

**Advanced Television Systems Committee**

1776 K Street, N.W.

Washington, D.C. 20006

202-872-9160

S38-153r3

23 January 2020

A/344:2019 Revision Change Log

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards and recommended practices for digital television. ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops digital television implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society 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). For more information visit [www.atsc.org](http://www.atsc.org).

|  |
| --- |
| *Note*: The user's attention is called to the possibility that compliance with the standard affected by these changes may require use of an invention covered by patent rights. By publication of the corresponding 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 |
| Candidate Standard approved | 20 January 2020 |

Table of Contents

[1. SCOPE 4](#_Toc30676607)

[1.1 Introduction and Background 4](#_Toc30676608)

[1.2 Organization 4](#_Toc30676609)

[2. User Agent Reference Updates 4](#_Toc30676610)

[2.1 Scope 4](#_Toc30676611)

[2.2 Rationale for Changes 4](#_Toc30676612)

[2.3 Compatibility Considerations 5](#_Toc30676613)

[2.4 List of Changes 5](#_Toc30676614)

[3. Remove DRM JSON Examples 6](#_Toc30676615)

[3.1 Scope 6](#_Toc30676616)

[3.2 Rationale for Changes 6](#_Toc30676617)

[3.3 Compatibility Considerations 6](#_Toc30676618)

[3.4 List of Changes 6](#_Toc30676619)

[4. Clarify Broadcast v. Broadband 10](#_Toc30676625)

[4.1 Scope 10](#_Toc30676626)

[4.2 Rationale for Changes 10](#_Toc30676627)

[4.3 Compatibility Considerations 10](#_Toc30676628)

[4.4 List of Changes 10](#_Toc30676629)

A/344:2019 Revision Change Log

# SCOPE

This document describes one or more changes to the ATSC 3.0 Interactive Content candidate standard. These changes have been accepted by the S38 Specialist Group on Interactive Content but are pending full TG3 and membership approval. Readers are cautioned that these changes may be amended in the future but are encouraged to provide feedback and comments.

## Introduction and Background

This document describes changes or updates to the ATSC 3.0 Interactive Content Candidate Standard. Each section identifies a single change including the scope, rationale, and backwards compatibility of the change along with the instructions for making the change to the baseline A/344 specification.

For each of the subsections below, unless otherwise noted, editing instructions are given in italics, inserted text, tables, and drawings are shown in red; 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. Where possible, the reference will be to a reference existing in the current standard but highlighted in red. It is often the case that these references will not need to be changed but should be given special attention to verify correctness.

The A/344 standard maintains a “revision log” of its included APIs from revision to revision by listing the changes in Table 9.1. In addition, each revision includes an Annex which captures the API from the previous edition in unchanged form. By maintaining the previous API definition in the document, implementers may look at the history of each API.

## Organization

This document is organized as follows:

* Section 1 – The scope, introduction, and background of this specification
* Section 2 – User Agent Reference Updates
* Section 3 – Remove DRM JSON Examples
* Section 4 – Clarify Broadcast v. Broadband

# User Agent Reference Updates

## Scope

This change provides an update to Section 5.2 and corresponding references to CTA-5000-A (WMAS 2018).

## Rationale for Changes

The published version of the A/344 User Agent requirements, specifically, Section 5.2, incorrectly references CTA 5000. This amendment describes the changes necessary to address the improper references. Furthermore, CTA 5000 has been updated as part of a yearly CTA WAVE process and this contribution reconciles the new revision with A/344:2019.The CTA WAVE group has reviewed Section 5.2 and pointed out discrepancies between what is referenced and what is provided by the CTA-5000 specification. This contribution addresses these discrepancies.

## Compatibility Considerations

The changes described in this document address referencing peculiarities that make complying with the standard difficult. Implementors cannot determine the appropriate reference material with the present standard so must do research regarding the intent and the appropriate reference. Therefore, any implementations will be compatible with the new, correct references described by this document.

