ATSC Candidate Standard: Content Recovery in Redistribution Scenarios (A/336)

Doc. S33-178r2
15 January 2016

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

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards for digital television. The ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

Specifically, ATSC is working to coordinate television standards among different communications media focusing on digital television, interactive systems, and broadband multimedia communications. ATSC is also developing digital television implementation strategies and presenting educational seminars on the ATSC standards.

ATSC was formed in 1982 by the member organizations of the Joint Committee on InterSociety Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Telecommunications Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). Currently, there are approximately 150 members representing the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

ATSC Digital TV Standards include digital high definition television (HDTV), standard definition television (SDTV), data broadcasting, multichannel surround-sound audio, and satellite direct-to-home broadcasting.

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.

This specification is being put forth as a Candidate Standard by the TG3/S33 Specialist Group. This document is an editorial revision of the Working Draft (S33-178r1) dated 18 December 2015. All ATSC members and non-members are encouraged to review and implement this specification and return comments to cs-editor@atsc.org. ATSC Members can also send comments directly to the TG3/S33 Specialist Group. This specification is expected to progress to Proposed Standard after its Candidate Standard period.

Revision History

Version	Date
Candidate Standard approved Notes to readers are shown in *blue italics*. Editorial update of the Table of Contents	15 January 2016 17 March 2016
Standard approved	Insert date here

Table of Contents

Index of Figures and Tables

ATSC Candidate Standard:
Content Recovery in Redistribution Scenarios

1. SCOPE

This document specifies payload formats for video and audio watermarks specified for use within ATSC 3.0 broadcast emissions.

1.1 Introduction and Background

A Redistribution scenario is a situation where an ATSC 3.0 receiver is getting TV programming through an intermediary such as a Multi-channel Video Program Distributor (MVPD) – e.g., a cable, satellite or IPTV operator. If there is supplementary content such as applications or alternate audio tracks that is not passed through by the intermediary, the original broadcaster of the content might want to make such supplementary content available via alternate means.

The document specifies how certain signaling information can be carried in audio watermark payloads, video watermark payloads, and the user areas of audio tracks, and how this information can be used to access supplementary content in a redistribution scenario.

1.2 Organization

This document is organized as follows:

- Section 1 – Outlines the scope of this document.
- Section 2 – Lists normative and informative references.
- Section 3 – Defines terms, acronyms, and abbreviations for this document.
- Section 4 – Provides a system overview
- Section 5 – Specifies audio/video watermark payload
- Annex A – Discusses implications of set-top box operations
- Annex B – Specifies the recovery file JSON schema
- Annex C – Provides an informative discussion of acquisition of signaling using video watermarks
- Annex D – Describes acquisition of signaling using VP1 payloads/messages

2. REFERENCES

All referenced documents are subject to revision. Users of this Standard are cautioned that newer editions might or might not be compatible.

2.1 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

[1] Ad-ID: "Ad-ID Structure," http://ad-id.org/how-it-works/ad-id-structure, Advertising Digital Television, LLC, 2014. *<Note: This organization is not currently on the list of accepted ATSC normative organizations. S33 proposes to address this during CS Phase.>*

[2] ATSC: "Signaling, Delivery, Synchronization and Error Correction," Document A/331, Advanced Television Systems Committee, *under development.*

[3] ATSC: "Service Announcement," Document A/332, Advanced Television Systems Committee, *under development.*

[4] ATSC: "Service Usage Reporting," Document A/333, Advanced Television Systems Committee, *under development.*

[5] ATSC: "Audio Watermark Emission," Document A/334, Advanced Television Systems Committee, *under development*.

[6] ATSC: "Video Watermark Emission," Document A/335, Advanced Television Systems Committee, *under development.*

[7] ATSC: "Application Signaling and Triggers," Document A/337, Advanced Television Systems Committee, *under development.*

[8] EIDR: "Id Format," http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf, Entertainment ID Registry Association, 30 January, 2012. *<Note: This organization is not currently on the list of accepted ATSC normative organizations. S33 proposes to address this during CS Phase.>*

[9] IEEE: "Use of the International Systems of Units (SI): The Modern Metric System," Doc. SI 10-2002, Institute of Electrical and Electronics Engineers, New York, N.Y.

[10] IETF: "Uniform Resource Identifiers (URI): Generic Syntax," RFC 3986, January, 2005.

[11] ISO: ISO 3166-1:2013 (E/F), "Codes for the representation of names of countries and their subdivisions — Part 1: Country codes," International Organization for Standardization, 3rd Edition, 11/13/2013.

[12] ISO: "ISO/IEC IS 13818-1(2013) (E), International Standard, Information technology – Generic coding of moving pictures and associated audio information: systems."

[13] ISO: ISO/IEC 23009-1:2014, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats," International Organization for Standardization, 2nd Edition, 5/15/2014.

[14] OASIS: "Common Alerting Protocol" Version 1.2, 1 July 2010. http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf

2.2 Informative References

The following documents contain information that may be helpful in applying this Standard.

[15] ATSC: "Application Runtime Environment," Document A/344, Advanced Television Systems Committee, *under development.*

3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute's published standards [9] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

shall – This word indicates specific provisions that are to be followed strictly (no deviation is permitted).

shall not – This phrase indicates specific provisions that are absolutely prohibited.

should – This word indicates that a certain course of action is preferred but not necessarily required.

should not – This phrase means a certain possibility or course of action is undesirable but not prohibited.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., restricted), may contain the underscore character (e.g., sequence_end_code) and may consist of character strings that are not English words (e.g., dynrng).

3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These are used primarily to enable adding new values to a syntactical structure without altering its syntax or causing a problem with backwards compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1.' There is no default value for other reserved elements. Use of reserved elements except as defined in ATSC Standards or by an industry standards setting body is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently-reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently-reserved elements to avoid possible future failure to function as intended.

3.3 Acronyms and Abbreviation

The following acronyms and abbreviations are used within this document.

Ad-ID – Format identifier for identifying advertising assets

ATSC – Advanced Television Systems Committee

BSID – Broadcast Stream ID

BCH(n, k, t) – Bose-Chaudhuri-Hocquenghem Error Correction Code having an n-bit codeword with k information bits (and hence n-k parity check bits) that is able to correct up to t bit-errors in the information bits.

bslbf – Bit string, left bit first.

CAP – Common Alerting Protocol

EIDR – Entertainment Industry Data Registry

HDMI – High-Definition Multimedia Interface

LSB – Least significant bit.

MSB – Most significant bit.

MVPD – Multi-channel Video Program Distributor

SMPTE – Society of Motion Picture and Television Engineers

uimsbf – Unsigned integer, most significant bit first.

URI – Uniform Resource Identifier

3.4 Terms

The following terms are used within this document.

Broadcast Stream ID – Identifier of a broadcast stream, as defined in Section 6.3.2 of A/331 [2].

Broadcast Stream – The abstraction for an RF Channel which is defined in terms of a carrier frequency centered within a specified bandwidth.

cell – Has the meaning given in the ATSC A/334 Audio Watermark Emission specification [5].

dynamic event – Has the meaning given in the A/337 [7].

I(x) mod G(x) – The remainder that results from the polynomial division of I(x) by G(x).

interval code – A value that identifies the interval of content in which the VP1 payload value is embedded.

large domain – A division of the payload wherein the interval field has a size of 25 bits, sufficient to support watermarking of approximately 1 year, 7 months of content.

major_channel_no – Part of the identification of a service, as defined in Section 6.3.2 of A/331 [2]. The combination of major and minor channel number is unique within a broadcast area. I.e., a receiver of over-the-air broadcasts will never see two different services with the same major/minor channel number combination.

minor_channel_no – Part of the identification of a service, as defined in Section 6.3.2 of A/331 [2].

reserved – Set aside for future use by a Standard.

packet – Has the meaning given in the ATSC A/334 Audio Watermark Emission specification [5].

parity – A division of the cell that carries the BCH code parity check bits.

payload – A division of the cell that carries the information bits.

query flag – The value of the query_flag field in an instance of the VP1 payload.

server code – A value that identifies a server which acts as the starting point for acquisition of supplementary content. This code is assigned by an administrative or regulatory authority designated by ATSC for use in a VP1 payload.

small domain – A division of the payload wherein the interval field has a size of 17 bits, sufficient to support watermarking of approximately 54 hours of content.

VP1 – The watermarking technology standardized in the ATSC A/334 Audio Watermark Emission specification [5].

VP1 payload – The specific arrangement of the 50-bit payload in domain_type, server_field, interval_field, and query_flag specified in the present document.

3.5 Extensibility

The ATSC watermark system is designed such that new message types and new data elements can be added at a later time without adverse impact to existing receivers. For the video watermark system:

- The syntax of the wm_message() includes a wm_message_type field. While the current standard defines four message types, future revisions of the standard could define additional types. Existing receivers are expected to use the wm_message_length field to gracefully discard any message types they do not recognize or support.
- The wm_message() may be extended in the future with new fields at the end (just in front of CRC_32. Legacy receivers would be expected to disregard the added fields.
- A video watermark with a different run-in pattern could be developed in the future. Legacy receivers would not recognize the new mark and treat it as unmarked content.
- In the content_id_message(), two different identifier schemes are specified in the present standard, EIDR and Ad-ID. In the future the content_id_message() could carry identifiers of

other types. Receivers not recognizing or supporting the new types are expected to discard the content_id_message().

