their `reference_size` field shall be identical.

### 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.

## 6.3 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 media stream access points for the ISO base media file format are defined in ISO/IEC 14496-12:—, Annex I.

### 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.

## 6.3.3 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 'traj' box shall contain a 'tfad' 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 'tfad' 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<sub>sau</sub> 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 media stream access points are defined in ISO/IEC 14496-12:—, Annex I. 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 90KHz 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 ISOBMFF-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 ISOBMFF-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 ISOBMFF files, and complete implementation of ISOBMFF is not necessary to utilize indexes in a MPEG-2 TS based client. A partial implementation would suffice, since only few ISOBMFF 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 'sisx' shall be present in the 'styp' box. The conformance requirement of the brand 'sisx' 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.

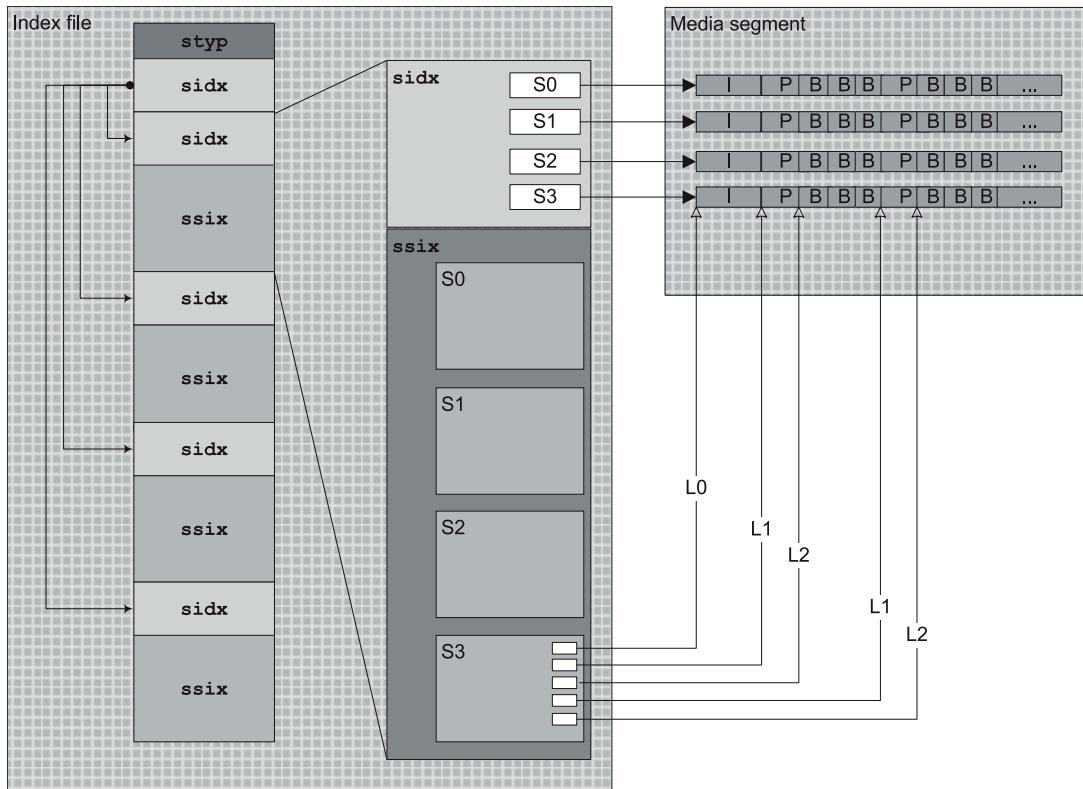
### 6.4.6.3 Representation Index Segment

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 'risx' shall be present in the 'styp' box. The conformance requirement of the brand 'risx' 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 5](#). 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 5 — 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 [6](#). 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  $T_M = T_p - T_0$ .

