# Adobe Primetime HTTP Live Streaming Profile: April 2014

## Adobe® Primetime HTTP Live Streaming Profile: April 2014

Adobe Primetime HTTP Live Streaming Profile

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This guide contains links to third-party websites that are not under the control of Adobe Systems Incorporated, and Adobe Systems Incorporated is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Adobe Systems Incorporated provides these links only as a convenience, and the inclusion of the link does not imply that Adobe Systems Incorporated endorses or accepts any responsibility for the content on those third-party sites. No right, license, or interest is granted in any third party technology referenced in this guide.

Updated Information/Additional Third Party Code Information available at http://www.adobe.com/go/thirdparty.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Published April 2014.

## Abstract

The Adobe Primetime HTTP Live Streaming Profile (PT-HLS) is a set of extensions to the HTTP Live Streaming (HLS) protocol that enable premium video use cases, such as Adobe Access protected content, advertising content, and ad signaling. PT-HLS provides additional definitions, requirements, and recommendations regarding the playlist file format, client behavior, and server behavior.

This document describes version 1 of PT-HLS.

# 1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

# 2. Introduction

This document is a specification of the Adobe Primetime HTTP Live Streaming profile (Primetime HLS), a set of extensions to the HTTP Live Streaming protocol. Primetime HLS enables premium video use cases such as:

- Protection of content with Adobe Access digital rights management

- Inclusion of advertising content and metadata
- Inclusion of ad cueing information.
- Interoperability with Adobe Primetime components

As of this document's publication the most recent version of Primetime HLS is version 1. Unless otherwise noted, this specification defines requirements that apply to playlist files, clients, and servers of Primetime HLS version 1.

Future publications of this document will supersede all requirements defined in this document. Always reference the latest version of the specification.

# 3. Basics

For clarity, the terms "HLS playlist", "HLS master playlist", "HLS media playlist", "HLS client", and "HLS server" refer to Playlist files, Master Playlists, Media Playlists, Clients, and Servers that conform to [I-D.pantos-http-live-streaming].

The terms "Primetime playlists", "Primetime master playlists", "Primetime media playlists", "Primetime clients", and "Primetime servers" refer to HLS playlists, HLS master playlists, HLS media playlists, HLS clients, and HLS servers that conform to the Primetime HLS protocol. Unless otherwise noted, the playlist, client, and server requirements in this document apply to Primetime playlists, clients, and server.

Unless otherwise noted, the terms "playlist" and "playlist file" have identical meanings in the scope of this document.

Primetime playlists, clients, and servers SHALL conform to HLS versions 1, 2, 3, or 4.

# 4. EXT-X-ADOBE Tag

The EXT-X-ADOBE tag advertises that an HLS (media or master) playlist is a Primetime playlist.

This tag SHOULD be present in a Primetime (media or master) playlist. This tag SHALL be present in an HLS playlist if the playlist supports any Primetime HLS version after version 1.

All variant playlists of a Primetime master playlist SHALL be Primetime media playlists that specify the same EXT-X-ADOBE tag value as the master playlist.

This tag SHALL appear at most once per file and SHALL apply to the entire file. If present, the tag SHALL appear on a line that follows the EXTM3U tag and is prior to any line whose default meaning would be affected by the tag.

The tag value is an attribute list where the following attributes are defined:

VERSION

This REQUIRED, enumerated-string attribute specifies a Primetime HLS version supported by the HLS playlist. The playlist and the server SHALL conform to all requirements of the Primetime HLS version specified in this attribute.

The attribute value is a version name that identifies a particular Primetime HLS version. The version name of Primetime HLS version 1 is "1". No other values are defined as of the publishing of this specification.

When the EXT-X-ADOBE tag is present, a Primetime client SHALL only act in accordance with the requirements of the specified Primetime HLS version. If the client does not support the specified version, it SHALL NOT attempt to play the content.

When the tag is missing, a Primetime client MAY assume an HLS playlist supports Primetime HLS version 1.