For the audio watermark system

- An alternate packet format may be defined to carry a different payload in redistribution content. Note that an alternate packet format may not be decodable by existing receivers and it may be required to define a different or expanded emission specification in order for the alternate packet format to be capable of being embedded simultaneously in content with the existing payload.
- New network protocols may be defined for receivers to obtain recovery and signaling information from signaling servers. Such protocols may be used in new receivers as replacements for the existing protocols or for use in addition to the existing protocols to provide new functionality.
- New data fields may be defined for the existing recovery and signaling files. Such data fields may be employed by new receivers to provide additional functionality.
- New data values may be defined for elements in the existing recovery and signaling files. Such data values may be employed, for example, to enable the identification of new types of signaling files within the recovery file which may be retrieved by new receivers.

4. SYSTEM OVERVIEW

Note: S33 intends to complete this section during the Candidate Standard period.

5. SPECIFICATION

5.1 Video Watermark Payload Format

The emission format for video watermarks shall conform to the ATSC A/335 Video Watermark Emission specification [6]. As described in [6], the "1X" emission format delivers 30 bytes of data per video frame, while the "2X" system delivers 60 bytes per frame.

The VP1 Message (Section 5.1.7), when used, is repeated across multiple video frames. In instances where a watermark payload is not recovered from an individual video frame, a receiver may attempt to recover a VP1 Message by combining luma values (e.g. via averaging) from two or more successive video frames.

The payload format for the video watermark is the same in both the 1X and 2X systems. The run-in pattern (as specified in Section 5.1 of [6]) is followed by one or more instances of a message block. The watermark payload shall conform to the syntax given in Table 5.1.

Table 5.1 Bit Stream Syntax of the Watermark Payload

Syntax	No. of Bits	Format
watermark_payload() {		
run_in_pattern	16	0xEB52
for (i=0; i<N; i++) {		
wm_message_block()	var	uimsbf
}		
for (i=0; i<M; i++) {		
zero_pad	8	0x00
}		
}		

run_in_pattern – This 16-bit value is set to 0xEB52 to identify that the video line includes a watermark of the format specified herein.

wm_message_block() – A full wm_message() or a fragment of a larger wm_message(), formatted per Section 5.1.1 below. The assembly of a wm_message() from one or more wm_message_block() instances shall be as described in Table 5.4.

zero_pad – A value of zero used to pad to the end of the frame.

5.1.1 Watermark Message Syntax

Watermark message blocks shall follow the syntax given in Table 5.2 below and the semantics that follow.

Table 5.2 Bit Stream Syntax for the Watermark Message Block

Syntax	No. of Bits	Format
wm_message_block() {		
wm_message_id	8	uimsbf
wm_message_block_length	8	uimsbf
wm_message_version	4	uimsbf
fragment_number	2	uimsbf
last_fragment	2	uimsbf
wm_message_bytes()	var	
if ((fragment_number == last_fragment) && (fragment_number != 0)) {		
message_CRC_32	32	uimsbf
}		
CRC_32	32	uimsbf
}		

wm_message_id – This 8-bit value shall uniquely identify the syntax and semantics of the data bytes carried in the message block, coded according to Table 5.3 below. As indicated in the table, the encodings of the wm_message() types defined in the present standard may be found in Sections 5.1.4 through 5.1.10.

Table 5.3 wm_message_id Encoding

wm_message_id Value	wm_message()	Reference
0x00	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	emergency_alert_message()	Sec. 5.1.9
0x07	display_override_message()	Sec. 5.1.10
0x08-0x7F	reserved	
0x80-0xFF	user private	

wm_message_block_length – This 8-bit value shall specify the number of remaining bytes in the wm_message_block() that immediately follows this field up to and including the CRC_32 field.

wm_message_version – This 4-bit value shall be incremented if and only if anything in the wm_message() changes, with wrap-around to 0 after the value reaches 15. The watermark processor in the receiving device is expected to use wm_message_version to discard duplicates. The video signal may include repeated instances of the same wm_message() to improve reliability of delivery.

fragment_number – This 2-bit value shall specify the number of the current message fragment minus one. For example, value of '00' in fragment_number indicates the wm_message_block() carries the first (or only) fragment of a message, a fragment_number value of '01' indicates the wm_message_block() carries the second fragment of a message, etc. The value of fragment_number shall be less than or equal to the value of last_fragment.

last_fragment – This 2-bit value shall specify the fragment number of the last fragment used to deliver the complete wm_message(). A value of '00' in last_fragment indicates no fragmentation is used (the wm_message() contained within is complete). A value of '01' in last_fragment indicates the wm_message() will be delivered in two parts, a value of '10' indicates the wm_message() will be delivered in three parts, and a value of '11' indicates it will be delivered in four parts. The pair of values fragment_number and last_fragment may be considered to signal "part M of N."

wm_message_bytes() – When the value of last_fragment is 0, wm_message_bytes() shall be a complete instance of the watermark message identified by the value of wm_message_id. When the value of last_fragment is non-zero, wm_message_bytes() shall be a fragment of that watermark message, where each fragment is constructed according to the rules given in Section 5.1.2. The concatenation of all instances of wm_message_block() with a given wm_message_id and wm_message_fragment_version number shall result in the complete wm_message() associated with that wm_message_id.

message_CRC_32 – When a message is sent in two or more fragments (e.g. last_fragment > 0) a 32-bit CRC covering the complete message (before fragmentation) shall be provided in the last fragment of a fragmented message. The message_CRC_32 field shall not be present for non-fragmented messages (e.g. when the value of last_fragment is 0) or in any fragment other than the last (e.g. when fragment_number ≠ last_fragment). The message_CRC_32, when present, shall contain the CRC value that gives a zero output of the registers in the decoder defined in ISO/IEC 13818-1 [12], Annex A after processing the entire re-assembled message payload. The generating polynomial is $1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26}$.

CRC_32 – This 32-bit field shall contain the CRC value that gives a zero output of the registers in the decoder defined in ISO/IEC 13818-1 [12], Annex A after processing the entire message block. The generating polynomial is $1 + x + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26}$.

Table 5.4 Bit Stream Syntax of the Watermark Message

Syntax	No. of Bits	Format
wm_message() {		
wm_message_block(0)	var	Table 5.2
if (last_fragment>0) {		
wm_message_block(1)	var	Table 5.2
}		
if (last_fragment>1) {		
wm_message_block(2)	var	Table 5.2
}		
if (last_fragment>2) {		
wm_message_block(3)	var	Table 5.2
}		
}		

5.1.2 Fragmentation and Reassembly

A fragmentation/reassembly mechanism allows the syntax of the watermark payload to accommodate the delivery of messages that exceed the size that will fit within one video frame. As noted, an "M of N" scheme is used. Figure 5.1 illustrates fragmentation and reassembly for four different example messages, each of different lengths. Note the presence of the 32-bit message_CRC_32 in the last fragment of the fragmented messages.

