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

**Part 8:
Session-based DASH operations**

*Technologies de l'information — Diffusion adaptative dynamique sur
HTTP (DASH) —*

Partie 8: Opérations de DASH basées sur la session



IECNORM.COM : Click to view the full PDF of ISO/IEC 23009-8:2022



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

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

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

Published in Switzerland

Contents

Page

Foreword.....	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	1
3 Terms, definitions, abbreviated terms and notations.....	1
3.1 Terms and definitions.....	1
3.2 Abbreviated terms.....	2
3.3 Notation.....	2
4 General overview.....	3
4.1 Sessions and session parameters.....	3
4.2 General architecture.....	3
4.3 Timeline and orderline addressing.....	4
5 MPD signalling for SBD.....	4
5.1 General.....	4
5.2 SBD descriptor.....	5
5.2.1 General.....	5
5.2.2 Semantics.....	5
5.2.3 XML schema.....	5
6 SBD document format.....	6
6.1 Overview.....	6
6.2 KeyValue Object.....	6
6.2.1 Semantics.....	6
6.2.2 JSON schema.....	9
7 Client operation.....	13
7.1 Normative requirements.....	13
7.2 Processing model (informative).....	13
7.2.1 Building SBD timeline/orderline table.....	13
7.2.2 Parameter value derivation.....	14
7.2.3 Segment request.....	14
8 Examples.....	14
8.1 General.....	14
8.2 MPD.....	14
8.3 Session-based description document.....	15
8.3.1 Example 1: Timeline.....	15
8.3.2 Example 2: Orderline.....	15
8.4 Operation.....	16
9 Example applications – Forensic watermarking.....	16
Annex A SBD MIME type.....	18
Bibliography.....	19

Foreword

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

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

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

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

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

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

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

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

Introduction

Dynamic adaptive streaming over HTTP (DASH) enables 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, the ISO/IEC 23009 series focuses not on client or server procedures but on the data formats used to provide a DASH Media Presentation.

This document provides methods, interfaces and data for session-based operations to be used with the DASH standard. Session-based operations allows customization of requested segment URLs based on the information provided for a specific streaming session.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23009-8:2022

IECNORM.COM : Click to view the full PDF of ISO/IEC 23009-8:2022

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

Part 8: Session-based DASH operations

1 Scope

This document specifies the format of the Session-Based Description document and the MPD's extension to be used in session-based operations with MPEG DASH (ISO/IEC 23009-1).

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 23009-1:2019, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

IETF RFC 7230, *Hypertext Transfer Protocol — HTTP/1.1*, June 2014

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, January 2005

IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*, December 2017

3 Terms, definitions, abbreviated terms and notations

3.1 Terms and definitions

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

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

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

3.1.1

SBDStart

starting time of the session-based description

3.1.2

session

sequence of DASH client actions resulting in a unique playback of a media asset

3.1.3

session-based description

collection of timed key-value pairs providing a value given a time range within the session

3.1.4

session-based client

client capable of parsing SBD and adding queries to the DASH segment requests accordingly

3.1.5

sidecar document

document that stores data which is not supported by the format of the main document

3.2 Abbreviated terms

DASH	dynamic adaptive streaming over HTTP, as specified in ISO/IEC 23009-1
HTTP	hypertext transfer protocol, as specified in RFC 7230 (HTTP/1.1).
HTTPS	secure version of the hypertext transfer protocol
ISO-BMFF	ISO Base Media File Format, as specified in ISO/IEC 14496-12
MIME	multipurpose internet mail extensions, as specified in RFC 6838
MPD	media presentation description, as specified in ISO/IEC 23009-1
SBD	session-based description, as defined in this document
URI	uniform resource identifier
URL	uniform resource locator
URN	uniform resource name

3.3 Notation

This document follows the notation used in ISO/IEC 23009-1.