# 5. Signaling digital program insertion

A Primetime media playlist MAY contain digital program insertion signaling information that conforms to [DPI].

# 6. Advertisement content

This section describes how a Primetime media playlist can denote portions of its content as advertisements.

## 6.1. Ad, midroll, and preroll intervals

An "ad interval" is a time interval of a Primetime media playlist whose content is a single advertisement. A Primetime media playlist MAY contain multiple ad intervals. An ad interval SHALL NOT overlap any other ad interval in the same playlist.

A "midroll interval" is a time interval of a Primetime media playlist that represents an ad break that occurs midstream. A Primetime media playlist MAY contain multiple midroll intervals. A midroll interval MAY contain ad intervals. A midroll interval SHALL NOT overlap any other midroll interval in the same playlist.

A "preroll interval" is a time interval of a Primetime media playlist that represents an ad break intended to be played prior to primary content. A Primetime media playlist SHALL contain at most one preroll interval. A preroll interval MAY contain ad intervals. A preroll interval SHALL start at the start time of the first segment advertised by the enclosing playlist. A preroll interval SHALL have a known end time that is less than the current end time of the last segment advertised by the enclosing playlist; That is, a preroll interval SHALL NOT be open ended. A preroll interval SHALL NOT overlap any midroll interval in the same playlist.

An ad interval SHALL be fully contained within either a single preroll or midroll interval. Midroll intervals and preroll intervals MAY include spans of time that are not covered by an ad interval.

In a Primetime master playlist, the ad, midroll, and preroll intervals SHOULD be logically identical across the variant playlists, since the primary content and advertising content across renditions is typically identical. However, a Primetime client SHALL continue playback when faced with inconsistencies across variant playlists, since minor inconsistencies are inevitable in production deployments. For example, a robust client could assume that, when switching renditions, intervals seen in previous renditions are implicitly present in the current rendition.

## 6.2. EXT-X-MARKER tag

The EXT-X-MARKER tag advertises a significant point in the timeline. The main usage for this tag is to indicate the begin and end points of ad, midroll, and preroll intervals. See Section 6.3 "Describing intervals with the EXT-X-MARKER tag".

A Primetime media playlist MAY contain this tag. A Primetime media playlist MAY contain multiple instances of this tag. A Primetime master playlist SHALL NOT contain this tag.

The tag value is an attribute list where the following attributes are defined:

ID

This REQUIRED, quoted-string attribute associates a unique identifier with the tag. All EXT-X-MARKER tags in a given Primetime media playlist file SHALL have different ID values.

ID uniqueness for the lifetime of a playlist is recommended but not guaranteed; If the server removes an EXT-X-MARKER tag from a Primetime media playlist, the ID value of the removed tag SHOULD NOT be used to identify any EXT-X-MARKER tag in playlist versions that occur after the removal. For robustness, clients are expected to correctly handle reuse of removed EXT-X-MARKER ID values; When the client refreshes a playlist, the client SHALL NOT consider an EXT-X-MARKER tag from the new playlist version as equivalent to an EXT-X-MARKER tag from the old playlist unless the tags share the same attribute set, share the same attribute values, and reference segments with the same sequence number.

TYPE

This REQUIRED, enumerated-string attribute associates a type with the tag. The following values are defined:

AdBegin - the beginning of an ad interval

AdEnd - the ending of an ad interval

PodBegin - the beginning of a midroll interval.

PodEnd - the ending of a midroll interval.

PrerollPodBegin - the beginning of a preroll interval.

PrerollPodEnd - the ending of a preroll interval.

To aid forward compatibility, a Primetime client SHALL ignore any tags whose TYPE it does not understand.

OFFSET

This OPTIONAL, decimal-floating-point attribute is used to associate a time with the tag. A tag's time is the next segment's start time plus the OFFSET value. The units of the value are seconds. When this attribute is missing, a tag's time is the next segment's start time.