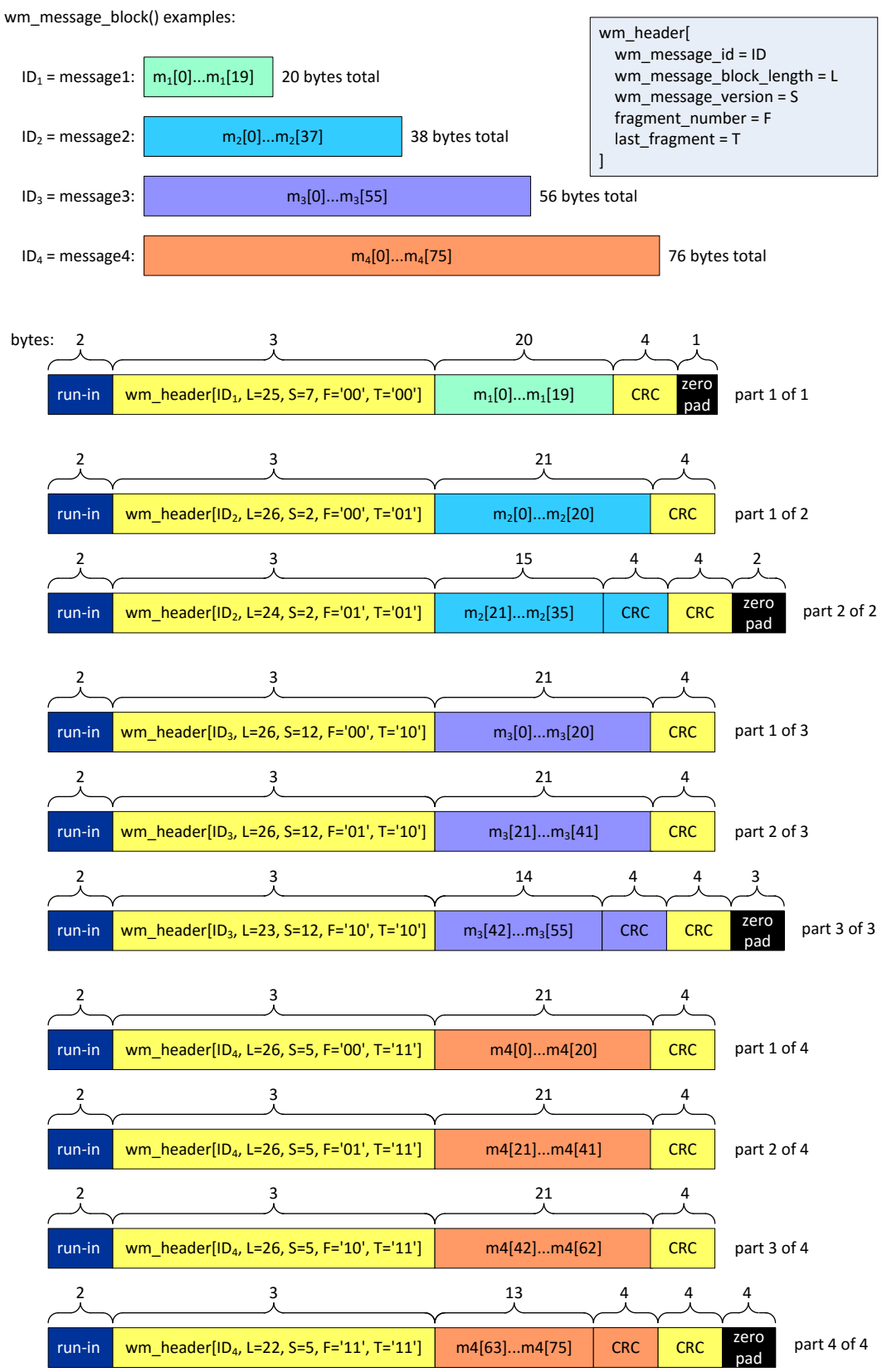

Figure 5.1 Illustration of Message Fragmentation (1X System).

5.1.3 Multiplexing and Processing Rules

When constructing the sequence of watermark messages, the sending side shall conform to the following rules:

1) If a wm_message_block() is sent with a fragment_number value of 0 (meaning first or only fragment) and a non-zero value of last_fragment (meaning that it will be sent in two or more fragments), each successive wm_message_block() shall have the same value in wm_message_id until the last fragment is sent (e.g. the value of fragment_number equals the value of last_fragment), with no intervening messages with other values of wm_message_id.
2) Fragments shall be sent in order. In other words, if n fragments are sent, the fragment_number values shall start at 0 and increase monotonically until the last fragment is sent. The fragment_number value in the last fragment shall be equal to the value of last_fragment.
3) Any given wm_message_block() (as indicated by the values of wm_message_id plus wm_message_sequence) may be sent multiple times. Receivers are expected to discard duplicates.

5.1.4 Content ID Message

The Content ID Message can deliver a content identifier associated with the program, and the major/minor channel number associated with the Service. The Content ID Message shall not be sent in fragments (e.g. the value of last_fragment in the message header shall be '00'). Two types of content identifiers are currently defined, one registered with the Entertainment Industry Data Registry (EIDR, (http://eidr.org/), and one an Ad-ID registered at http://ad-id.org. The syntax and bitstream semantics of the Content ID Message shall be as given in Table 5.5 and the parameter descriptions that follow.

Table 5.5 Bit Stream Syntax for the Content ID Message

Syntax	No. of Bits	Format
content_id_message() {		
content_ID_present	1	bslbf
channel_ID_present	1	bslbf
reserved	6	'111111'
if (content_ID_present) {		
reserved	3	'111'
content_ID_type	5	uimsbf
if (content_ID_type==EIDR) {		
EIDR	96	uimsbf
} else if (content_ID_type==adID) {		
adID_strlen	8	uimsbf
adID_string()	adID_strlen*8	ASCII char. string
} else {		
reserved1_field_length (N1)	8	uimsbf
reserved1	8*N1	'11..'
}		
}		
if (channel_ID_present) {		
BSID	16	uimsbf
reserved	4	'1111'
major_channel_no	10	uimsbf
minor_channel_no	10	uimsbf
}		
}		

content_ID_present – This one-bit Boolean flag shall indicate, when set to '1', that the fields associated with content ID are present in the message. When set to '0',fields associated with content ID shall not be present.

channel_ID_present – This one-bit Boolean flag shall indicate, when set to '1', that the fields associated with channel ID are present in the message. When set to '0', the fields associated with channel ID shall not be present.

content_ID_type – This 5-bit unsigned integer field shall identify the type of content identifier provided in the message coded according to Table 5.6 below.

Table 5.6 content_ID_type field Encoding

content_ID_type Value	Meaning
0x00	Reserved
0x01	EIDR
0x02	Ad-ID
0x03-0x1F	Reserved

<Note: S33 is considering allocating 0x1F value to user private data such as for an ISCI code (predecessor to Ad-ID) or house number or other application. S33 expects to resolve this question during CS Phase with input from broadcasters and other ATSC members.

EIDR – This 96-bit (8 byte) value shall be an identifier registered with EIDR (http://eidr.org/) in the "compact binary format" defined by EIDR [8]. (It consists of a16-bit sub-prefix generated by interpreting the sub-prefix as a binary value, and 80-bit suffix representing the non-checksum part of the suffix. It shall be the EIDR identifier associated with the current broadcast content.

adID_strlen – This is an 8-bit unsigned integer that shall signal the number of characters in the adID_string() to follow.

adID_string() – This ASCII character string shall represent an Ad-ID identifier registered at http://www.ad-id.org [1]. The length of the string shall be as given by the value of adID_strlen. The length (as specified by Ad-ID as of the date of this publication) is 11 or 12 characters.

reserved1_field_length – This 8-bit unsigned integer field shall give the length in bytes of the reserved1 field which immediately follows.

reserved1 – This is reserved for future use.

BSID – This 16-bit unsigned integer field shall identify the Broadcast Stream ID of the original emission signal.

major_channel_no – This 10-bit unsigned integer field shall identify the major channel number associated with the content. The combination of major and minor channel numbers shall be scoped to the broadcast area.

minor_channel_no – This 10-bit unsigned integer field shall identify the minor channel number associated with the content.

The following constraints apply:

- At least one of **content_ID_present** or **channel_ID_present** shall have a value equal to '1'.
- When **channel_ID_present** is equal to 0 and **content_ID_present** is equal to 1, the value of **adID_strlen** shall be less than or equal to 85 for 1X video watermark emission format (1X System) and shall be less than or equal to 205 for 2X video watermark emission format (2X System) [6].
- When **channel_ID_present** is equal to 1 and **content_ID_present** is equal to 1, the value of **adID_strlen** shall be less than or equal to 80 for 1X video watermark emission format (1X System) and shall be less than or equal to 200 for 2X video watermark emission format (2X System) [6].
- When **channel_ID_present** is equal to 0 and **content_ID_present** is equal to 1, and **content_ID_type** is in the range of 0x03 to 0x1F (inclusive), the value of **reserved_field_length** shall be less than or equal to 85 for 1X video watermark emission format (1X System) and shall be less than or equal to 205 for 2X video watermark emission format (2X System) [6].
- When **channel_ID_present** is equal to 1 and **content_ID_present** is equal to 1, and **content_ID_type** is in the range of 0x03 to 0x1F (inclusive), the value of **reserved_field_length** shall be less than or equal to 80 for 1X video watermark emission format (1X System) and shall be less than or equal to 200 for 2X video watermark emission format (2X System) [6].

5.1.5 Presentation Time Message

A Presentation Time Message shall be carried within a single watermark payload. (I.e., value of last_fragment shall be 0 for a message block carrying a Presentation Time Message.) The Presentation Time Message provides an indication to the receiver of the presentation time of the video frame carrying the watermark. If a given instance of the presentation_time_message() is repeated

in immediate succession, the indicated presentation time shall correspond with the video frame of the first occurrence. The syntax and bitstream semantics of the presentation_time_message() shall be as given in Table 5.7 and the parameter descriptions that follow.

Table 5.7 Bit Stream Syntax for the Presentation Time Message

Syntax	No. of Bits	Format
presentation_time_message() {		
presentation_time	32	uimsbf
reserved	6	'111111'
presentation_time_ms	10	uimsbf
}		

presentation_time – This 32-bit unsigned integer shall indicate the presentation time of the frame associated with the watermark, as the least-significant 32 bits of the count of the number of seconds since January 1, 1970 00:00:00, International Atomic Time (TAI).

presentation_time_ms – This 10-bit unsigned integer in the range 0 to 999 shall indicate the milliseconds offset from the time indicated in presentation_time, such that the formula presentation_time + (presentation_time_ms/1000) yields the actual presentation time to the nearest 1 millisecond.

In order for the presentation time to be meaningful in a ROUTE/DASH service, the **MPD**@suggestedPresentationTime shall be present, and the resulting wall clock presentation time corresponding to the StartTime of each Period in the MPD shall be equal to the duration of the previous Period plus the wall clock presentation time corresponding to the StartTime of the previous Period.