The following naming conventions apply in this document:

- Elements in an XML document are identified by an upper-case first letter and in boldface 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, PascalCase is typically used, e.g. **ImportantElement**. Elements may be present either exactly once, or the minimum and maximum occurrence as defined by `<minOccurs>` ... `<maxOccurs>`.
- Attributes in an XML document are identified by a lower-case first letter and 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, camelCase is typically used after the first word, e.g. `@veryImportantAttribute`. Attributes may have assigned a status in the XML as mandatory (M), optional (O), optional with a 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.
- JSON items are identified by a lower-case first letter. If an item consists of two or more combined words, camel-casing is typically used.
- Variables defined in the context of this document are specifically highlighted with *italics*, e.g., *InternalVariable*.
- 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.

4 General overview

4.1 Sessions and session parameters

While the DASH viewing session per client is a unique interaction, the MPD is often cached on the Content Delivery Network (CDN) and shared by multiple viewing sessions of the same asset. The session-based description mechanism allows adding session-specific elements to segment URLs without generating unique per-session MPDs.

Session parameter strings are strings pertaining to a specific time range on the timeline. These strings are parsed by the Session-based description client and are translated into parameters to be added to HTTP segment request(s) issued by the DASH client.

A Session-Based description Document (SBD) includes the time ranges and the corresponding strings, along with additional metadata. The SBD is referenced in the MPD by its URL. This clause describes the overall SBD data model.

For example, let parameters "p1" and "p2" to be expected carrying information in the query part of the segment request URL. For time range 00:00:00-00:00:42, the "p1" value is "foo" and the "p2" value is 42. For the time range 00:00:42-00:04:20, "p1" is "bar" and "p2" is 420. In the case of two-second segments, the first 21 segments would include "p1=foo&p2=42" as a part of their URL query parameters, and the following 109 segments would include "p1=bar&p2=420" as a part of the segment URL in the HTTP GET request.

The information carried in SBD can be conceptually presented in a table with each row corresponding to a time range, and each column corresponding to one named parameter. [Table 1](#) demonstrates the conceptual layout of SBD carrying the information for the above example.

Table 1 — Conceptual sidcar document layout

Start time	p1	p2
00:00:00	foo	42
00:00:42	bar	420
00:04:20	null	null

4.2 General architecture

[Figure 1](#) shows the general architecture for session-based DASH operations.

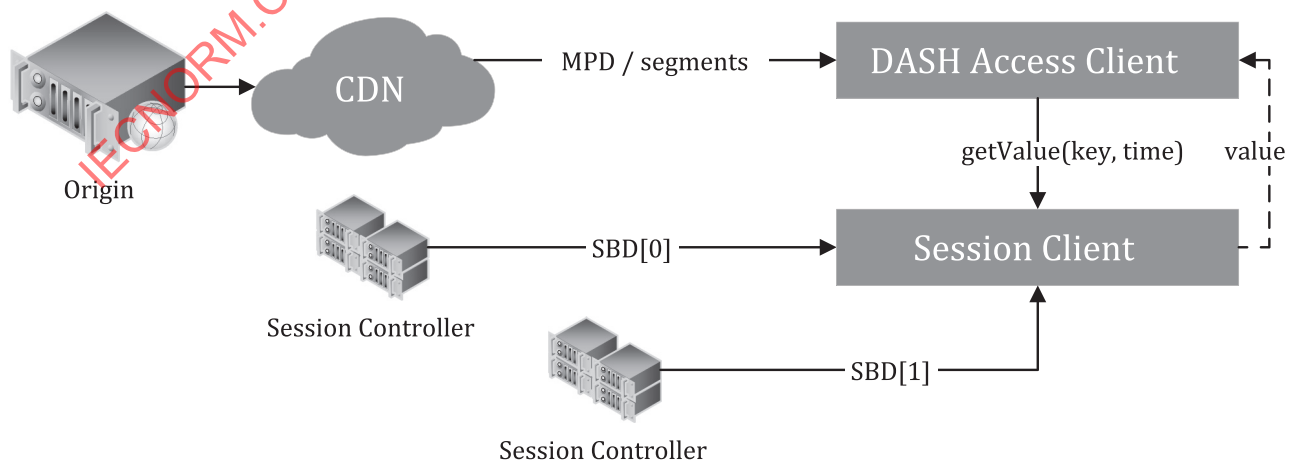


Figure 1 — High-level architecture of session-based DASH operations.