For example, if the next segment after an EXT-X-MARKER tag has an absolute start time of 100 seconds and the OFFSET of the EXT-X-MARKER tag is 5, the tag's time is an absolute time of 105 seconds.

DURATION

This REQUIRED, decimal-floating-point attribute associates a duration with the tag. The units of the value are seconds. A value of 0 indicates that the no duration is associated with the tag.

DATA

This OPTIONAL, quoted-string attribute indicates data to be associated with the tag.

If TYPE is "AdBegin" or "AdEnd", the data SHALL be formatted according to Section 6.5 "VAST Fragments data format".

If TYPE is "PrerollPodBegin", "PrerollPodEnd", "PodBegin", or "PodEnd", the data SHALL be formatted according to Section 6.4 "VMAP Fragments data format".

## 6.3. Describing intervals with the EXT-X-MARKER tag

This section describes how a Primetime media playlist advertises ad, midroll, and preroll intervals using the EXT-X-MARKER tag.

A Primetime media playlist advertises ad intervals as follows:

> The boundaries of ad intervals are indicated by EXT-X-MARKER tags with TYPE values of "AdBegin" or "AdEnd". An "ad begin tag" is an EXT-X-MARKER tag whose TYPE value is "AdBegin". An "ad end tag" is an EXT-X-MARKER tag whose TYPE value is "AdEnd". An "ad marker tag" is an EXT-X-MARKER tag that is either an ad begin tag or an ad end tag.
>
> An ad begin tag indicates the presence of an ad interval that begins at the ad begin tag's time. The end time of the interval is determined as follows:
>
> - If next ad marker tag in the file is an ad end tag, the interval ends at the time of the ad end tag. In this case, the DURATION of the ad begin tag and ad end tag have no meaning and the time of the ad end tag is the sole determinant of the ad interval endpoint.
> - If the ad begin tag's DURATION is non-zero and the next ad marker tag is not an ad end tag, the interval ends DURATION seconds after the begin time.
> - If the ad begin tag's DURATION is zero and there is no next ad marker tag, the interval ends at a time that is unknown, but greater than end time of the most recent segment.
>
> The next ad marker tag after an ad begin tag with zero DURATION SHALL NOT be an ad begin tag.
>
> An ad end tag that is the first ad marker tag to appear in a file indicates the presence of an ad interval that begins at some unknown time before the time of the earliest content and ends at the ad end tag's time.
>
> The first segment that follows an ad end tag SHALL be the last segment containing content for the associated ad interval. The first segment that follows an ad begin tag SHALL be the first segment containing content for the associated ad interval.

A Primetime media playlist advertises preroll intervals in a similar manner to the process for advertising ad intervals, with the following exceptions:

> Replace all instances of the word "ad" with "preroll".
>
> Replace all instances of the word "AdBegin" with "PrerollPodBegin".
>
> Replace all instances of the word "AdEnd" with "PrerollPodEnd".
>
> Since preroll intervals are required to start at the start of content, the first preroll marker tag SHALL either not exist or be a preroll begin tag that affects the first segment of the playlist and has an OFFSET of 0. Since preroll intervals are required to have a defined end time, the final preroll marker tag, if it exists, SHALL be a preroll end tag.

A Primetime media playlist advertises midroll intervals in a similar manner to the process for advertising ad intervals, with the following exceptions:

> Replace all instances of the word "ad" with "midroll".
>
> Replace all instances of the word "AdBegin" with "PodBegin".
>
> Replace all instances of the word "AdEnd" with "PodEnd".
>
> The meaning of a midroll end tag that is the first midroll marker tag to appear in a file is changed. A midroll end tag that is the first midroll marker tag to appear in a file indicates the presence of a midroll interval that ends at the midroll end tag's time. The interval begin time is determined by the presence of a preroll interval. If a preroll interval is present, the midroll interval begins at the end time of the preroll interval. Otherwise, the midroll interval begins at some unknown time before the time of the earliest content.