5.1.6 URI Message

The URI Message is used to deliver URIs of various types. The URI Message may be sent in fragments (e.g. the value of last_fragment in the message header may be non-zero).

Table 5.8 Bit Stream Syntax for the URI Message

Syntax	No. of Bits	Format
uri_message() {		
uri_type	8	uimsbf
uri_strlen	8	uimsbf
URI_string()	8*uri_strlen	
}		

uri_type – An 8-bit unsigned integer field that shall identify the type of URI to follow, according to the encoding given in Table 5.9.

Table 5.9 uri_type field Encoding

uri_type Value	**Meaning**
0x00	Reserved
0x01	Signaling server (providing access to the Service Layer Signaling (SLS), as specified in section 7 of A/331 [2]).
0x02	ESG data server (providing access to the ESG data, as specified in A/332 [3]).
0x03	URL of Service Usage Data Gathering Report server (for use in reporting service usage, as specified in A/333 [4]).
0x04-0xFF	Reserved

uri_strlen – An 8-bit unsigned integer that shall signal the number of characters in the uri_string() to follow. The value of the uri_strlen field shall be less than or equal to 86 for 1X video watermark emission format (1X System) and shall be less than or equal to 206 for 2X video watermark emission format (2X System) [6].

URI_string() – A URI consisting of characters whose values shall be restricted to those allowed for Uniform Resource Identifiers (URIs) by RFC 3986 [10]. The length of the string shall be as given by the value of uri_len. The character string, after reassembly if the URI is sent in fragments, shall be a valid URI per RFC 3986 [10].

5.1.7 VP1 Message

The VP1 Message enables the recovery process (specified in Section 5.3) to be employed in conjunction with the Video Watermark.

The bit stream syntax of the VP1 Message shall be as shown in Table 5.10.

Table 5.10 Bit Stream Syntax for the VP1 Message

Syntax	**No. of Bits**	**Format**
vp1_message() {		
header	32	bslbf
packet()	127	bslbf
zero_pad	1	'0'
}		

header – This 32-bit field shall consist of a header element as specified in ATSC A/334 Audio Watermark Emission [5].

packet() – This 127-bit field shall be as given by Table 5.16 and the parameter descriptions that follow.

zero_pad – This one-bit field shall be set to value '0'.

When present, the VP1 Message shall be the first (i.e. left-most) wm_message() present in a video frame. When present, VP1 Messages carrying identical data shall be repeated for all successive video frames across at least a 1/6 second duration of content (a "message block"). The value of wm_message_sequence shall not increment within a message block.

A video watermark segment shall consist of video content carrying a series of successive message blocks whose initial video frames are nominally at 1.5 second intervals such that if the initial video frame of the first message block in a watermark segment occurs at time *T* seconds, the initial video frame of the *n*th successive message block in the watermark segment occurs within +/-0.5 video frames of time *T*+1.5*n* seconds. All message blocks in a video watermark segment

shall have the same server code and successive message blocks in a video watermark segment shall have sequentially incrementing interval codes. The query_flag value in the VP1 payload may change between successive message blocks in a video watermark segment.

When the vp1_message() is carried in a video component of audiovisual content for which an audio component employs an audio watermark carrying the same VP1 payload, the message blocks of the video watermark shall be time-aligned such that the initial video frame in every message block occurs within +/-0.5 video frames of the corresponding starting cell boundary in the VP1 audio watermark on the presentation timeline.

Figure 5.2 illustrates the temporal structure of message blocks carrying VP1 Messages in a watermark segment, with time alignment to a VP1 audio watermark carrying the same VP1 payload.

Because the VP1 Message contains header and parity bits in addition to payload bits and because it is always repeated in multiple video frames, it may be recoverable from content for which the run-in sequence is not recoverable or where there are bit errors which cause the CRC-32 check to fail. Receivers may attempt to recover the VP1 Message in instances where run-in sequence recovery or CRC-32 check are unsuccessful.

A VP1 message block is defined to be a series of successive video frames spanning at least a 1/6 second duration of content in which each video frame carries an identical vp1_message().

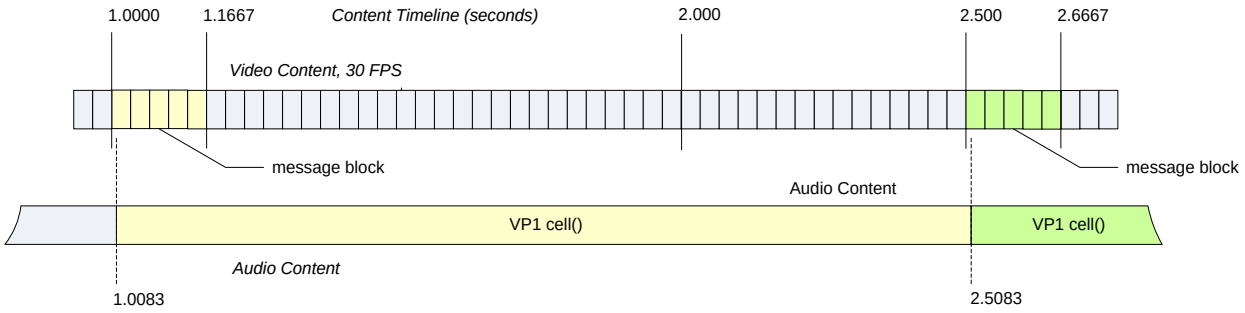

Figure 5.2 Temporal Structure of VP1 Message Blocks

Figure 5.2 illustrates the temporal structure of VP1 message blocks carrying a VP1 Message in a watermark segment, with time alignment to a VP1 audio watermark. The yellow-shaded message block carries the same watermark information as does the yellow-shaded VP1 cell() in the audio signal. Note that the VP1 message block spacing is exactly 1.5 seconds and the audio signal VP1 cell() is offset from the initial video frame of each message block by ¼ of a video frame period.

5.1.8 Dynamic Event Message

The dynamic_event_message() supports delivery of dynamic Events in video watermarks. The syntax and bitstream semantics of the Dynamic Event Message shall be as given in Table 5.11 and the parameter descriptions that follow.

Table 5.11 Bit Stream Syntax for the Dynamic Event Message

Syntax	No. of Bits	Format
dynamic_event_message() {		
delivery_protocol_type	4	uimsbf
reserved	4	'1111'
if (delivery_protocol_type == '1' || '2'){		
scheme_id_uri_length (N1)	8	uimsbf
scheme_id_uri_string	8*N1	
value_strlen (N2)	8	uimsbf
value_string	8*N2	
timescale	32	uimsbf
presentation_time	32	uimsbf
reserved	6	'111111'
presentation_time_ms	10	
duration	32	uimsbf
id	32	uimsbf
data_length (N3)	8	uimsbf
data	8*N3	
} else {		
reserved1_field_length (N1)	8	uimsbf
reserved1	8*N1	'11..'
}		
}		

delivery_protocol_type – This 4-bit field shall signify the delivery protocol (e.g., MMT or ROUTE/DASH) of the service to which the dynamic event applies. The Table 5.12 below describes the encoding of this field.

Table 5.12 delivery_protocol_type field Encoding

delivery_protocol_type	Meaning
0	Reserved
1	ROUTE/DASH
2	MMTP
3-15	Reserved for future use

scheme_id_uri_strlen – This 8-bit unsigned integer field shall give the length of the scheme_id_uri_string field in bytes.

scheme_id_uri_string – This string shall give the `schemeIdUri` for the Event Stream of the Event.

value_strlen – This 8-bit unsigned integer field give the length of the value_string field in bytes.

value_string – This string shall give the `value` for the Event Stream of the Event.

timescale – This 32-bit unsigned integer shall give the time scale for the Event Stream of the Event, in ticks/second as defined in the MPEG DASH standard [13], to be used for the duration field.

presentation_time – This 32-bit unsigned integer shall indicate the presentation time of the Event, as the least-significant 32 bits of the count of the number of seconds since January 1, 1970 00:00:00, International Atomic Time (TAI).

presentation_time_ms – This 10-bit unsigned integer in the range 0 to 999 shall indicate the milliseconds offset from the time indicated in presentation_time, such that the formula presentation_time + (presentation_time_ms/1000) yields the actual presentation time to the nearest 1 millisecond.

In order for the presentation_time to be meaningful in a ROUTE/DASH service, the **MPD**@suggestedPresentationTime shall be present in the DASH MPD, and the resulting wall clock presentation time corresponding to the start time of each Period in the MPD shall be equal to the duration of the previous Period plus the wall clock presentation time corresponding to the start time of the previous Period.

duration – This 32-bit unsigned integer shall give the duration of the Event, in the time scale of the Event.

id – This 32-bit unsigned integer field shall be an ID for the Event, unique within the Event Stream.

data_length – This 8-bit integer shall give the length of the data field in bytes.

data – This field shall contain data needed for responding to the event, if any. The format and use of the data is determined by the Event Stream specification, which will be defined in the standard for any standards-based Event, and which will be known to any application registering to receive the Event for any Event targeted to applications.

reserved1_field_length – This 8-bit unsigned integer field shall give the length in bytes of the reserved1 field, which immediately follows this field.

reserved1 – This is reserved for future use.