In this figure, the media content is described by MPD, and the corresponding media segments are delivered in the media delivery path. MPD also has a reference to an external document, Session-Based Description (SBD) document, which defines the variables and their values for the current session. SBD is parsed and interpreted by an SBD client.

This SBD client is conceptually a simple timed key-value store. It accepts requests for a named variable value given a time range and returns a string value.

The DASH client uses the variables and the associated values provided in the SBD to derive an added part to the segment URL's query when the segment is requested using HTTP GET.

Note that there may be multiple session controllers controlling multiple aspects of the session, each of which providing one SBD document (e.g., one controlling forensic watermarking and another controlling access tokens).

Note This document does not preclude the use of newer versions of HTTP, such as HTTP/2, as this document only relies on the existence of the GET method.

4.3 Timeline and orderline addressing

The SBD document may conceptually include one or more tables, each containing the key-value pair assignments for different time intervals, as shown in [Table 2](#).

Table 2 — An example of SBD timing table

Start time	k_1	k_2	...	k_n
t_1	$v_{1,1}$	$v_{1,2}$	...	$v_{1,n}$
t_2	$v_{2,1}$	$v_{2,2}$	...	$v_{2,n}$
...	...	...	...	...
t_m	$v_{m,1}$	$v_{m,2}$	...	$v_{m,n}$

In [Table 2](#), t_i indicates the moment i in the media timeline, and $(k_j, v_{i,j})$ are the key-value pair j for time interval (t_i, t_{i+1}) .

Alternatively, The SBD document may conceptually include one or more tables, each containing the key-value pair information for segments order as shown in [Table 3](#).

Table 3 — An example of SBD order table

Order	k_1	k_2	...	k_n
n_1	$v_{1,1}$	$v_{1,2}$	...	$v_{1,n}$
n_2	$v_{2,1}$	$v_{2,2}$	...	$v_{2,n}$
...	...	...	...	...
n_m	$v_{m,1}$	$v_{m,2}$	...	$v_{m,n}$

In [Table 3](#), n_i indicates segment i and $(k_j, v_{i,j})$ are the key-value pair j for all segments from segment i to segment $i + 1$ but not including segment $i + 1$.

5 MPD signalling for SBD

5.1 General

To signal the session-based information in the DASH manifest, a descriptor is defined in this clause. This descriptor shall be used in MPD, ISO/IEC 23009-1.

5.2 SBD descriptor

5.2.1 General

An SBD descriptor is an **EssentialProperty** descriptor with URN “urn:mpeg:dash:sbd:2020” as its @schemeIdUri and may contain one or more **KeyList** elements, as is shown in [Table 4](#).

5.2.2 Semantics

Table 4 — MPD EssentialProperty Descriptor attributes for session-based DASH

Element or Attribute Name	Use	Description
EssentialProperty		instantiation of EssentialProperty (defined in ISO/IEC 23009-1) for session-based DASH operations.
@schemeIdUri	M (string)	shall be set to “urn:mpeg:dash:sbd:2020”.
@value	M (string)	URL of the SBD document for this session
@template	O	template for applying to the key-value pair found in SBD document. When the value of a Key @name in the @template is found in the SBD document, its corresponding key-value pair of SBD document shall replace the string between unescaped ‘\$’ characters in the @template. The @template value shall have no whitespace characters. If absent, Key @name and its corresponding value in the SBD document, separated by ‘=’, shall be added to the end of the query, where each consecutive key-value pairs are separated by ‘&’. The order of key-value pairs in the query is defined by the order of Key elements in this descriptor.
Key	1..N	a key name to be found in SBD document and its default value. If absent, all keys and corresponding values in the corresponding keyList of the SBD document shall be added to the (sub) segment request URL query.
@name	M	Name of the key of the SBD document to be added to the queries for this session after processing described by @template. This value shall be a string without whitespaces, start with a letter, and contain only unreserved characters per RFC 3986. If @name value does not appear in the keyList of SBD document, the @name value-@defaultValue value pair shall be used in the template or in the absence of the template.
@defaultValue	OD (‘null’)	default value in key-value pair if @name value not found in SBD document or if there is no value defined for a requested time range or segment number in the SBD document.
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 @.		