When multiple EXT-X-MARKER tags are associated with the same segment, those tags SHALL appear in time order. When the time of multiple tags is identical, those tags SHALL be ordered by the following TYPE order: "PrerollPodBegin", "PodBegin", "AdBegin", "AdEnd", "PodEnd", then "PrerollPodEnd".

### The following additional constraints apply to Primetime HLS version 1 playlists:

- All ad begin tags SHALL specify a non-zero DURATION.
- If a midroll begin tag specifies a non-zero DURATION, the next midroll marker tag SHALL be a midroll end tag or not exist.

See Section 9.1 "Example playlist with preroll interval", Section 9.2 "Example playlist with midroll interval", or Section 9.3 "Example playlist with complex intervals" for examples.

### 6.4. VMAP Fragments data format

If the TYPE of an EXT-X-MARKER tag is "PrerollPodBegin", "PrerollPodEnd", "PodBegin", or "PodEnd", its DATA value, if present, SHALL be the Base64 encoding of an xml document where the following elements are defined:

AdTrackingFragments element

This REQUIRED element serves as the parent element for AdTrackingFragment elements. The parent of this element is the document root. The document SHALL contain exactly one instance of this element.

AdTrackingFragment element

An AdTrackingFragments SHALL contain at least one AdTrackingFragment element. Note that although AdTrackingFragment elements beyond the first MAY be present, only the meaning of the first element is currently defined.

The first AdTrackingFragment element SHALL contain exactly one VMAP element that conforms to [VMAP1]. The VMAP element SHALL contain exactly one AdBreak element.

The AdBreak element provides information about the midroll or preroll interval. The timeOffset attribute of the AdBreak element does not currently have a meaning in the context of the stream. When VMAP information is provided with both the begin and end tags of an interval, that information SHOULD be identical for both tags. When they are not identical, a client SHALL use the VMAP information of the end tag.

### 6.5. VAST Fragments data format

If the TYPE of an EXT-X-MARKER tag is "AdBegin" or "AdEnd", its DATA value, if present, will be the Base64 encoding of an xml document where the following elements are defined:

AdTrackingFragments element

This REQUIRED element serves as the parent element for AdTrackingFragment elements. The parent of this element is the document root. The document SHALL contain exactly one instance of this element.

AdTrackingFragment element

The AdTrackingFragments element SHALL contain one or more AdTrackingFragment elements.

Each AdTrackingFragment element SHALL contain exactly one VAST element that conforms to [VAST3]. The VAST element SHALL contain exactly one Ad element.

For a given AdTrackingFragments element, exactly one descendent VAST element SHALL contain an Inline element, and all other descendent VAST elements SHALL contain a Wrapper element. The Inline element indicates the final VAST response, while Wrapper elements indicate intermediate responses.

The Ad elements provide information about the ad interval. If VAST information is provided with both the begin and end tags of an interval, the VAST information provided by both tags SHOULD be identical. When they are not identical, a client SHALL use the VAST information of the end tag.

## 7. Adobe® Access Digital Rights Management

This section describes how a Primetime playlist can utilize Adobe Access (AAXS) Digital Rights Management (DRM).

In this section, the terms "content encryption key", "DRM metadata", "license", and "policy" are equivalent to the definitions outlined in [AAXS-OVERVIEW]. The "AAXS" prefix will be present before these terms to disambiguate their usage.

An "AAXS segment" is an encrypted media segment that can be decrypted using the AAXS DRM system.

### 7.1. EXT-X-FAXS-CM tag

The EXT-X-FAXS-CM tag is used to provide AAXS DRM metadata for a Primetime playlist.

In a Primetime media playlist, the EXT-X-FAXS-CM tag can be used to designate a subset of segments as AAXS segments. See Section 7.2

"EXT-X-FAXS-CM tag in Primetime media playlists".

In a Primetime master playlist, the EXT-X-FAXS-CM tag can be used to specify the AAXS DRM metadata of the variant streams. See Section 7.3 "EXT-X-FAXS-CM tag in Primetime master playlists".