## List of Changes

*The following indicate changes to the references in Section 2 and User Agent Definitions in section 5.2.*

2.1 Normative References

[7] CTA: “CTA Specification: Web Application Video Ecosystem – Web Media API Snapshot ~~2017~~2018”, Doc. CTA-5000-A, Consumer Technology Association, December ~~2017~~2018.

5.2 User Agent Definition

Receivers shall implement an HTML5 User Agent that complies with all normative requirements specified in the CTA Web Media API Snapshot (CTA-5000-A) [7]. In addition, the features described in the following sections shall be supported.

5.2.1 HTTP Protocols

The User Agent shall implement the HTTP protocols specified in RFC 7230 through RFC 7235, references [10], [11], [12], [13], [14] and [15]. User Agents shall implement the Web Origin Concept specification [19] and the HTTP State Management Mechanism specification (Cookies) ([7] ~~§~~Section 4.2) as well. ~~These are referenced in [7] as [HTTP], [ORIGIN], and [COOKIES].~~

5.2.2 XMLHttpRequest (XHR)

The User Agent shall support the XMLHttpRequest and related interfaces of the [XHR] reference in [7]. In the case of an XHR request where the request URL identifies a broadcast resource, the request is delivered to the Receiver Web Server, rather than to an Internet web server.

5.2.3 Cross-Origin Resource Sharing (CORS)

The User Agent shall support the [ORIGIN] specification referenced in [19]~~[7]~~.

5.2.4 Mixed Content

The User Agent shall handle fetching of content over unencrypted or unauthenticated connections in the context of an encrypted and authenticated document according to the W3C Mixed Content specification [27] though Broadcaster Applications are encouraged to only reference trusted content. References to files within the Application Context Cache (see Section 5.3 below) shall be considered to be “a priori authenticated” in the terminology of W3C Mixed Content. Any resource accessed from the Application Context Cache shall be considered to have been accessed within a secure context.

5.2.5 Transparency

The background of the User Agent’s drawing window might be transparent by default, nevertheless, it is recommended that Broadcaster Applications explicitly specify the areas desired to be opaque or transparent to maintain consistency across Receivers. Thus, for example, if any element in the web page (such as a table cell) includes a CSS style attribute "background-color: transparent", and that area is not covered by another layer with an opaque element, then video content presented by the Receiver Media Player (see Section 4.2) might be visible in that area. Note that certain areas can be specified as transparent while others are opaque.

5.2.6 Full Screen

As stated in Section 4.2, a Broadcaster Application can expect that the User Agent graphics window, 0,0 to a full 100% in both axes, maps directly to the RMP logical video display window at its full dimensions. The “width” media feature of CSS MediaQuery [7] shall align with the width of the RMP logical video display window. In most viewing conditions, the RMP logical video display window will fill the entire screen.

5.2.7 Visibility and Focus

The Receiver shall use the W3C Page Visibility Level 2 API as required by CTA-5000-A [7] to inform the Broadcaster Application whether its display output is visible or not. The Receiver may choose to obscure or mute the Broadcaster Application display output for a variety of reasons including but not limited to display of Receiver preference dialogs, content blocking, or emergency information display. Similarly, the Receiver shall ~~use~~ provide the W3C Focus ~~APIs~~ Events as required by CTA-5000-A [UIEvents] [7] as well as the standard DOM activeElement property to allow the Broadcaster Application to determine if it can receive user input or not.

# Remove DRM JSON Examples

## Scope

This change removes the confusing examples from Section 9.1.5.

## Rationale for Changes

The examples provided in the DRM Section 9.1.5 use properties with values in data blocks that are not specified. This is consistent with the text but is leading some readers to believe that the attributes are required or applicable. Removing the examples removes the issue.

## Compatibility Considerations

There are no backwards compatibility issues since the removed text is all informative.