Media Segments should not contain any presentation time TP that is smaller than the value of the `@presentationTimeOffset`,  $T_0$ . However, if this is the case, 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 TP 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. Also in the beginning of a Period if the earliest presentation time  $T_p$  of any access unit of a Representation is not equal to  $T_0$  then the playout procedures need to be adjusted accordingly.

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 signaled. Alternatively, gaps in the segment timeline may be signaled 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  $\text{MPD@type}$  is "dynamic" and the attribute  $\text{MPD@suggestedPresentationDelay}$  is present, then the sum of value of the  $\text{MPD@availabilityStartTime}$ , the *PeriodStart* value, the presentation time within the Period of an access unit,  $T_M$ , and the value of the attribute  $\text{MPD@suggestedPresentationDelay}$  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 [Clauses 5](#) and [6](#) is instantiated in this subclause using the ISO base media file format as defined in ISO/IEC 14496-12 as Segment formats.

An ISOBMFF-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 ISOBMFF-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 ISOBMFF-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, as defined in ISO/IEC 14496-12:—, 8.16.3.

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 ('`sidx`') box is present, then it is further recommended that the track for which the Segment Index ('`sidx`') 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 a value other than '`false`' 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 ('leva') 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 ('leva') 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  $\text{PTS}(i)$  be the PTS of the  $i^{\text{th}}$  access unit in the media stream. Furthermore, let  $\text{PTS}_A(i)$  be  $\text{PTS}(i)$  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  $\text{PTS}(i)$  and  $\text{PTS}(0)$ , and therefore  $T_P(i) = [\text{PTS}_A(i) - \text{PTS}_0] * S / 90\,000$  with  $\text{PTS}_0$  typically  $\text{PTS}(0)$ . With appropriate scaling,  $\text{PTS}_0$  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 '`sidx`' box.

If a media stream contains a discontinuity, the  $\text{PTS}_A(i)$  calculation assumes relative timing is maintained. Therefore,  $\text{PTS}_A(i)$  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 ('`sidx`') box is present, then it is further recommended that the media stream for which the Segment Index ('`sidx`') 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 not set to '`false`', 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  $i=1, \dots, M$ , but for any consecutive sequence of segments with any start index  $i_S=1, \dots, M$  and any end index  $i_E=i_S, \dots, M$ .

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 subclause [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 not set to 'false' 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 subclause [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 signaling 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 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 the files referenced 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 the files referenced 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 six 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.

Three profiles are defined relying on the ISO base media FF as Segment formats. Both the ISO Base media file format On Demand profile defined in subclause [8.3](#) and the ISO Base media file format live profile defined in subclause [8.4](#) are a subset of the ISO Base media file format main profile defined in subclause [8.5](#). Two profiles are defined for MPEG-2 TS based Media Segment formats: The MPEG-2 TS simple profile defined in subclause [8.7](#) is a subset of the MPEG-2 TS main profile defined in subclause [8.6](#). 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 necessary 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 ('`ssidx`') 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 ('`ssidx`') 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 ('`ssidx`') 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' or '1';
  - `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 ('`ssidx`') 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 ('`ssidx`') 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 subclauses [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 signaling 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 signaling 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  $\text{@r} (-1)$  or closed  $\text{@r} (>=0)$ .
  - 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 ' $\text{dums}$ ' brand in the Segment Type box (' $\text{styp}$ ') as major brand and all other Segments of the Segment Sequence shall carry ' $\text{dums}$ ' brand in the Segment Type box (' $\text{styp}$ ') 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 ' $\text{emsg}$ ' 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`.

## 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. *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 media for at least a duration matching the value of `@minBufferTime` attribute 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  $\text{MPD}@availabilityStartTime + PeriodStart + T_{SAP}$  and not after  $\text{MPD}@availabilityStartTime + PeriodStart + T_{SAP} + @timeShiftBufferDepth$  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  $\text{MPD}@availabilityStartTime + PeriodStart + T_{SAP}$  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 on Segment list generation, seeking, support for trick modes and switching Representations is 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
<b>Segments</b>	1	Provides the Segment URL list.
<b>InitializationSegment</b>	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).
<b>MediaSegment</b>	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.
<b>IndexSegment</b>	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  $d = @duration / @timescale$ ,
- the MPD start time  $\text{MediaSegment}[i].startTime$  is obtained as  $(i-1)*d$ ,
- the MPD duration  $\text{MediaSegment}[i].duration$  is obtained as  $d$  unless this Segment is the last Segment in this Period, then the  $\text{MediaSegment}[i].duration$  is obtained as  $\text{PeriodEnd} - \text{MediaSegment}.StartTime[i]$ .