If a Primetime (media or master) playlist contains the EXT-X-FAXS-CM tag, the URL of the playlist SHALL contain the query parameter "faxs" with a value of 1.

The value of the EXT-X-FAXS-CM tag provides an AAXS DRM metadata. The value of the EXT-X-FAXS-CM tag SHALL supply either a URI for the AAXS DRM metadata or an embedded AAXS DRM metadata.

If the value does not begin with the string "URI=", the AAXS DRM metadata embedded in the value of the tag. In this case, the value of the tag is a string whose contents are the AAXS DRM metadata expressed as base64. The value SHALL NOT be enclosed in double quotes (").

If the value begins with the string "URI=", the AAXS DRM metadata can be acquired from a remote location. The value of the tag is an attribute list with the following attributes defined:

URI

This REQUIRED, quoted-string attribute is a URI at which the AAXS DRM metadata can be found. If the URI is relative, the URI is relative to the location of the containing playlist. For example, if the URI of the playlist is "http://a.com/b/c", a relative URI value of "e" is equivalent to an absolute URI of "http://a.com/b/e".

## 7.2. EXT-X-FAXS-CM tag in Primetime media playlists

This section describes the usage of the EXT-X-FAXS-CM tag in Primetime media playlists. Unless otherwise specified, the requirements presented in this section apply only to Primetime media playlists.

A Primetime media playlist MAY contain the EXT-X-FAXS-CM tag.

When present, the EXT-X-FAXS-CM tag SHALL appear before all EXT-X-KEY tags (in line order) whose METHOD is "AES-128".

An "AAXS EXT-X-KEY tag" is an EXT-X-KEY tag that has a METHOD value of "AES-128" and appears after an EXT-X-FAXS-CM tag (in line order). The meaning of an AAXS EXT-X-KEY tag differs from the meaning of an EXT-X-KEY tag in the following respects:

- The segments affected by an AAXS EXT-X-KEY tag are AAXS segments.
- The EXT-X-FAXS-CM tag that appears prior to an AAXS EXT-X-KEY tag (in line order) provides the AAXS DRM metadata of the AAXS EXT-X-KEY tag's segments.
- The URI attribute of an AAXS EXT-X-KEY tag is REQUIRED.
- If local key delivery is specified by the AAXS policy, the URI's scheme SHALL be "faxs://". If remote key delivery is specified by the AAXS policy, the URI SHALL be the URI of the remote key server. In this case, the URI's scheme SHALL be "https://".
- The URI MAY contain the "EncryptedRK" query parameter. The value of this parameter is the rotation key, the key used to encrypted the segment, encrypted with AES-128 ECB using the AAXS content encryption key. The value is in hexadecimal format, beginning with the string "0x". See Section 9.6 "Example playlist with key rotation" for an example.
- The meaning of any other URI contents for AAXS EXT-X-KEY tags is currently undefined.
- An AAXS EXT-X-KEY tag MAY have the CMSha1Hash attribute. The attribute value is a hexadecimal-integer that contains the message digest that results from applying SHA-1 (as defined by [FIPS.180-2.2002]) to the tag's AAXS DRM metadata. This attribute SHALL be present when the stream makes use of license rotation.

See Section 9.4 "Example media playlist with AAXS segments" for an example.

When an EXT-X-KEY tag does not have a preceding EXT-X-FAXS-CM tag or the method of an EXT-X-KEY tag is not "AES-128", the meaning of the EXT-X-KEY tag is unaltered from the meaning specified in [I-D.pantos-http-live-streaming].

A Primetime media playlist MAY contain multiple instances of the EXT-X-FAXS-CM tag. This allows the playlist to associate different AAXS DRM metadata or AAXS content encryption keys with different media segments of a media playlist. See Section 9.5 "Example playlist with multiple DRM metadatas" for an example.

## 7.3. EXT-X-FAXS-CM tag in Primetime master playlists

This section describes the usage of the EXT-X-FAXS-CM tag in Primetime master playlists. Unless otherwise specified, the requirements presented in this section apply only to Primetime master playlists.