The above **EssentialProperty** element may appear at one or more levels for the MPD hierarchical data model, one or more times.

NOTE Adding parameterized queries to SBD URL can be achieved by using ISO/IEC 23009-1:2019, Annex I.

5.2.3 XML schema

```

<!--Descriptor schema -->
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sbd="urn:mpeg:dash:sbd:2020" targetNamespace="urn:mpeg:dash:sbd:2020" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- Key -->
  <xs:complexType name="Key">
    <xs:attribute name="name" type="xs:Name"/>
    <xs:attribute name="defaultValue" type="StringNoWhitespaceType"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:attribute name="template" type="StringNoWhitespaceType"/>
</xs:schema>

```

A electronic copy of the XML schema is provided at <https://standards.iso.org/iso-iec/23009/-8/ed-1/en>.

6 SBD document format

6.1 Overview

SBD document is a sidecar document using the JSON format and shall conform to IETF RFC 8259. It consists of a sequence of one or more Key-Value objects, which are defined in subclause 6.2.

This Clause defines the syntax of the document and provides a normative schema in subclause 6.2.2. This schema uses the JSON Schema notation available at <https://tools.ietf.org/html/draft-bhutton-json-schema-00>.

The SBD document has MIME type application/dash-sbd+json normatively defined in Annex A.

6.2 Key-Value Object

6.2.1 Semantics

The **Key-Value** object defines one timing or order table in the SBD document, as shown in Table 5. The `timeline` and `orderline` are array of objects. The semantics of these objects are described in Table 6 and Table 7, respectively.

Table 5 — Key-Value semantics

Element or Attribute Name	Use	Description
Key-Value		provides information on a single key-value pair in the query string.
Key: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. JSON objects are bold ; name-value pairs are non-bold.		

Table 5 (continued)

Element or Attribute Name	Use	Description
keylist	M (array of strings)	list of unique key names of key-value properties described in this element. Each key name shall be a string without whitespaces, start with a letter, and contain only unreserved characters per RFC 3986.
comment	O (string)	a human-readable comment about this KeyValue
timescale	OD (integer) Default: 1	timescale in ticks per second for the @s and @d attributes. Default value is 1 (i.e., by default, v@d is in seconds).
type	OD (string) Default: "static"	possible values: "dynamic", "static" "static": the SBD document and its values shall not change during the streaming session. "dynamic": the SBD document and its value may be updated every ttl seconds.
ttl	CM (integer)	the minimum guaranteed duration in seconds that the dynamic SBD is not updated. This attribute shall not be present when "type": "static".
starttime	O (integer)	anchor for the starting time of SBD (SBDStart) in NTP time. This attribute shall not be present for MPD@type='static'. If absent, SBDStart is obtained as the following: — If the SBD descriptor is included in the MPD element, the anchor is MPD@availabilityStartTime. — If the SBD descriptor is included in a Period's element, the anchor is the PeriodStart.
duration	O (integer)	duration (in seconds) to which SBD is applied. If not present, Timeline or Orderline elements will define the duration. If present, the Timeline or Orderline outside of this duration is ignored.
loop	OD (boolean) Default: 'false'	for value "true", after running out of all Timeline or Orderline elements, the new set of Timeline or Orderline element(s) is added by repeating the same element until the session ends, i.e., all the remaining segment requests of the session shall have query according to this element. for value "false", if the Timeline or Orderline element(s) ends before the end of the session, no query is added to the remaining segment requests in the session.
Timeline	array of objects	values with their time values. Either Timeline or Orderline shall be present but not both.
Orderline	array of objects	values with their order values. Either Timeline or Orderline shall be present but not both. This item shall not be present when MPD@type='dynamic'.
Key: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. JSON objects are bold ; name-value pairs are non-bold.		