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  $s=1, \dots N_S$ 
  - $N=N+1$
  - $\text{MediaSegment}[N].startTime}=t[s]-o$

- MediaSegment[N].duration=d[s]
- for  $j=1, \dots, r[s]$ 
  - $N=N + 1$
  - MediaSegment[N].startTime=MediaSegment[N- 1].startTime + d[s]
  - MediaSegment[N].duration=d[s]
- else
  - the MPD duration MediaSegment[i].duration is obtained as  $d[0]$  unless this Segment is the last Segment in this Period, then the MediaSegment[i].duration is obtained as  $PeriodEnd - MediaSegment.StartTime[i]$ .

If neither the `@duration` nor the `segmentTimeline` element is given, then

- $N=1$ ,
- MediaSegment.startTime[1]=0,
- MediaSegment.duration[1]= $PeriodEnd - PeriodStart$ .

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  $i$ , 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  $i$ , MediaSegment.URL[i], is obtained by replacing the `$Time$` identifier by 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.

In addition, for services with `MPD@type='dynamic'`, the Segment availability start time  $T_{avail}[i]$  for a Segment  $i$  in a specific Period is determined as `MPD@availabilityStartTime + PeriodStart + MediaSegment[i].startTime + MediaSegment[i].duration + @availabilityTimeOffset` and the Segment availability end time is determined as `MPD@availabilityStartTime + PeriodStart + MediaSegment[i].startTime + @timeshiftBufferDepth + 2*MediaSegment[i].duration + @availabilityTimeOffset`. If the attribute `@availabilityTimeOffset` is not present, the value is assumed to be 0.

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.  $CheckTime = FetchTime + MPD@minimumUpdatePeriod$ . The *CheckTime* is defined on the MPD-documented media time axis; when the client's playback time reaches *CheckTime* - `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  $i_{max} = \max_i \{ T_{avail}[i] \leq \min(\text{CheckTime}, \text{NOW}) \}$ .

## 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 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 `foo` 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 `foo` 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 signaled 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  $BW[i] \leq rate * buffer / max\_segment\_duration$ .

**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-3/en>.

## Annex C (normative)

### MIME type registration for MPD

#### C.1 General

This annex provides the formal MIME type registration for the MPD. 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, <i>Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats</i>
— 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:PSS: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 in the same format as defined in W3C Media Fragments URI 1.0 (basic).  See W3C Media Fragments URI 1.0 (basic) for validity rules and recommended behaviour.  Optionally, prefixed comma-separated pair of numbers. See <a href="#">C.4.4</a> for validity rules and recommended behaviour.	If the parameter starts from an integer, it signifies the time since the beginning of the period indicated by the period parameter.  If the t parameter is not present, its default value is t=0 (i.e. start from the beginning of the Period).  NOTE If period parameter is not present, the default Period is the first period of the presentation.  If the parameter starts from prefix posix: 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.  The posix notation documents the absolute time in the MPD.  This notation shall only be used if MPD@availabilityStartTime is present.  This t=posix:xxx notation parameter shall not be used if a period parameter is used. In addition, at most one of the two parameters, period and t shall be present in an anchor.  A special value "now" indicates the latest available segment, i.e. "live edge".
period	String	Value of a Period parameter Period@id. If period parameter is not present, the default value of the @id attribute value of the Period with the earliest PeriodStart.
track	string	Value of a single AdaptationSet@id
group	string	Value of a single AdaptationSet@group

NOTE 1 Percent coding, defined in IETF RFC 5986, needs to be used for all reserved characters in parameter values.

