ATSC
ADVANCED TELEVISION
SYSTEMS COMMITTEE

Amendment No. 3 to ATSC Standard A/336:2018, “Dynamic Event Message Enhancements”

Doc. A/336:2018-Amend-No-3
3 October 2019

Advanced Television Systems Committee
1776 K Street, N.W.
Washington, D.C. 20006
202-872-9160

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at https://www.atsc.org/feedback/.

Revision History

Version	Date
Amendment approved	3 October 2019

ATSC Amendment No. 3 to A/336:2018, Dynamic Event Message Enhancements

1. OVERVIEW

1.1 Definition

An Amendment is generated to document an enhancement, an addition or a deletion of functionality to previously agreed technical provisions in an existing ATSC document. Amendments shall be published as attachments to the original ATSC document. Distribution by ATSC of existing documents shall include any approved Amendments.

1.2 Scope

This document amends the A/336 specification, specifying a new video watermark message type for delivery of dynamic event messages in long-form watermark messages. The proposed amendment adds the dynamic_event_message() to the set of video watermark messages that can be delivered in long form in order to:

- increase the maximum size of dynamic_event_message() that can be conveyed using the video watermark; and
- enable more flexible scheduling of dynamic_event_message() fragments with other messages in the video watermark.

In addition, this document corrects an inconsistency between multiple locations in the text that specify conflicting semantics for the **last_fragment** field.

This document also provides editorial corrections to the terminology used in describing the display override state.

The changes described in this document are relative to A/336:2018, dated 11 December 2018, and do not reflect changes proposed in other Amendments.

1.3 Rationale for Changes

The current A/336 specification allows dynamic_event_message() events to be delivered only as short-form messages. Limitations of the short-form message structure impose a maximum size on a dynamic_event_message() of 58 bytes for the three variable-length fields (scheme_id_uri, value_str, and data fields) if the message is sent using the 1X video watermark system or 178 bytes using the 2X system.

This limitation can be problematic because no corresponding limitation exists for delivery of dynamic events using other delivery methods (e.g. ISO BMFF 'emsg' box or MPD) and some important use cases employ stream events that exceed this maximum size. One example of such a use case is the delivery of a SCTE 35 cue message as a dynamic event.[1] Even the simplest SCTE 35 cue message can exceed the limit of a 1X video watermark dynamic event message.[2]

While a dynamic message whose payload exceeds the limit could be divided into segments and transmitted using multiple separate video watermark messages, this would require that application developers implement message handling protocols that are transport-specific and incur the significant overhead of repeatedly transmitting the non-data message fields (e.g. scheme_id_uri and value_str) with each message segment.

This amendment addresses this issue by substantially increasing the maximum size of a dynamic event message transmitted through the video watermark to sufficient size to support known use cases.

Another limitation addressed by this amendment derives from the constraint in Section 5.1.2 of A/336 that short-form messages can only be multiplexed (i.e. interleaved) with long-form messages. If a broadcaster wants to transmit a multi-fragment short-form dynamic_event_message() during the same content interval as other short-form watermark messages subject to precise time constraints, such as presentation_time_message() and extended_vp1_message (defined in A/336:2018 Amendment 1), the messages are not permitted to be multiplexed with one another. This constraint will introduce latency in dynamic event delivery and additional scheduling complexity because it becomes necessary to identify and resolve scheduling conflicts and delay transmission of multi-fragment dynamic events to avoid a scheduling conflict with a time-sensitive message. This can be avoided by using long-form messages as described in this amendment for multi-fragment dynamic events, which are permitted to be multiplexed with short-form watermark messages.

The correction to the text defining the semantics of the last_fragment field is necessary as the current definition in the text is inconsistent with the definitions and usage of the field included in Tables 5.2 and 5.3.

1.4 Compatibility Considerations

The changes described in this document for the new video watermark message type are backward-compatible relative to the currently published version of the standard to which this Amendment pertains and any previously approved Amendments for that standard. It is anticipated that support for the new message will only require modifications to software (i.e., no hardware changes should be needed as long messages already exist in the current standard).

Implementers are advised that the newly defined long-form dynamic event message may not be reported to Broadcaster Applications by A/344 runtimes that conform to earlier versions of the specification.

[1] For live production scenarios in particular, where it is desirable to distribute an ad break announcement with the lowest possible latency, the video watermark provides an beneficial path for SCTE 35 cue message delivery because transmission of the message in the video essence can begin immediately (without needing to wait for an ISO BMFF segment start or an MPD update).

[2] See, for example, the example binary SCTE 35 cue message given on slide 5 of the "Ad Insertion in MPEG DASH" presentation given at the April 2, 2015 meeting of the San Diego Section of the IEEE.

The inconsistency between the definitions of the last_fragment field in the table and the semantics should not affect any existing implementations, as it is not possible to implement this field as it is currently defined in the text.

The proposed changes to the display override definitions are editorial clarifications and are not anticipated to impact implementations.

2. LIST OF CHANGES

Change instructions are given below in *italics*. Unless otherwise noted, inserted text, tables, and drawings are shown in blue; deletions of existing text are shown in ~~red strikeout~~. The text "[ref]" indicates that a cross reference to a cited referenced document should be inserted.

2.1 Normative References

No changes.