Table 6 — timeline element semantics

Element or Attribute Name	Use	Description
v	CM (array of string)	list of values in order matching their names in <code>keylist</code> , separated by white space. Default value is specified in Key @ <code>defaultValue</code> attribute. Each value shall contain only unrestricted characters per RFC 3986 Section 2.2. Namely, the permitted characters are uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde.
s	CM (integer)	start of the time range to which the value applies. The anchor of this time is the <code>SBDStart</code> parameter. <code>SBDStart</code> is obtained as defined in <code>starttime</code> semantics. Mandatory unless <code>v@n</code> addressing is used. Only one of <code>s</code> or <code>n</code> attributes may be present, but not both.
n	CM (integer)	starting number of the (sub)segment to which the value applies, i.e. for " <code>n</code> ": " <code>5</code> ", the first 4 (sub)segments are skipped and the fifth (sub)segment is the starting (sub)segment. If absent, the starting (sub)segment is the first (sub)segment in corresponding MPD or Period element, depending on which element the SBD descriptor is included. Only one of <code>s</code> or <code>n</code> attributes may be present, but not both.
r	CM (integer) Default: 0	specifies the number of consecutive (sub)segments in a time range. Default: 0 (no repetitions).
d	CM (integer)	duration of time range, unless the start of next range coincides with the start of the range described in the immediately following <code>v</code> element. If this is the last element, the lack of <code>@d</code> indicates the applicability of the segment to the remaining period.
Key: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. JSON: objects are bold ; name-value pairs are non-bold.		

Table 7 — orderline element semantics

Element or Attribute Name	Use	Description
v	CM (array of string)	list of values in order matching their names in <code>keylist</code> , separated by white space. The default value is specified in Key @ <code>defaultValue</code> attribute. Each value shall contain only unrestricted characters per RFC 3986 Section 2.2. Namely, the permitted characters are uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde.
n	O (integer)	starting number of the (sub)segment to which the value applies, i.e., for " <code>n</code> ": " <code>5</code> ", the first 4 (sub)segments are skipped and the fifth (sub)segment is the starting (sub)segment. If absent, the starting (sub)segment is the first (sub)segment in corresponding MPD or Period element, depending on which element the SBD descriptor is included. Only one of <code>s</code> or <code>n</code> attributes may be present, but not both.
Key: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. JSON: objects are bold ; name-value pairs are non-bold.		

Table 7 (continued)

Element or Attribute Name	Use	Description
<code>r</code>	0 (integer)	number of additional (sub)segments in this <code>o</code> element using the key-value pairs, e.g. " <code>r</code> ": "1", two (sub)segments are using the key-value pairs of this element. If not present, only one (sub)segment in this element is using the key-value pairs defined in this element.
Key: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. JSON: objects are bold ; name-value pairs are non-bold.		

6.2.2 JSON schema

```
{
  "$id": "https://github.com/MPEGGroup/DASHPart8/blob/master/keyvalue-schema.json",
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "The schema for the array of keyvalue objects, as defined in ISO/IEC 23009-8 Session-based DASH operations",
  "type": "array",
  "items": {
    "type": "object",
    "required": [
      "keyList"
    ],
    "oneOf": [
      {
        "required": [
          "timeline"
        ]
      },
      {
        "required": [
          "orderline"
        ]
      }
    ]
  }
},
```

```
"properties": {
  "keyList": {
    "type": "array",
    "item": "string"
  },
  "comment": {
    "type": "string"
  },
  "timescale": {
    "type": "integer",
    "minimum": 0,
    "default": 1
  },
  "type": {
    "type": "string",
    "enum": [
      "static",
      "dynamic"
    ],
    "default": "static"
  },
  "startTime": {
```