NOTE 2 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

42<sup>nd</sup> second of Period1 my.mpd#t=42&period=Period1

```
my.mpd#t=42&period=Period1
```

42<sup>nd</sup> 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  $N$ .

## Annex D (normative)

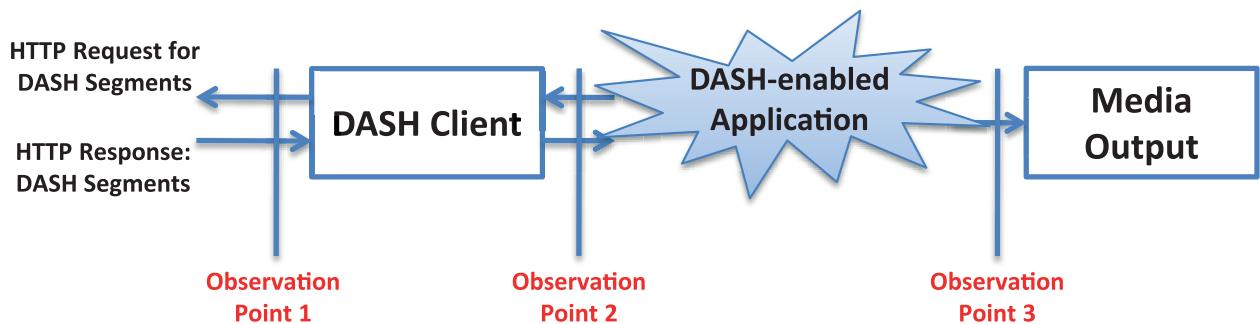
### DASH Metrics

#### 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 subclause [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**

<b>Key</b>		<b>Type</b>	<b>Description</b>
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 SYN to receiving the 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**

<b>Key</b>		<b>Type</b>	<b>Description</b>
HttpList		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.
	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"> <li>— MPD</li> <li>— XLink expansion</li> <li>— Initialization Segment</li> <li>— Index Segment</li> <li>— Media Segment</li> <li>— Bitstream Switching Segment</li> <li>— other</li> </ul>
	url	String	The original URL (before any redirects or failures)

**Table D.2** (continued)

Key		Type	Description
	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
		s	Real-Time
		d	Integer
		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.

**Table D.3 (continued)**

<b>Key</b>		<b>Type</b>	<b>Description</b>
	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 <b>Representation@id</b> identifying the switch-to Representation.
	lto	Integer	If not present, this metric concerns the Representation as a whole. If present, lto indicates the value of <b>SubRepresentation@level</b> 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**

<b>Key</b>		<b>Type</b>	<b>Description</b>
BufferLevel		List	List of buffer occupancy level measurements during playout at normal speed.
<i>Entry</i>		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**

<b>Key</b>		<b>Type</b>	<b>Description</b>
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.
<i>Entry</i>		Object	A record of a single playback period.
	start	Real-Time	Timestamp of the user action that starts the playback period.

**Table D.5** (continued)

Key		Type	Description
	mstart	Media-Time	The presentation time at which playout was requested by the user action.
	starttype	Enum	Type of user action which triggered playout: — New playout request (e.g. initial playout or seeking) — Resume from pause — Other user request (e.g. user-requested quality change) — Start of a metrics collection period (hence earlier entries in the play list not collected)
	trace	List	List of periods of continuous rendering of decoded samples.
	<i>Entry</i>		Single entry in the list.
	representationid	String	The value of the <b>Representation@id</b> of the Representation from which the samples were taken.
	subreplevel	Integer	If not present, this metrics concerns the Representation as a whole. If present, <b>subreplevel</b> indicates the greatest value of any <b>Subrepresentation@level</b> 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: — Representation switch (not relevant in case of progressive download) — rebuffering — user request — end of Period — end of content — end of a metrics collection period — failure

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.