A Primetime master playlist MAY contain the EXT-X-FAXS-CM tag. A Primetime master playlist SHALL contain the EXT-X-FAXS-CM tag if any of its variant streams contains the EXT-X-FAXS-CM tag.

The EXT-X-FAXS-CM tag SHOULD provide the AAXS DRM metadata of one variant stream. Since the variant streams typically use the same AAXS DRM metadata, the AAXS DRM metadata provided in the master playlist is typically the same as that of all streams. When multiple AAXS DRM metadatas exist for a variant stream, the EXT-X-FAXS-CM tag SHOULD provide the first AAXS DRM metadata of the stream. This improves start-up performance when beginning a stream.

All variant streams in an AAXS master playlist SHOULD use the same AAXS DRM metadata. This improves performance when switching between bitrates of those streams.

# 8. Signed playlists

This section describes how the authenticity of a Primetime media playlist can be verified through the use of signatures.

See Section 9.7 "Example signed Primetime media playlist" for an example.

## 8.1. Authenticity

A client can validate that a "signature enabled playlist" is "validly signed". The outcome of this validation provides the client with evidence regarding the authenticity of the playlist:

- When a signature enabled playlist is validly signed, this provides strong evidence in favor of the playlist's authenticity. A typical client will treat such a playlist as authentic.
- When a signature enabled playlist is not validly-signed, this provides no evidence in favor or against the playlist's authenticity. Although such a playlist is not necessarily inauthentic, a typical clients will treat an unverifiable, signature enabled playlist as inauthentic.

A client SHOULD NOT validate playlists that are not signature enabled. The validity of such a signature has no implications on the authenticity of the file.

This specification does define what actions should be taken by a client if it determines that a playlist is potentially inauthentic. Although any behavior is possible, a typical client might refuse playback of a playlist that is signature enabled and not validly signed.

A method for embedding signatures within master playlists is not currently defined. The meaning of signature related tags and AAXS policy properties within master playlists is currently undefined.

## 8.2. Enabling signatures

A "signature enabled playlist" is a Primetime media playlist where all of the following apply:

- The playlist contains at least one EXT-X-FAXS-CM tag.
- The playlist contains at least one "signature enabled policy".
- The playlist is not a master playlist.

A "signature enabled policy" is an AAXS policy that contains a property "ManifestSigning" whose value is "1".

A Primetime media playlist can contain AAXS policies embedded within its EXT-X-FAXS-CM tags. As described in [AAXS-OVERVIEW], AAXS policies are embedded within the AAXS DRM metadata. As described in [AAXS-OVERVIEW], AAXS DRM metadatas are embedded within EXT-X-FAXS-CM-tags.

If a Primetime media playlist contains at least one signature enabled policy, all policies contained within the playlist SHALL be signature enabled policies.

## 8.3. Playlist validity

A signature enabled playlist is "validly signed" if all the following conditions are true:

- The playlist contains exactly one EXT-X-FAXS-SIGNATURE tag whose VERSION is "1".
- The EXT-X-FAXS-SIGNATURE tag with VERSION "1" is a "valid signature tag", as defined by Section 8.7 "EXT-X-FAXS-SIGNATURE tag validity".
- The playlist contains exactly one EXT-X-FAXS-PACKAGINGCERT tag.
- The Subject Key Identifier (as specified in [RFC3280]). of the certificate embedded within the EXT-X-FAXS-PACKAGINGCERT tag equals the value of the CertID AAXS policy property.

## 8.4. EXT-X-FAXS-SIGNATURE tag

The EXT-X-FAXS-SIGNATURE tag contains a signature for the playlist.

The tag MAY be present in a Primetime media playlist. If the EXT-X-FAXS-SIGNATURE tag is present in a playlist, the EXT-X-FAXS-CM tag SHALL also be present in the playlist. A Primetime media playlist SHALL NOT contain more than one EXT-X-FAXS-SIGNATURE tag whose VERSION is "1". Additional instances of the tag MAY be present in a playlist if their VERSION is not "1".