## List of Changes

*The following indicate changes to the references in Section 2 and User Agent Definitions in section 5.2.*

9.15 DRM APIs

The APIs in this section can be used by the Broadcaster Application to support the RMP or AMP handling of encrypted content. Two generic APIs are defined. A “notification” API is used by the Receiver to pass a message associated with an identified DRM System to the Broadcaster Application. An “operation” API is used by the Broadcaster Application to pass a message associated with an identified DRM System to the Receiver. These APIs support the needs of both the AMP and the RMP.

9.15.1 DRM Notification API

The DRM Notification API may be issued by the Receiver to the Broadcaster Application in order to deliver a DRM-related notification. A Broadcaster Application which receives this notification can use the DRM Operation API, defined in Section 9.15.2, to exchange a message with the Receiver’s underlying content protection system, ultimately resulting in the delivery of the license/key required by the RMP or AMP for decryption of protected content.

The DRM Notification API is defined as follows:

method: "org.atsc.notify"

params: A JSON object containing three key/value pairs.

params JSON Schema:

|  |
| --- |
| {  "type": "object",  "properties": {  "msgType": {"type": "string", "enum": ["DRM"]},  "systemId": {"type": "string"},  "message": {"type": "array", "items": {"type":"object"}},  },  "required": ["systemId", "message"] } |

msgType – This required string shall be set to "DRM" to identify this notification API.

systemId – This string shall be set to a DRM system identifier, @schemeIdUri, as defined in the DASH-IF IOP, section 7.6 [31]. For example, the UUID string value “1077efec-c0b2-4d02-ace3-3c1e52e2fb4b” corresponds to the common system ID of W3C EME.

message –This shall be the message passed from the content protection system formatted as an array of JSON objects.

The DRM Notification API may be used by the Receiver to notify the Broadcaster Application that a particular DRM license object that had been requested via the DRM Operation API (Section 9.15.2) has been retrieved from the broadcast. ~~In the following example, the message object includes key/value pairs providing the URI of the requested license file, and the contents of the retrieved file. The key/value pairs within the message object (in this example, the keys are licenseUri and licenseData) are as specified by the DRM system identified in the systemId key:~~

|  |
| --- |
| ~~<-- {~~  ~~"jsonrpc": "2.0",  "method": "org.atsc.notify",  "params": {  "msgType": "DRM",  "systemId": "urn:uuid:1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",  "message": [{"~~*~~licenseUri~~*~~":"mybroadcast/myDRM/mysubID/license.dat",  "~~*~~licenseData~~*~~":"ZkxhQwAAACIQABAAAAUJABtA…"}]  } }~~ |

~~As another example, the Receiver may notify the application if the base layer of video content is playing properly but a decryption key is needed for the enhancement layer of a spatial scalable video codec, e.g. SHVC.~~

|  |
| --- |
| ~~<-- {  "jsonrpc": "2.0",  "method": "org.atsc.notify",  "params": {  "msgType": "DRM",  "systemId": "urn:uuid:1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",  "message": [{"~~*~~<proprietary>~~*~~":"~~*~~<proprietary>~~*~~"}, {…}]  } }~~ |

It is out of scope of the present specification how a Broadcaster Application interacts with a license server and the exact format of messages passed from the content protection system to the Broadcaster Application using this DRM Notification API. The format of messages passed across the interface defined in this API depends on the content protection system used by the broadcaster.

9.15.2 DRM Operation API

The DRM Operation API can be issued by a Broadcaster Application to pass a message to the Receiver in order to play protected content as defined by Section 5.7 of [6]. This API can be used along with the DRM Notification API as defined in Section 9.15.1 which is issued by the Receiver to notify a message to a Broadcaster application in order to inform information about content protection.

Similar to W3C EME [24] which is an extension of the HTML5 media element, a Broadcaster Application can communicate with a license server and pass messages for license/key exchange to the underlying content protection system via this API. These APIs are simpler than W3C EME since the APIs only provide a message exchange mechanism and the content of the messages conveyed in the API, including any control such as installing/updating/removing licenses, are specific to an underlying content protection system and thus not specified here. Note that a Broadcaster Application needs to know the sequence of interactions with a broadcaster’s web server and a license server and also the procedure for exchange of messages with the underlying content protection system of the Receiver.