The following constraints apply:

- When **delivery_protocol_type** has a value equal to 1 or 2, the sum of the values of the **scheme_id_uri_length, value_strlen,** and **data_length** fields shall be less than or equal to 66 for 1X video watermark emission format (1X System) and shall be less than or equal to 186 for 2X video watermark emission format (2X System) [6].
- Otherwise (i.e. when **delivery_protocol_type** has a value other than value 1 or 2), the value of **reserved1_field_length** shall be less than or equal to 87 for 1X video watermark emission format (1X System) and shall be less than or equal to 207 for 2X video watermark emission format (2X System) [6].

5.1.9 Emergency Alert Message

The Emergency Alert Message supports delivery of emergency alert information in video watermarks. The syntax and bitstream semantics of the emergency_alert_message() shall be as given in Table 5.13 and the parameter descriptions that follow.

Table 5.13 Bit Stream Syntax for the Emergency Alert Message

Syntax	No. of Bits	Format
emergency_alert_message() {		
CAP_message_ID_length (N1)	8	uimsbf
CAP_message_ID	8*(N1)	
CAP_message_url_length (N2)	8	uimsbf
CAP_message_url	8*(N2)	
expires	32	uimsbf
urgency	1	bslbf
severity_certainty	4	bslbf
reserved	3	"111"
}		

CAP_message_ID_length – This 8-bit unsigned integer field gives the length of the CAP_message_ID field in bytes.

CAP_message_ID – This string shall give the ID of the CAP message defined in [14]. It shall be the value of the `cap.alert.identifier` element of the CAP message indicated by **CAP_message_url.**

CAP_message_url_length – This 8-bit unsigned integer field gives the length of the CAP_message_url field in bytes.

CAP_message_url – This string shall give the URL that can be used to retrieve the CAP message.

expires – This parameter shall indicate the latest expiration date and time of any `<info>` element in the CAP message, encoded as a 32-bit count of the number of seconds since January 1, 1970 00:00:00, International Atomic Time (TAI).

urgency – When set to '1', this flag shall indicate that the urgency of the most urgent `<info>` element in the CAP message is "Immediate." When set to '0', it shall indicate otherwise.

severity_certainty – This is a 4-bit field code that is derived from the values of the required CAP elements of certainty and severity. For both elements, the lowest two values have been merged. The encoding of severity_certainty shall be as given in Table 5.14.

Table 5.14 Codes for severity_certainty field

severity_certainty	Certainty	Severity
'00 00'b	Unknown/Unlikely	Unknown/Minor
'00 01'b	Unknown/Unlikely	Moderate
'00 10'b	Unknown/Unlikely	Severe
'00 11'b	Unknown/Unlikely	Extreme
'01 00'b	Possible	Unknown/Minor
'01 01'b	Possible	Moderate
'01 10'b	Possible	Severe
'01 11'b	Possible	Extreme
'10 00'b	Likely	Unknown/Minor
'10 01'b	Likely	Moderate
'10 10'b	Likely	Severe
'10 11'b	Likely	Extreme
'11 00'b	Observed	Unknown/Minor

’11 01’b	Observed	Moderate
’11 10’b	Observed	Severe
’11 11’b	Observed	Extreme

The following constraint shall apply: the sum of the values of the CAP_message_ID_length and CAP_message_url_length fields shall be less than or equal to 80 for 1X video watermark emission format (1X System) and shall be less than or equal to 200 for 2X video watermark emission format (2X System) [6].

<Note: During the CS Phase S33 will explore the possibility of carrying the essence (e.g., banner text or similar) of a CAP message in the video watermark.>

5.1.10 Display Override Message

The Display Override Message provides an indication that, for a specified duration, the source broadcast audio is expected to be presented without modification and the source broadcast video is expected to be rendered without any overlaid graphics or other obstructions.

Table 5.15 Bit Stream Syntax for the Display Override Message

Syntax	No. of Bits	Format
display_override_message() {		
reserved	4	‘1111’
override_duration	4	uimsbf
}		

override_duration – A 4-bit unsigned integer field, that when in the range 1–15 shall indicate the number of seconds the override should continue, following receipt of the message – unless superseded by a subsequent display override message. When the value of override_duration is ‘0’, the override is immediately over.

5.2 Audio Watermark Payload Format

The emission format for audio watermarks shall conform to the ATSC A/334 Audio Watermark Emission specification [5].

5.2.1 Packet

The bit stream syntax of the packet() structure shall be as shown below in Table 5.16. It is divided into two regions: scrambled parity and scrambled payload bits.

Table 5.16 Syntax of packet() Structure

Syntax	No. of Bits	Format
packet() {		
scrambled_parity	77	uimsbf
scrambled_vp1_payload	50	uimsbf
}		

scrambled_parity – The scrambled parity bits, where parity bits shall be as specified in Section 5.2.2.1, below.

scrambled_payload – The scrambled payload, where payload shall be as specified in Section 5.2.3, below.

5.2.2 Scrambling

The scrambled_parity shall be the parity bits scrambled via bitwise exclusive OR operation with the parity_whitening_sequence, shown below in Table 5.17. The scrambled_payload shall be the payload bits scrambled via bitwise exclusive OR operation with the payload_whitening_sequence, shown below in Table 5.17.

Table 5.17 Whitening Sequences

	parity_whitening_sequence	**payload_whitening_sequence**
Binary	11100110111111111101101101011110110010001000010010111000010010000000101100101	00100001000010100011000000001011100000011100110111

5.2.2.1 Parity Bits

A BCH (127, 50, 13) code shall be applied to the VP1 payload. We shall define the payload polynomial, $P(x)$, as the polynomial that results when the 50 payload bits are used to form a polynomial with the leftmost bit being the coefficient for order 49 and the rightmost bit being the coefficient for order 0. We shall define the extended payload polynomial, ($x^{77}P(x)$) as the payload polynomial $P(x)$, multiplied by x^{77}. This is equivalent to the polynomial whose coefficients are the payload bits appended (to the right) with 77 zero bits.

We shall define the generator polynomial $G(x)$ for the BCH (127, 50, 13) code as specified in in Table 5.18 below.

We shall define the remainder polynomial $R(x)$ as the remainder resulting from dividing the extended payload polynomial by the generator polynomial, or:

$$R(x) = (x^{77}P(x)) \bmod G(x)$$

The order of the remainder polynomial will be one less than that of the generator polynomial.

The parity bits shall have the value of the coefficients of the remainder polynomial $R(x)$, where the leftmost parity bit is the order 76 coefficient and the rightmost parity bit is the order 0 coefficient.

Table 5.18 Polynomials used for BCH code (127, 50, 13)

Primitive Polynomial	**Generator Polynomial**
$x^7 + x^6 + 1$	$x^{77} + x^{76} + x^{75} + x^{74} + x^{72} + x^{71} + x^{68} + x^{67} + x^{66} + x^{64} + x^{63} + x^{62} + x^{60} + x^{59} + x^{51} + x^{50} + x^{49} + x^{44} + x^{42} + x^{41} + x^{40} + x^{39} + x^{35} + x^{34} + x^{32} + x^{30} + x^{29} + x^{26} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{13} + x^{12} + x^9 + x^5 + x^2 + 1$

5.2.3 VP1 Payload

The syntax of the vp1_payload() structure is shown below in Table 5.19, Table 5.20 and Table 5.21.

Table 5.19 Syntax of payload() Structure

Syntax	No. of Bits	Format
vp1_payload() {		
domain_type	1	bslbf
If (domain_type == 0) {		
small_domain()	var	Table 5.20
}		
else {		
large_domain()	var	Table 5.21
}		
}		

Table 5.20 Syntax small_domain() Structure

Syntax	No. of Bits	Format
small_domain() {		
server_field	31	uimsbf
Interval_field	17	uimsbf
query_flag	1	bslbf
}		

Table 5.21 Syntax large_domain() Structure

Syntax	No. of Bits	Format
large_domain() {		
server_field	23	uimsbf
Interval_field	25	uimsbf
query_flag	1	bslbf
}		

domain_type – This 1-bit value specifies whether the payload information is for a small_domain() or a large_domain().

server_field – This field contains the server code.

interval_field – This field contains the interval code.

query_flag – This field signals when a dynamic event is available. A change in the value of this field between successive watermark payloads in a watermark segment indicates that a dynamic event (as defined in A/337 [7]) is available from the dynamic event HTTP server.

The range of the server code and interval code as a function of domain type is shown below in Table 5.22.

Table 5.22 Supported Range of Server Codes and Interval Codes per Domain Type

Domain Type	Server Code Range (hexadecimal)	Interval Code Range (hexadecimal)
Small Domain	00000000-7FFFFFFF	00000000-0001FFFF
Large Domain	00000000-007FFFFF	00000000-01FFFFFF

5.2.4 Display Override Indicator

Display override 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.

Display override condition shall be indicated by the use of inverse signaling in the audio watermark emission as defined in A/334 [5].