If the playlist contains the EXT-X-ENDLIST tag, all instances of the EXT-X-FAXS-SIGNATURE tag SHALL appear prior to the EXT-X-ENDLIST tag.

The value of this tag is an attribute-list. The following attributes are defined:

VERSION

This REQUIRED, enumerated-string attribute indicates the format of the tag. The only currently defined value of this tag is "1".

To aid forward compatibility, the client SHOULD ignore any EXT-X-FAXS-SIGNATURE tags whose VERSION it does not understand.

SignatureValue

This attribute is defined and REQUIRED if the value of the VERSION attribute is "1". This quoted-string attribute indicates the signature data. The value is a base64 representation of the signature data. For compatibility with Primetime playlists written prior to the publishing of this specification, a Primetime client SHALL accept values that are not enclosed in double quotes ("). See Section 8.7 "EXT-X-FAXS-SIGNATURE tag validity" for additional constraints on this attribute for valid EXT-X-FAXS-SIGNATURE tags.

## 8.5. EXT-X-FAXS-PACKAGINGCERT tag

The EXT-X-FAXS-PACKAGINGCERT tag provides the end entity certificate of the signer.

The tag MAY be present in a Primetime media playlist. If the EXT-X-FAXS-PACKAGINGCERT tag is present in a playlist, the EXT-X-FAXS-CM tag SHALL also be present in the playlist. A Primetime media playlist SHALL contain at most one EXT-X-FAXS-PACKAGINGCERT tag.

The value of the tag is an quoted-string. The contents of the string is the X.509 end entity certificate, expressed as base64. The value SHALL NOT be enclosed in double quotes (").

See Section 8.3 "Playlist validity" for additional requirements on the EXT-X-FAXS-PACKAGINGCERT tag for validly signed playlists.

## 8.6. CertID AAXS policy property

An AAXS policy MAY contain a custom property "CertID".

The value of this property is the Subject Key Identifier (as specified in [RFC3280]). of the signing certificate. All policies contained within a signature enabled playlist SHALL specify the same value for "CertID".

See Section 8.3 "Playlist validity" for additional requirements on the CertID AAXS policy property for validly signed playlists.

## 8.7. EXT-X-FAXS-SIGNATURE tag validity

An EXT-X-FAXS-SIGNATURE tag whose VERSION attribute is "1" is valid if its signature data is the RSA1024-SHA256 signature of the "canonicalized form" of the input playlist; That is, the tag is valid if the output of the RSASSA-PKCS1-V1_5-VERIFY is "valid signature" when the following inputs are supplied:

- The bit-length of the RSA public key, n, is 1024
- The hash function is SHA-256 (as defined by [FIPS.180-2.2002]).
- The public key is provided by the EXT-X-FAXS-PACKAGINGCERT tag.
- The message whose signature is to be verified is the canonicalized form of the playlist.
- The signature to be verified is the signature data, as expressed by SignatureValue.

The "canonicalized form" of a Playlist file is a string that is identical to the return value of the pseudocode below.

The "splitting" operation on line 3 is defined as an operation that partitions an input string based on a delimiter (in this case, LF) and returns an array of the partitions in the order in which they appear in the original string, with the delimiter removed. The single argument form of the String.split(...) function of Java or JavaScript languages is conformant with the "splitting" operation defined in this specification.