## 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. Still it is expected that this annex will be deprecated in future editions of 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**

<code>\$&lt;Identifier&gt;\$</code>	Substitution parameter
<code>\$\$</code>	Is an escape sequence, i.e. <code>\$\$</code> is replaced with a single <code>\$</code>
<code>\$base\$</code>	The identifier shall be substituted by the <code>scheme ":" hier-part</code> of the original URL as defined in IETF RFC 3986.

**Table E.1** (continued)

\$<Identifier>\$	<b>Substitution parameter</b>
\$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 \$query\$ identifier in the template string if that character is not the "?" character, or, otherwise, the separator character immediately following the \$query\$ 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 'first-byte-pos' of 'byte-range-spec' 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 'last-byte-pos' of 'byte-range-spec' of IETF RFC 7231:2014, 2.1, if this request would be executed using a partial GET request.

### E.3 Examples

<b>Original URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926">http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926</a></u>
<b>Byte Range</b>	1876-23456
<b>BaseUrl@byteRange</b>	\$base\$/range/\$first\$-\$last\$
<b>Byte range URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4 range/1876-23456?token=8787r08f2gf087g28gf926">http://cdn.example.com/movies/134532/audio/en/aac64.mp4 range/1876-23456?token=8787r08f2gf087g28gf926</a></u>

<b>Original URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4">http://cdn.example.com/movies/134532/audio/en/aac64.mp4</a></u>
<b>Byte Range</b>	1876-23456
<b>BaseUrl@byteRange</b>	\$base\$\$query\$&range=\$first\$-\$last\$
<b>Byte range URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4?range=1876-23456">http://cdn.example.com/movies/134532/audio/en/aac64.mp4?range=1876-23456</a></u>

<b>Original URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926">http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926</a></u>
<b>Byte Range</b>	1876-23456
<b>BaseUrl@byteRange</b>	\$base\$\$query\$&range=\$first\$-\$last\$
<b>Byte range URL</b>	<u><a href="http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926&amp;range=1876-23456">http://cdn.example.com/movies/134532/audio/en/aac64.mp4?token=8787r08f2gf087g28gf926&amp;range=1876-23456</a></u>

## 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 ('sidx') or Subsegment Index ('ssix'), 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 media 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:—, Annex I 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

#### 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"?>
<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/50000/$Time$.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"?>
<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"/>
    </AdaptationSet>
  </Period>
</MPD>
```

```

<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 subclause [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="$RepresentationID$.sidx"
                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"?>
<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="mvcl.760028">

```

```

<Role schemeIdUri="urn:mpeg:dash:stereoid:2011" value="l0"/>
<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"?>
<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"?>
<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" />
    </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"?>
<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=kljklksdfiowek</
        drm:License>
      </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"?>
<!-- Attention: this is not a complete MPD and thus will not validate against the MPD Schema -->
<MPD>
  <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 `@schemeIdU  
ri="urn:org:example:xscte35"`, 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). 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">
    <xsc3te35:signal>
      <id name="segmentEventId" value="1" />
      <time name="start" value="10" />
      <time name="end" value="20" />
    </xsc3te35:signal>
  </metadata>

```

```

<?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="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>

```

## 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"?>
<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>
<!--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%2f0EEFB-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"/>
        <SegmentTemplate timescale="48000" duration="94175"
media="BBB_32k_$Number$.mp4" startNumber="1" initialization="BBB_32k_init.mp4"/>
    </Representation>
</AdaptationSet>
</Period>
<Period xlink:href="remote.period" xlink:actuate="onLoad" 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>
```

## Contents of remote.period element:

```

<?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" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT704S" profiles="urn:mpeg:dash:profile:isoff-live:2011">

<Period id="1" duration="PT110S" start="PT250S" xmlns="urn:mpeg:dash:schema:mpd:2011">
    <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>
</MPD>