The DRM Operation API shall be defined as follows:

method: "org.atsc.drmOperation"

params: A JSON object consisting of systemId and message properties as follows.

params JSON Schema:

|  |
| --- |
| {"type": "object",  "properties": {  "systemId": {"type": "string"},  "service": {"type": "string"},  "message": {"type": "array", "items": {"type":"object"}}  },  "required": ["systemId", "message", "message"] } |

systemId – This string shall be set to a DRM system identifier, @schemeIdUri, as defined in the DASH-IF, section 7.6 [39].

service – This required string shall indicate the globally unique Service ID associated with the currently selected service as given in the SLT in [SLT.Service@globalServiceID](mailto:SLT.Service@globalServiceID) to which this DRM message is associated.

message – This shall be a message specific to the content protection system in use formatted as an array of JSON objects.

Response:

result: The following JSON object is returned upon success.

result JSON Schema:

|  |
| --- |
| {  "type": "object",  "properties": {  "message": {"type": "array", "items": {"type":"object"}}  } } |

error: The following error codes may be returned:

* + - -14: The specified content protection system is not supported by the Receiver.

message – This shall be the response to the command formatted as an array of JSON objects specific to the content protection system in use.

In the case of the AMP, when the DRM Operation API is used by the Broadcaster Application to ask the Receiver to fetch a particular DRM license file from the broadcast Entitlement Management service, the message object may include key/value pairs to specify the Content-Location of the file.~~, e.g. mybroadcast/myDRM/mysubID/license.dat in the following example:~~

|  |
| --- |
| ~~--> {  "jsonrpc": "2.0",  "method": "org.atsc.drmOperation",  "params": {  "systemId": "urn:uuid:1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",  "service": "https://doi.org/10.5239/8A23-2B0",  "message": {  "operation": "licenseMessageRequest",  "licenseUri": "mybroadcast/myDRM/mysubID/license.dat"  }  },  "id": 101 }~~ |

~~Upon successfully accepting the request to fetch this file, the Receiver may respond with:~~

|  |
| --- |
| ~~<-- {  "jsonrpc": "2.0",  "result": {},  "id": 101 }~~ |

~~As another example, the Broadcaster Application can pass a proprietary message for the DRM system corresponding to the UUID 1077efec-c0b2-4d02-ace3-3c1e52e2fb4b by issuing a command to the Receiver as follows:~~

|  |
| --- |
| ~~--> {  "jsonrpc": "2.0",  "method": "org.atsc.drmOperation",  "params": {  "systemId": "urn:uuid:1077efec-c0b2-4d02-ace3-3c1e52e2fb4b",  "message": [{"~~*~~<proprietary>~~*~~":"~~*~~<proprietary>~~*~~"},{…}]  },  "id": 104 }~~ |

If the Receiver supports this DRM system, it could respond without an error:

|  |
| --- |
| <-- {  "jsonrpc": "2.0",  "result": {},  "id": 104 } |

If the Receiver does not support this DRM system, it could respond with an error message:

|  |
| --- |
| <-- {  "jsonrpc": "2.0",  "error": {"code": -14, "message": "The specified content protection system is not supported by the Receiver"},  "id": 104 } |

# Clarify Broadcast v. Broadband

## Scope

This change is to a normative statement in Section 6.4.1.

## Rationale for Changes

The document is not always clear with language regarding broadband delivery of applications and data. NRT is often used with broadband where it should be both broadcast and broadband.

## Compatibility Considerations

While this change does modify a normative statement, the addition only clarifies the intent and does not change any expected behavior of Receivers and Broadcaster Applications.

## List of Changes

*In the first sentence of Section 6.4.1 add words as follows:*

The file components comprising the Broadcaster Application shall be delivered over broadcast within one or more multi-part MIME packages using ROUTE or over broadband as individual files using HTTPS. All files made available through the Receiver Web Server shall be delivered to the receiver as signed packages as described in A/331[1].

— End of Document —