IECNORM.COM : Click to view the full PDF of ISO/IEC 23009-8:2022


```

    "type": "integer",
    "minimum": 0
  },
  "duration": {
    "type": "integer",
    "minimum": 0
  },
  "loop": {
    "type": "boolean",
    "default": "false"
  },
  "timeline": {
    "type": "array",
    "item": {
      "type": "object",
      "required": [
        "v"
      ],
      "properties": {
        "v": {
          "type": "array",
          "item": "string"
        },
        "s": {
          "type": "integer",
          "minimum": 0
        },
        "n": {
          "type": "integer",
          "minimum": 0
        },
        "r": {
          "type": "integer",

```

```
        "minimum": -1,
        "default": 0
    },
    "d": {
        "type": "integer",
        "minimum": 0
    }
}
},
"orderline": {
    "type": "array",
    "item": {
        "type": "object",
        "required": [
            "v"
        ],
        "properties": {
            "v": {
                "type": "array",
                "item": "string"
            },
            "n": {
                "type": "integer",
                "minimum": 0
            },
            "r": {
                "type": "integer",
                "minimum": -1,
                "default": 0
            }
        }
    }
}
},
},
```

```

    "if": {
      "properties": {
        "type": {
          "const": "dynamic"
        }
      }
    },
    "then": {
      "properties": {
        "ttl": {
          "type": "integer",
          "minimum": 0
        }
      }
    }
  }
}

```

A electronic copy of the JSON schema is provided at <https://standards.iso.org/iso-iec/23009/-8/ed-1/en>.

7 Client operation

7.1 Normative requirements

The client shall recognize and process the SBD descriptor's @schemeIdUri according to [Clause 6](#).

The client shall download the SBD document from the location indicated by the SBD descriptor's @value.

The client shall parse the SBD document JSON object and build the timeline/orderline table according to [Clause 7](#).

The client shall find the matching row in the timeline/orderline table of a requesting (sub)segment according to [Clause 7](#).

The client shall find the corresponding value for each key listed in the SBD descriptor for the matching row, and build the query with the order of keys defined in the SBD descriptor according to [Clauses 6 and 7](#).

The client shall calculate the query for the matching row, build the query string, and add it to the segment request according to [Clause 6](#).

7.2 Processing model (informative)

7.2.1 Building SBD timeline/orderline table

Input: MPD

Output: none

- a) The DASH client parses the SBD descriptor and passes the @value and SBDStart to the SBD client.
- b) The SBD client downloads KeyValue object from the URL address indicated in @value.
- c) The SBD client builds the timeline/orderline table.

7.2.2 Parameter value derivation

Inputs: @name, (sub)segment's EPT/(sub)segment number

Output: string q

- a) The SBD client calculates the earliest presentation time (EPT)/the number of a requested (sub) segment.
- b) The SBD client finds the row i in the timeline/orderline table such that (sub)segment's EPT is $t_i \leq EPT < t_{i+1}$ or segment number = n_i .
- c) For each @name in the j th key element in the Essential Property descriptor
 - 1) Find a matching item key in keylist array
 - 2) Find the corresponding item value in the v array of the entry row i
 - 3) If @template exists,
 - i) replace the @template's substring between "\$" with value and add the results to the end of query string
 - ii) Otherwise, add key=value to the end of query q
- d) Return the query q to the DASH client

NOTE when the full URL is assembled, query q will be prefixed with the character "?" if it starts the query part of the URL, or "&" otherwise.

7.2.3 Segment request

The DASH client appends query q to the end of the segment URL used in the (sub)segment request.

NOTE Without SBD, all requests for different subsegments within a representation will use the same URL, but different byte ranges. Application of SBD will result in different q values for different segments (depending on their earliest presentation time), hence the single URL assumption no longer holds.

8 Examples

8.1 General

Electronic copies of examples are provided at <https://standards.iso.org/iso-iec/23009/-8/ed-1/en>.

8.2 MPD