```

**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"?>
<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 -->
    <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, AdaptationSet) 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" mediaPresenta-
tionDuration="PT3256S" minBufferTime="PT1.500000S" profiles="urn:mpeg:dash:profile:isoff-
live:2011">
    <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>
            <SegmentTemplate startNumber="1" timescale="1000" duration="3840" media="$Rep-
resentationID$/${Number%06d}.m4s" initialization="$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-
live:2011">

    <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="50/0" codecs="avc3.64001f" maxPlayoutRate="1" scanType="progressive">
                    <SegmentTemplate initialization="data:video/mp4;base64,AAAAAHGZ0eXBpc282AA
AAAWF2YzFpc29tZGFzaAAAAldtb292...AAAAAAAAAAAAAQc3RjbwAAAAAAA"/>
                </Representation>
                <Representation id="192x108p6_25" bandwidth="31368" width="192" height="108" frameRate="25/0" codecs="avc3.42c015" maxPlayoutRate="1" scanType="progressive">
                    <SegmentTemplate initialization="data:video/mp4;base64,AAAAAHGZ0eXBpc282AA
AAAWF2YzFpc29tZGFzaAAAAlNtb292...AAAAAAAAABzdHNjAAAAAAAAAAAAUc3RzegAAAAAAAAAAAAAAA
ABBzdGNvAAAAAAA=/>
                </Representation>
            </AdaptationSet>
        </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**

<code>EssentialProperty@value</code> or <code>SupplementalProperty@value</code> 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:
<b>Key</b>		
M=Mandatory, O=Optional		

**Table H.1** (continued)

<b>EssentialProperty@value or SupplementalProperty@value parameter</b>	<b>Use</b>	<b>Description</b>
		<ul style="list-style-type: none"> <li>— There shall be at least one descriptor providing a value for the <code>total_width</code> parameter.</li> <li>— 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>.</li> <li>— 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.</li> <li>— 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>.</li> </ul> <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"> <li>— There shall be at least one descriptor providing a value for the <code>total_height</code> parameter.</li> <li>— 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>.</li> <li>— 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.</li> <li>— 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>.</li> </ul> <p>When the value <code>total_height</code> is present, the value <code>total_width</code> shall be present.</p>
<code>spatial_set_id</code>	0	<p>optional non-negative integer in decimal representation providing an identifier for a group of Spatial Object.</p> <p>When not present, the Spatial Object associated to this descriptor does not belong to any spatial set and no spatial set information is given.</p> <p>When the value of <code>spatial_set_id</code> is present, the value of <code>total_width</code> and <code>total_height</code> shall be present.</p>

**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**

<code>EssentialProperty@value</code> or <code>SupplementalProperty@value</code> pa- rameter	Use	Description
<code>source_id</code>	M	non-negative integer in decimal representation providing the identifier for the source of the content
<code>coordinate_id</code>	M	specifies the <code>@id</code> attribute of the Representation that provides the 2D coordinates of the Spatial Object as timed metadata track according to ISO/IEC 23001-10.
<code>spatial_set_id</code>	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.
<b>Key</b>		
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 following mapping between the '`2dcc`' sample parameters and the SRD parameters shall apply in order to determine the coordinates and size of a Spatial Object whose coordinates and size is provided as '`2dcc`' metadata Representation.

2D Cartesian Coordinates Sample (ISO/IEC 23001-10)	SRD parameters
<code>top_left_x</code>	<code>object_x</code>
<code>top_left_y</code>	<code>object_y</code>
<code>width</code>	<code>object_width</code>
<code>height</code>	<code>object_height</code>
<code>reference_width</code>	<code>total_width</code>
<code>reference_height</code>	<code>total_height</code>

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 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"
  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">
      <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
      <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,3,3,3,3"/>
      <Representation 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">
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,1,1,1,1,3,3"/>
  <Representation 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.3.2 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"
  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">

```

```
<Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
<SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,2,2,2,2"/>
<Representation 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 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 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">
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014" value="0,0,0,1,1,2,2"/>
  <Representation 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 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 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>
<!-- 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.3.3 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"
  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"/>
    </AdaptationSet>
  </Period>
</MPD>
```