The display override 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 segment, cell, header, packet, or payload field boundaries.

5.2.5 Segment

An audio watermark segment is a continuously marked interval of content which contains VP1 payloads in contiguous cells with identical server codes and sequentially incrementing interval codes.

The query flag value may change between contiguous VP1 payloads in an audio watermark segment.

An audio watermark segment shall contain VP1 payloads in contiguous cells with identical server codes and sequentially incrementing interval codes.

The query flag value may change between contiguous VP1 payloads in an audio watermark segment.

5.2.6 Examples (Informative)

As an aid to calculating the specified BCH(127,50,13) code parity bits, examples bit sequences for valid cell() structures are shown below in Table 5.23. Note that the concatenation of the parity and payload form a valid BCH(127,50,13) codeword.

Table 5.23 Example cell() Sequences, Shown as Hexadecimal Values.

Header	Parity	VP1 Payload	Scrambled Parity	Scrambled Payload
AE0AB9E4	00000000000000000000	0000000000000	39BFE6D7B2212E120365	08428C02E0737
AE0AB9E4	1D9DD80E178D643E3225	0000000000001	01422ED9A5AC4A2C3140	08428C02E0736
AE0AB9E4	0CD1D8526D369D4A6D8E	1004B5A1C3B7F	100E2E85DF17B3586EEB	184639A323C48

5.3 Fingerprint Methodology

<Note: S33 may add a section to this document during CS Phase describing how content may be recovered in a redistribution scenario using fingerprints. If S33 adds this section, it is likely to normatively reference A/105 for certain aspects of this methodology. S33 will take into consideration the point that fingerprints can distinguish one piece of content from another but cannot distinguish different airings of the same piece of content.>

5.4 Audio User Data Methodology

<Note: S33 may add a section to this document during CS Phase describing how content may be recovered in a redistribution scenario using the "private user data" feature that is available in both Next Generation Audio systems proposed for ATSC 3.0. If this section is added, S33 is likely to define the data to be carried, which may be similar to the format of the data carried in the video watermark, and normatively reference A/342 for the data carriage mechanism.>

5.5 Recovery Process

A “redistribution setting” is a situation in which a device is receiving audio/video TV content via an input, such as an HDMI cable, without direct access to the ATSC broadcast stream from which the content is derived (e.g., a device getting TV content from a cable, satellite or IPTV set-top box). The “recovery process” is the process that a device in a redistribution setting goes through to get broadband access to supplementary content provided by the broadcaster of the content that the device is receiving via the cable.

There is one recovery process that a device in a redistribution setting can go through if it is receiving ATSC audio watermarks in the content it is receiving, or if is receiving VP1 messages in video watermarks. There is a different recovery process that such a device can go through if it is receiving a full set of ATSC video watermarks in the content it is receiving.

5.5.1 Recovery File Retrieval via Broadband

A recovery data request can be made to a recovery server by a receiver via issuance of a HTTP GET or HTTPS GET request for a resource specified by the URL constructed from a VP1 payload using the URL template:

```
http[s]://{hostName}/a336/rdt/{subdName}/{serverCode}-{intervalCode}.rdt
```

where:

`[string]` indicates that inclusion of the text `string` is optional and

`{element}` indicates the inclusion of the value of the named `element`.

The request shall use the prefix `http` when the query is employing HTTP. The request shall use the prefix `https` when the query is employing HTTPS.

If the domain_type of the VP1 payload is ‘0’, then `hostName` shall have the value:

```
a336.{serverCode1}.{serverCode2}.{serverCode3}.{serverCode4}.0.vp1.tv
```

`subdName` shall have the value:

```
{serverCode4}{serverCode3}/{serverCode2}/{serverCode1}
```

`serverCode` shall have the value:

```
{serverCode4}{serverCode3}{serverCode2}{serverCode1}
```

and if the domain_type of the VP1 watermark code is ‘1’, then `hostName` shall have the value:

```
a336.{serverCode1}.{serverCode2}.{serverCode3}.1.vp1.tv
```

`subdName` shall have the value:

```
{serverCode3}{serverCode2}/{serverCode1}
```

and `serverCode` shall have the value:

```
{serverCode3}{serverCode2}{serverCode1}
```

where `serverCode1`, `serverCode2`, `serverCode3`, and `serverCode4` shall respectively mean the least-to-most significant bytes of the server_field of the VP1 payload (zero-padded at the most-

significant bit to the next byte boundary) expressed as two-character uppercase-only hexadecimal values.

`intervalCode` shall be the value of the interval_field of the VP1 payload (zero-padded at the most-significant bit to the next byte boundary) expressed as an uppercase-only hexadecimal value (6 characters in length for VP1 payloads with domain_type=0 and 8 characters in length for VP1 payloads with domain_type=1).

DNS resolution of `{hostName}` to the unspecified IP address (0.0.0.0/32 in the case of an IPv4 address or ::/128 in the case of an IPv6 address) shall indicate that the recovery protocol is not supported for the code domain. Recovery file requests should not be issued to the unspecified address.

The response to a recovery data request shall contain a recovery file as set forth in Section 5.5.2 packaged together with zero or more signaling files delivered as a multi-part MIME message encapsulated in an MBMS "metadata envelope" as defined in Section 6.5 of A/331 [2]. If any signaling files are present, the metadata envelope shall include a "valid from" and a "valid until" and a "next URL" attribute associated with each signaling file. The "valid from" and "valid until" attributes define the interval of validity of the signaling file, and the "next URL" attribute is the URL of the next scheduled version of that signaling file.

5.5.2 Recovery File Format

The recovery file shall be a JSON document. The normative JSON schema for recovery file format shall be as specified in Annex B. Informative Table 5.24 describes the structure of the recovery file format in a more illustrative way. The specification following the table gives the normative semantics.

Note that there is a unique recovery file for each VP1 payload.

Table 5.24 Recovery File Format Logical Structure

Element or Attribute Name	Use	Data Type	Format
RecoveryDataTable	1		Root element
thisComponent	1		The media component from which the VP1 payload carrying `serverCode` and `intervalCode` was detected.
serverCode	0..1	integer	The serverCode value from the query request to which the recovery data table was provided as a response.
intervalCode	0..1	integer	The intervalCode value from the query request to which the recovery data table was provided as a response.
componentDescription	1	object	The description of this component, as defined in Table 5.25.
querySpread	0..1	integer	Time in milliseconds over which device should randomly choose time for querying Dynamic Event Server when signaled by the “query_flag” in the VP1 payload.
otherComponent	0..N	object	The audio or video components other than this component, which carries coincident VP1 payloads.
contentID	0..N		Content identifier
type	1	string	Defined values: “EIDR”, “Ad-ID”
cid	1	string	Either an EIDR (34-character canonical form with hyphens) or Ad-ID (12-character canonical form) string.
validFrom	1	string (date-time)	When `contentID` is valid from
validUntil	0..1	string(date-time)	Up to when `contentID` is valid until
sourceID	0..1		
country	1	string	ISO 3166-1 alpha-2 [11] country code associated with the primary administrative entity under which the given `bsid` is assigned.
bsid	1	integer	Identifier of the whole Broadcast Stream. The value of BSID shall be unique on a regional level (for example, North America). An administrative or regulatory authority may play a role.
majorChannelNo	1	integer	An integer number in the range 1 to 999 representing the “major” channel number of the service.
minorChannelNo	1	integer	An integer number in the range 1 to 999 representing the “minor” channel number of the service.
service	1		
serviceId	1	integer	Number that shall uniquely identify this service within the scope of `bsid.`
sltSvcSeqNum	1	integer	Version of service information in this table – increments by one for each new version of service data in RDT; wraps around to 0.
slsProtocol	0..1	integer	Protocol used to deliver the service layer signaling for this service (See Table 5.12).
slsMajorProtocolVersion	0..1	integer	Major version number of the protocol used to deliver the service layer signaling for this service.

slsMinorProtocolVersion	0..1	integer	Minor version number of the protocol used to deliver the service layer signaling for this service.
svcInetUrl	0..N		
urlType	1	unsignedByte	Type of files available with this URL (See Table 5.9)
urlValue	1	string (uri)	URL to access Internet signaling files for this service.

thisComponent – A description of the media component embedded with a watermark containing the VP1 payload containing serverCode and intervalCode.

serverCode – When present, this element shall provide the serverCode value employed in the HTTP request to which this recovery file was provided as a response.

intervalCode – When present, this element shall provide the intervalCode value from the query request to which the recovery data table was provided as a response.

componentDescription – A data element describing thisComponent in the format defined in Table 5.25 and the parameter descriptions that follow.

querySpread – When present, this element shall express the maximum duration that the receiver is recommended to delay submission of a dynamic event HTTP request, in units of 1 millisecond. Specification of a non-zero value expresses a request that upon each instance that the receiver detects a change in the value of the query_flag between successive VP1 payloads in an audio watermark segment or successive VP1 message blocks in a video watermark segment, the receiver is requested to delay the submission of an associated HTTP request for a dynamic event by an amount of time with a duration randomly selected with a probability uniformly distributed between 0 milliseconds and the specified value of querySpread, expressed in millisecond increments. The expectation is that the receiver will apply a small enough level of granularity to achieve an even spread of queries across the querySpread duration, such as 1 millisecond.