2.2 Informative References

No changes.

2.3 Acronyms and Abbreviations

No changes.

2.4 Terms

No changes.

2.5 Change Instructions

In Section 5.1.1, following Table 5.2, *correct the bit number and bitmask for the last_fragment semantics to match the definition in Table 5.2.*

last_fragment – This 2-bit or 8-bit value shall specify the fragment number of the last fragment used to deliver the complete watermark message. When (wm_message_id & 0x~~4~~80)==0) , i.e. bit ~~6~~7 is value '0', then last_fragment shall be 2 bits in length. When (wm_message_id & 0x~~4~~80)==1) , i.e. bit ~~6~~7 is value '1', then last_fragment shall be 8 bits in length. A value of zero in last_fragment indicates no segmentation is used (the watermark message contained within is complete). A value of 1 in last_fragment indicates the wm_message() will be delivered in two parts, a value of 2 indicates the watermark message will be delivered in three parts, and a value of 3 indicates it will be delivered in four parts, etc. The pair of values fragment_number and last_fragment may be considered to signal "part M of N~~.~~".

In Section 5.1.1, Table 5.3, *add an entry for dynamic_event_message() to the "Max Fragments = 256" section of the table.*

Table 5.3 wm_message_id Encoding

wm_message_id Value	Max Fragments	Message	Reference
0x00	4	reserved	
0x01		content_id_message()	Sec. 5.1.4
0x02		presentation_time_message()	Sec. 5.1.5
0x03		uri_message()	Sec. 5.1.6
0x04		vp1_ message()	Sec. 5.1.7
0x05		dynamic_event_message()	Sec. 5.1.8
0x06		display_override_message()	Sec. 5.1.9
0x07-0x7E		reserved	
0x7F		user_private_message()	Sec. 5.1.11
0x80	256	AEA_message()	Sec. 5.1.10
0x81		dynamic_event_message()	Sec. 5.1.8
0x82~~1~~-0xFE		reserved	
0xFF		user_private_message()	Sec. 5.1.11

***In Section 5.1.8,** extend constraints defined at the end of the section (following the "reserved1" field) to specify which constraints apply to the short-form message and which apply to the long-form message.*

The following constraints apply:

- When **delivery_protocol_type** has a value equal to 1 or 2 and the message is sent in a short-form message, the sum of the values of the **scheme_id_uri_length**, **value_strlen**, and **data_length** fields shall be less than or equal to 58 for 1X video watermark emission format (1X System) and shall be less than or equal to 178 for 2X video watermark emission format (2X System) [5].
- When **delivery_protocol_type** has a value equal to 1 or 2 and the message is sent in a long-form message, the sum of the values of the **scheme_id_uri_length**, **value_strlen**, and **data_length** fields shall be less than or equal to 4,838 for 1X video watermark emission format (1X System) and shall be less than or equal to 12,518 for 2X video watermark emission format (2X System).
- ~~Otherwise (i.e. w~~When **delivery_protocol_type** has a value ~~other than value 1 or 2),~~ designated in Table 5.12 as "Reserved" or "Industry Reserved", and does not have an encoding defined in the ATSC Code Point Registry [ref], and the message is sent in a short-form message, the value of **reserved1_field_length** shall be less than or equal to 78 for 1X video watermark emission format (1X System) and shall be less than or equal to 198 for 2X video watermark emission format (2X System) [5].
- When **delivery_protocol_type** has a value designated in Table 5.12 as "Reserved" or "Industry Reserved", and does not have an encoding defined in the ATSC Code Point Registry [ref], and the message is sent in a long-form message, the value of **reserved1_field_length** shall be less than or equal to 4,858 for 1X video watermark emission format (1X System) and shall be less than or equal to 12,538 for 2X video watermark emission format (2X System).
- When **delivery_protocol_type** has a value designated in Table 5.12 as "Industry Reserved" and does have an encoding defined in the ATSC Code Point Registry [ref], the **reserved1_field_length** and **reserved1** fields are replaced with encoding defined by the registrant.

***In Section 5.2.4,** consistently use the term "display override state" instead of switching between "display override state", "display override condition", and "display override indicator".*

5.2.4 Display Override State ~~Indicator~~

The d~~D~~isplay override state may be signaled in the audio watermark to indicate that receivers should suspend any modification (audio or video) to the presentation of the marked content which is occurring as a result of signaling obtained via the recovery process.

The d~~D~~isplay override state ~~condition~~ shall be indicated by the use of inverse signaling in the audio watermark emission as defined in A/334 [4].

The display override state is in effect solely for the duration of the marked content where inverse signaling is employed.

Transitions between standard signaling and inverse signaling shall occur at symbol boundaries and are not required to occur at ~~a~~ VP1 Audio Watermark Segment, Cell, header, Packet, or payload field boundaries.

***In Section 5.4.3, following Table 5.28**, clarify the description of the displayOverride element.*

`displayOverride` – When present, this element indicates the display override state corresponding to the VP1 payload used to form the query request that resulted in this recovery data table. This element shall have the value 1 if the display override state was true for any part of the Cell conveying that~~whose~~ VP1 payload ~~contained the intervalCode and serverCode in the query request to which the recovery data table was provided as a response, or~~; 0 otherwise.

End of Document