```
<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 signaled 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 signaled 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 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 Annex. The complete normative URL Query Info schema is provided below. In case of any inconsistencies the schema below takes precedence both over the XML syntax snippets provided in this clause and all prose text in this part of the standard.

```
<?xml version="1.0"?>
<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:das:schema:urlparam:2014" elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>

<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"/>
  <xs:attribute name="queryString" type="xs:string"/>
  <xs:attribute ref="xlink:href"/>
</xs:complexType>
```

```

<xs:attribute ref="xlink:actuate" default="onRequest"/>
<xs:anyAttribute namespace="#other" processContents="lax"/>
</xs:complexType>

<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="" />
      <xs:attribute name="sameOriginOnly" type="xs:boolean" default="false"/>
    </xs:extension>
  </xs:complexContent>
</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>	O (string)	<p>provides URL parameters template information</p> <p>This string shall contain one or more \$&lt;ParamIdentifier&gt;\$ template identifiers, as listed in <a href="#">Table I.2</a>. These template identifiers are to be replaced to build a query string (see <a href="#">I.2.3</a>). If \$&lt;ParamIdentifier&gt;\$ is not in <a href="#">Table I.2</a>, it will be replaced with an empty string. If the template has an opening \$ without a matching closing \$, the result is undefined, and the client will act as if it did not understand the <code>EssentialProperty</code>'s scheme.</p> <p>When selection of URL parameters is enabled through the use of \$query: param\$ template identifiers, URL parameters shall be defined as name=value pairs separated by &amp;, as defined by W3C HTML 4.01 Specification (section on Forms#Form submission).</p>
<code>@useMPDUrlQuery</code>	O (bool) default: <code>false</code>	<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><b>NOTE</b> Simple parameter signaling can be used ("a=X&amp;b=Y"), as well as scheme-dependent signaling ("a=\$urn:XYZ\$&amp;b=\$urn:ABC\$").</p> <p>When scheme-dependent signaling is used, the scheme shall be inserted between two enclosing \$ characters. See <a href="#">I.2.3.3</a> 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. See <a href="#">I.2.3.3</a> for further details.</p>
<code>@queryString</code>	O (string)	<p>provides a query string to be used in the construction of media segment URLs.</p> <p><b>NOTE</b> Simple parameter signaling can be used ("a=X&amp;b=Y"), as well as scheme-dependent signaling ("a=\$urn:XYZ\$&amp;b=\$urn:ABC\$").</p> <p>When scheme-dependent signaling is used, the scheme shall be inserted between two enclosing \$ characters. See <a href="#">I.2.3.3</a> for further details.</p>
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table I.1 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@xlink:href	O	specifies a reference to a remote <code>UrlQueryInfo</code> element
@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.

**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 @.

**Table I.2 — Parameter identifiers**

\$<ParamIdentifier>\$	Substitution parameter
\$\$	Is an escape sequence, i.e. " \$\$ " is replaced with a single "\$".
\$querypart\$	This identifier is substituted with the query part of the computed query string (referred to as <i>initialQueryString</i> in <a href="#">I.2.3</a> ). 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).  The query part starts after the "?" sign and lasts until the "#" sign
\$query:<param>\$	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 <a href="#">I.2.3</a> ). 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.  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)

### I.2.2.3 XML Syntax

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Url query information -->
<xs:complexType name="UrlQueryInfoType">
  <xs:attribute name="queryTemplate" type="xs:string"/>
  <xs:attribute name="useMPDUrlQuery" type="xs:boolean"/>
  <xs:attribute name="queryString" type="xs:string"/>
  <xs:attribute name="xlink:href"/>
  <xs:attribute name="xlink:actuate" default="onRequest"/>
</xs:complexType>

<xs:element name="UrlQueryInfo" type="UrlQueryInfoType"/>
```

## I.2.3 Modified template-based segment URL construction, according to `UrlQueryInfo` element

### I.2.3.1 General

When signalized 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 subclause [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 signaling may be used (@queryString="a=X&b=Y"), as well as scheme-dependent signaling (@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 signaling 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>

```
<?xml version="1.0" encoding="utf-8"?>
<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:scheme:urlparam:2014">
    <up:UrlQueryInfo queryTemplate="$querypart$" 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>
```