otherComponent – An element describing another watermarked media component associated with the same service as thisComponent in the format defined in Table 5.25 and the parameter descriptions that follow.

contentID – This field shall identify a content identifier.

contentID.type – A field that is required when contentId element is included. Two values are defined currently:

- "EIDR" indicates a content identification per the EIDR registry (http://eidr.org).
- "Ad-ID" indicates a content identifier per the Ad-ID registry (http://ad-id.org).

contentID.cid – A field that is required when contentId element is included that provides the content identification. The type of content identifier shall be as given in the contentID.type attribute. Either an EIDR (34-character canonical form with hyphens) or Ad-ID (12-character canonical form) can be included.

contentID.validFrom – A field that provides information about when the contentId is valid from.

contentID.validUntil – A field that provides information about up to when the contentId is valid until.

sourceID – An element describing a distribution an attributable distribution source that employs ATSC emission specifications. This element is applicable to circumstances where the

watermarked content is included in the redistribution of a service that is broadcast in accordance with ATSC specifications.

country – Country code associated with the primary administrative entity under which the value provided in bsid is assigned, using the applicable alpha-2 country code format as defined in ISO 3166-1 [11].

bsid – The BSID of the attributable ATSC distribution source.

majorChannelNo – The major channel number assigned to the attributable ATSC distribution source. This value is scoped to the BSID.

minorChannelNo – The minor channel number assigned to the attributable ATSC distribution source. This value is scoped to the BSID.

service – This element describes the service, its signaling formats and broadband locations.

serviceId – 16-bit integer that shall uniquely identify this Service within the scope of this Broadcast area.

sltSvcSeqNum – This integer number shall indicate the sequence number of the SLT service information with service ID equal to the serviceId attribute above. sltSvcSeqNum value shall start at 0 for each service and shall be incremented by 1 every time any attribute or child of this service element is changed. If no attribute or child element values are changed compared to the previous service element with a particular value of serviceID then sltSvcSeqNum shall not be incremented. The sltSvcSeqNum field shall wrap back to 0 after reaching the maximum value.

slsProtocol – Specifies the signaling format associated with this service, with permitted values and their meanings defined in Table 5.12.

slsMajorProtcolVersion – Major version number for the signaling protocol specified in slsProtocol.

slsMinorProtocolVersion – Minor version number for the signaling protocol specified in slsProtocol.

svcInetUrl – Base URL to access ESG or service level signaling files for this service via broadband, if available.

urlType – Type of files available with svcInetUrl. See Table 5.9 for values.

urlValue – URL to access Internet signaling files for this service identified by service identifier **serviceId**.

Table 5.25 Component Description Type

Element or Attribute Name	Use	Data Type	Format
componentDescription			
serverCode	0..1	integer	The server code of the component's VP1 watermarks. It is allowed to be present only for the otherComponent element.
componentAnchor	1		The mapping between the first VP1 payload in this watermark segment and its presentation time as defined in Table 5.26.
mediaType	1	string	The media type of this component. Three values are defined: "audio" indicates an audio component, "video" indicates a video component, and "both" indicates that serverCode and intervalCode are present in both audio and video content.
descriptor	0..1	string	Information associated with the component intended for consumption by an application.
priority	0..1	integer	The relative priority of the component that carries serverCode and intervalCode. Recovery data indicated in components with larger priority values shall take precedence over that of components with lower priority values. When no priority is indicated, the default priority of a component is 0.

componentDescription – A description of a watermarked media component associated with a service.

componentAnchor – Information about the first VP1 payload in the watermarked media component as defined in Table 5.26.

mediaType – A string with value "audio" to indicate that the description applies to an audio component only, "video" to indicate that the description applies to a video component only, or "both" to indicate that the description applies to both an audio and video component.

descriptor – An arbitrary descriptive string associated with the watermarked media component intended for consumption by an application.

priority – A numeric value indicating the relative priority of the described component. When a receiver finds watermarks in more than one media component in content that it receives, it is expected to perform the recovery process and utilize signaling associated with the component with the highest priority value. When no priority value is indicated for a component, its priority shall be 0.

Table 5.26 Component Anchor

Element or Attribute Name	Use	Data Type	Format
componentAnchor	1		
intervalCodeAnchor	1	integer	The interval code to be used as anchor for presentation time.
presentationTime	1	integer	Presentation time in seconds corresponding to IntervalCodeAnchoras the least-significant 32 bits of the count of the number of seconds since January 1, 1970 00:00:00, International Atomic Time (TAI).
presentationTimeMs	1	integer	milliseconds offset from the time indicated in presentationTime

componentAnchor – An element that specifies characteristics of the first VP1 payload in a video or audio watermark segment.

intervalCodeAnchor – The intervalCode (as defined in Section 5.5.1) in the first VP1 payload in a video or audio watermark segment.

presentationTime – The wall clock presentation time of the first frame of the first VP1 message block in the video watermark segment, or, for audio components, the wall clock presentation time of the first sample of the first symbol in the first cell of the audio watermark segment, in the form of a 32-bit count of the number of seconds since January 1, 1970 00:00:00, International Atomic Time (TAI).

presentationTimeMs –This integer in the range 0 to 999 shall indicate the milliseconds offset from the time indicated in presentationTime**,** such that the formula presentationTime + (presentationTimeMs / 1000) yields the actual presentation time to the nearest 1 millisecond.

In order for the presentation_time to be meaningful in a ROUTE/DASH service, the **MPD**@suggestedPresentationTime shall be present in the DASH MPD, and the resulting wall clock presentation time corresponding to the StartTime of each Period in the MPD shall be equal to the duration of the previous Period plus the wall clock presentation time corresponding to the StartTime of the Previous Period.

When the validUntil time of the last Content ID entry in the Content ID List approaches, the device can retrieve a new Recovery file to get an updated list.

If there is an unscheduled change to the Content ID List, or if there is a change to the Other Component list, the query flag in the VP1 watermark shall change state to indicate the availability of an Event to announce that a new Recovery file is available with the new information, as described in Section 5.5.3 below.

5.5.3 Dynamic Event Retrieval via Broadband

Availability of a dynamic Event is indicated by a change in the query_flag value between successive VP1 payloads within a watermark segment.

A dynamic Event request can be made to an Event server by a receiver via issuance of a HTTP GET or HTTPS GET request for a resource specified by the URL constructed from the VP1 payload whose query_flag value differs from the preceding payload using the URL template:

```
http[s]://{hostName}/a336/dyn/{subdName}/{serverCode}-{intervalCode}.dyn
```

where hostName, subdName, serverCode and intervalCode shall have the meanings given in Section 5.5.1.

DNS resolution of {hostName} to the unspecified IP address (0.0.0.0/32 in the case of an IPv4 address or ::/128 in the case of an IPv6 address) shall indicate that the recovery protocol is not supported for the code domain. Recovery file requests should not be issued to the unspecified address.

The response to the dynamic Event request shall contain a dynamic Event.

In the case when the Event is intended for an Application (i.e., the case when the schemeIdUri of the Event is not recognized as the schemeIdUri for an Event defined in a standard and directed to the device itself), it should be passed into any Application in the runtime environment that has registered a listener for it, as specified in A/344 [15].

In the case when the Event announces the availability of one or more updated signaling files, the Event may be delivered alone, or it may be packaged along with one or more signaling files as a multi-part MIME message encapsulated in an MBMS “metadata envelope” as defined in Section 6.6 of A/331 [2].

The format of the dynamic Event shall be as defined in Table 5.10 of this document.

The attributes of an Event that announces updates to signaling files (the `schemeIdUri`, value, etc.) are specified in A/337 [7], with one addition. The abbreviation "RRF" shall be added to the list of signaling tables for which updates can be announced, standing for "Redistribution Recovery File."