- 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"*

- 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_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_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](http://www.example.com/dash/video_3_3000000bps.mp4?token=1234&ip=1.2.3.4)

#### I.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:

```

<up:UrlQueryInfo xmlns:up="urn:mpeg:dash:schema:urlparam:2014"
queryTemplate="$querypart$" queryString="param=justintimecomputedvalue"/>

<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"/>
  </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>
```

- 1) Computation of an initial query string (computed on request, according to `xlink:actuate`):

*initialQueryString="param=justintimecomputedvalue"*

- 2) Computation of an 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_1_3000000bps.mp4?param=justintimecomputedvalue)

[http://www.example.com/dash/video\\_2\\_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](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.

```
<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"/>
  <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>
```

- 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>

```
<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>
```

- 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
<code>ExtendedUrlInfoType</code>		provides information for derivation of parameter string. This is an extension of <code>UrlQueryInfoType</code> element defined in <a href="#">Table I.1</a> .
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table I.3 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@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> <ul style="list-style-type: none"> <li>1) "segment" (all segment requests),</li> <li>2) "xlink" (all XLink resolution requests),</li> <li>3) "mpd" (all MPD requests),</li> <li>4) "callback" (all requests triggered by DASH callback events),</li> <li>5) "chaining" (requests for chained-to MPDs),</li> <li>6) "fallback" (requests for the alternative MPDs).</li> </ul> <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 <b>ExtUrlQueryInfo</b>) or HTTP headers (for <b>ExtHttpHeaderInfo</b>)</p>
@headerParamSource	OD (default: "seg- ment")	<p>specifies HTTP responses from which HTTP header values, identified by the template <code>\$header:header-name\$</code>, should be extracted from. Value is a white-spaced concatenated list of the following keys :</p> <ul style="list-style-type: none"> <li>1) "segment" (all segment requests),</li> <li>2) "xlink" (all XLink resolution requests),</li> <li>3) "mpd" (all MPD requests),</li> <li>4) "callback" (all requests triggered by DASH callback events).</li> </ul> <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 <code>\$header:&lt;header-name&gt;\$</code> identifier, and (b) neither @useMPDUrlQuery nor @queryString attribute shall be present.</p>
<b>Key</b>		
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 <b>bold</b> ; attributes are non-bold and preceded with an @.		

**Table I.3 (continued)**

<b>Element or Attribute Name</b>	<b>Use</b>	<b>Description</b>
@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>

**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 @.

The following identifiers are defined, in addition to the ones defined in [Table I.2](#)

**Table I.4 — Parameter identifiers**

\$<ParamIdentifier>\$	<b>Substitution parameter</b>
\$header:<header-name>\$	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: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="" />
      <xs:attribute name="sameOriginOnly" type="xs:boolean" default="false"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ExtUrlQueryInfo" type="ExtendedUrlInfoType"/>
<xs:element name="ExtHttpHeaderInfo" type="ExtendedUrlInfoType"/>

```

### 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 the following table:

**SupplementalProperty@value attributes resolutionSwitching:2016**

<b>SupplementalProperty@value parameter</b>	<b>Use</b>	<b>Description</b>
switchableTo	M	specifies all Representations with representation switch points specified with <b>RandomAccess</b> with <code>@type</code> 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 <code>@id</code> 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 <code>switchableTo</code> .  NOTE The Representation contains a media stream conforming to <code>@mimeType</code> and additionally one or more media streams conforming to <code>switchingMimeType</code> that are only intended for switching to another representation, as indicated by <code>switchableTo</code> .
<b>Key</b>		
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.

## 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] SECURITY ASSERTION MARKUP LANGUAGE OASIS, (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, <http://www.iab.net/media/file/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*

---

---

---

**ICS 35.040.40**

Price based on 225 pages