Annex A: Implications of Set-top Box Operations

<Note: S33 intends to complete this section during CS Phase. This section will describe various operations that may occur on the STB that may impact a video and/or audio watermark, such as muting or displaying the Program Guide. This will be an informative section describing what the watermark detector might encounter and what the implications might be.>

Annex B: JSON Recovery File Schema

Normative JSON schema for recovery file format is as follows:

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
   "id": "http://atsc.org/version/3.0/wm/recoveryfileformat#",
   "title": "Recovery file format schema",
   "description": "Return messages from server use this schema.",
   "@context": {"RecoveryDataTable": "http://www.atsc.org/contexts/3.0/RecoveryFFV1"},
   "RecoveryDataTable": {
      "type": "object",
      "properties": {
         "thisComponent": {
            "type": "object",
            "properties": {
               "serverCode": {"type": "integer"},
               "intervalCode": {"type": "integer"},
               "componentDescription": {
                  "type": "object",
                  "properties": {
                     "componentAnchor": {
                        "type": "object",
                        "properties": {
                           "intervalCodeAnchor": {
                              "type": "integer",
                              "minimum": 0,
                              "maximum": 33554431
                           },
                  "presentationTime": {"type": "integer", "minimum": 0, "maximum": 4294967295},
                  "presentationTimeMs": {"type": "integer", "minimum": 0, "maximum": 999}
                        },
                        "required": [
                           "intervalCodeAnchor",
                           "presentationTime",
                           "presentationTimeMs"
                        ]
                     },
                     "mediaType": {
                        "type": "string",
                        "enum": [
                           "audio",
                           "video",
                           "both"
                        ]
                     },
                     "descriptor": {
                        "type": "string"
},
                     "priority": {
```

```
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 255
                }
            },
            "required": [
                "componentAnchor",
                "mediaType"
            ]
        }
    },
    "required": ["componentDescription"]
},
"querySpread": {"type": "integer"},
"otherComponent": {
    "type": "object",
    "properties": {
        "componentAnchor": {
            "type": "object",
            "properties": {
                "intervalCodeAnchor": {
                    "type": "integer",
                    "minimum": 0,
                    "maximum": 33554431
                },
               "presentationTime": {"type": "integer", "minimum": 0, "maximum": 4294967295},
               "presentationTimeMs": {"type": "integer", "minimum": 0, "maximum": 999}
                    },
                    "required": [
                        "intervalCodeAnchor",
                        "presentationTime",
                        "presentationTimeMs"
                    ]
         },
        "mediaType": {
            "type": "string",
            "enum": [
                "audio",
                "video",
                "both"
            ]
        },
        "Descriptor": {
            "type": "string"
             },
        "priority": {
            "type": "integer",
            "minimum": 0,
            "maximum": 255
        }
    },
    "required": [
        "componentAnchor",
        "mediaType"
    ]
}
```

```
    },
    "contentID": {
       "type": "array",
       "items": {
          "type": "object",
          "properties": {"oneOf": [
             {
                "Type": {
                   "type": "string",
                   "enum": ["EIDR"]
                },
                "cid": {
                   "type": "string",
                   "pattern": "^10\\.5240\\/([0-9a-fA-F]{4}-){5}[0-9A-Z]$",
                   "minLength": 34,
                   "maxLength": 34
                },
                "validFrom": {
                   "type": "string",
                   "format": "date-time"
                },
                "validUntil": {
                   "type": "string",
                   "format": "date-time"
                }
             },
             {
                "type": {
                   "type": "string",
                   "enum": ["AD-ID"]
                },
                "cid": {
                   "type": "string",
                   "pattern": "^[1-9a-zA-Z]{1}[0-9a-zA-Z]{10}(H|D)?$",
                   "minLength": 11,
                   "maxLength": 12
                },
                "validFrom": {
                   "type": "string",
                   "format": "date-time"
                },
                "validUntil": {
                   "type": "string",
                   "format": "date-time"
                }
             }
          ]}
       },
       "minItems": 0
    },
  "sourceID": { "type": "object",
             "properties": {
                "country": {"type": "string","pattern": "^[a-zA-Z]{2}$"},
                "bsid":  { "type": "integer","minimum": 0, "maximum": 65535 },
                "majorChannelNo":{"type": "integer","minimum": 1, "maximum": 999 },
                "minorChannelNo":{"type": "integer","minimum": 1, "maximum": 999 }
```

```
            },
            "required": ["country","bsid","majorChannelNo","minorChannelNo"]
        },
"service": {
        "type": "object",
        "properties": {
            "serviceId": {
                "type": "integer",
                "minimum": 0,
                "maximum": 65535
            },
            "sltSvcSeqNum": {
                "type": "integer",
                "minimum": 0,
                "maximum": 255
            },
            "slsProtocol": {
                "type": "integer",
                "minimum": 0,
                "maximum": 255
            },
            "slsMajorProtocolVersion": {
                "type": "integer",
                "minimum": 0,
                "maximum": 255
            },
            "slsMinorProtocolVersion": {
                "type": "integer",
                "minimum": 0,
                "maximum": 255
            },
            "svcInetUrl": {
             "type": "array",
             "items": {
                "type": "object",
                "properties": {
                    "urlType": {
                        "type": "integer",
                        "minimum": 0,
                        "maximum": 255
                    },
                    "urlValue": {
                        "type": "string",
                        "format": "uri"
                    }
                },
                "required": [
                    "urlType",
                    "urlValue"
                ]
            }
        },
        "required": [
            "serviceId",
            "sltSvcSeqNum"
        ]
```

```
            }
        },
        "required": ["thisComponent","service"],
        "additionalProperties": true
    },
    "required": ["RecoveryDataTable"]
}
```

Annex C: Acquisition of Signaling Using Video Watermarks

The following is a summary of the signaling acquisition process using video watermarks:

- The device acquires video watermarks with a Channel ID payload and a signaling URL payload.
- The device uses the signaling URL from the signaling URL payload to obtain the signaling files in the form of a multi-part MIME message, exactly as it would use the signaling URL in the audio watermark recovery file.
- The device uses the signaling files to access and present the supplementary content using the timing information from the Channel ID payload to synchronize the content with the audio and video coming from the cable.
- Each signaling file has a `validFrom` and a `validUntil` and a `nextURL` attribute associated with it. The `validFrom` and `validUntil` fields define the interval of validity of the signaling file, and the `nextURL` is the URL of the next scheduled version of that signaling file. Thus, the device can get scheduled updates to the signaling files as needed.
- From time to time dynamic events may appear in a dynamic event payload of a video watermark.
- Such an event can be intended for an application running in the run-time environment, or it can be signaling the availability of an unscheduled update to a signaling file. In the former case the device can just make the event available to any application that has registered a callback routine for it, in the usual way. In the latter case the event will contain the URL of the updated signaling file (or files), and the device can retrieve it (or them). The file or files will be in the same format as files retrieved by a signaling URL, with `validFrom` and `validUntil` and `nextURL` attributes attached to them.
- If at any time the video watermarks disappear, the device is expected to terminate presentation of any supplementary content, interpreting the disappearance as evidence of a channel change to a new service that does not support any supplementary content.
    - Note that if audio watermarks are also present and have not simultaneously disappeared, this may be an indication of the presence of graphics generated by the set-top box, such as displaying the MVPD program guide, rather than a channel change. Please see Annex A for more information.

If at any time no Channel ID watermark appears, or a channel ID watermark appears with a new combination of BSID and channel number, or there is a discontinuity in the media time, then the device will terminate presentation of any supplementary content, and start the signaling acquisition over again.

Annex D: Acquisition of Signaling Using VP1 Payloads/Messages

The following is a summary of the watermark-based signaling acquisition process via broadband, starting from a VP1 audio watermark payload or VP1 video watermark message:

- The device recovers the VP1 payload from the audio or video watermark.
- The device makes an HTTP request to a recovery file server, using the domain type, server field and the interval field of the VP1 payload to form the URL for the request (defined in Section 5.5.1).
- The device receives in response a "recovery file" (defined in Section 5.5.2) that includes the original presentation time of the content at the start of the VP1 payload, information about the service being viewed, and a URL that can be used to retrieve the set of signaling files needed to access and present the supplementary content. These signaling files will be current as of the presentation time of the VP1 payload.
- The device uses the URLs in the recovery file to retrieve the set of signaling files –
  - MPD for ROUTE/DASH services and for MMT services with supplementary streaming content available via Internet
  - AST for services that include App-based Feature available via Internet
  - AEI for MMT services
  - MPT for MMT services
- The device uses the signaling files to access and present the supplementary content, using the timing information in the recovery file to synchronize the content with the audio and video coming from the cable. The "query" flag in the VP1 payload is used to signal the availability of an event. When such a flag is detected, the device can request the event, using information from the VP1 payload (defined in Section 5.5.2).
- The recovery file, dynamic event, and signaling files are delivered as a multi-part MIME message encapsulated in an MBMS "metadata envelope" (defined in Section 6.5 of A/331 [2]) that includes a "valid from" and a "valid until" and a "next URL" attribute associated with each signaling file. The "valid from" and "valid until" attributes define the interval of validity of the signaling file, and the "next URL" attribute is the URL of the next scheduled version of that signaling file. Thus, the device can get scheduled updates to the signaling files as needed.
- Such an event can be intended for an application running in the run-time environment, or it can be signaling the availability of an unscheduled update to a signaling file. In the former case the device can just make the event available to any application that has registered a callback routine for it, in the usual way. In the latter case the event will contain the URL of the updated signaling file (or files), and the device can retrieve it (or them).

- If at any time the audio watermarks disappear, the device will terminate presentation of any supplementary content, interpreting the disappearance as evidence of a channel change to a new service that does not support any supplementary content.
  - Note that if video watermarks are also present and have not simultaneously disappeared, this may be an indication of muting at the set-top box, rather than a channel change. Please see Annex A for more information.
- If at any time the server field in the audio watermarks changes, or the interval field has a discontinuity, the device will terminate presentation of any supplementary content, request a new recovery file and restart the recovery process.
- In the case of a ROUTE/DASH-based service, the original presentation time in the recovery file of the content at the start of the watermark will be relative to the media presentation timeline of the current MPD for the service. In the case of an MMT-based service, the original presentation time in the recovery file of the content at the start of the watermark will be NPT time.

— End of Document —