The "trim" operation on line 7 is defined as an operation that removes all whitespace characters from the start and end of a string. A whitespace character (as referenced by lines 5-6) is defined as any character in the range U+0000 through U+0020, inclusive.

```
let INPUT = the contents of the Playlist file expressed a single UTF-8 string
let OUTPUT = the empty string
let INPUT_LINES = the array of strings that results from splitting INPUT on the LF character (U+000A)
for each INPUT_LINE in INPUT_LINES:
    if the INPUT_LINE is not a "#EXT-X-FAXS-SIGNATURE" tag:
        let OUTPUT_LINE = a copy of INPUT_LINE
        trim OUTPUT_LINE
        append the LF character to OUTPUT_LINE
        append OUTPUT_LINE to OUTPUT
remove all LF characters from the end of OUTPUT
return OUTPUT
```

If the Client encounters a EXT-X-FAXS-SIGNATURE tag with a VERSION attribute it doesn't understand, the Client SHALL ignore the tag during the validation process.

# 9. Examples

## 9.1. Example playlist with preroll interval

This playlist contains a single preroll interval that spans s1.ts, s2.ts, s3.ts, and s4.ts. The preroll interval contains 2 ad intervals. The first ad interval spans s1.ts and s2.ts. The second ad interval spans s3.ts and s4.ts. The remaining segments (s5.ts, s6.ts) contain primary content.

The end tags (lines 8, 15, and 16) can be omitted and still convey the same logical intervals since their corresponding begin tags (lines 4, 5, and 12) provide non-zero DURATIONs. However, if the PrerollPodBegin (line 4) marker instead provided a zero DURATION, the PrerollPodEnd (line 16) would have to be present to convey the same logical intervals.

```
00  #EXTM3U
01  #EXT-X-ADOBE:1
02  #EXT-X-TARGETDURATION:10
03  #EXT-X-MEDIA-SEQUENCE:1
04  #EXT-X-MARKER:ID="m1",TYPE=PrerollPodBegin,DURATION=40,
      DATA="AAA..."
05  #EXT-X-MARKER:ID="m2",TYPE=AdBegin,DURATION=20,DATA="BBB..."
06  #EXTINF:10,
07  s1.ts
08  #EXT-X-MARKER:ID="m3",TYPE=AdEnd,OFFSET=10,DURATION=0,
      DATA="BBB..."
09  #EXTINF:10,
10  s2.ts
11  #EXT-X-DISCONTINUITY
12  #EXT-X-MARKER:ID="m4",TYPE=AdBegin,DURATION=20,DATA="CCC..."
13  #EXTINF:10,
14  s3.ts
15  #EXT-X-MARKER:ID="m5",TYPE=AdEnd,OFFSET=10,DURATION=0,
      DATA="CCC..."
16  #EXT-X-MARKER:ID="m6",TYPE=PrerollPodEnd,OFFSET=10,DURATION=0,
      DATA="DDD..."
17  #EXTINF:10,
18  s4.ts
19  #EXT-X-DISCONTINUITY
20  #EXTINF:10,
21  s5.ts
22  #EXTINF:10,
23  s6.ts
24  #EXT-X-ENDLIST
```

## 9.2. Example playlist with midroll interval

This playlist contains a single midroll interval that spans s3.ts, s4.ts, s5.ts, and s6.ts. The midroll interval contains 2 ad intervals. The first ad interval spans s3.ts and s4.ts. The second ad interval spans s5.ts and s6.ts. The remaining segments (s1.ts, s2.ts, s7.ts, and s8.ts) contain primary content.

As in earlier examples, the AdEnd and PodEnd markers of this example can be omitted since the corresponding begin tags provide non-zero DURATIONs.

```
#EXTM3U
#EXT-X-ADOBE:1
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:1
#EXTINF:10,
s1.ts
#EXTINF:10,
s2.ts
#EXT-X-DISCONTINUITY
#EXT-X-MARKER:ID="m1",TYPE=PodBegin,DURATION=40,DATA="AAA..."
#EXT-X-MARKER:ID="m2",TYPE=AdBegin,DURATION=20,DATA="BBB..."
#EXTINF:10,
s3.ts
#EXT-X-MARKER:ID="m3",TYPE=AdEnd,OFFSET=10,DURATION=0,
  DATA="BBB..."
#EXTINF:10,
s4.ts
#EXT-X-DISCONTINUITY
#EXT-X-MARKER:ID="m4",TYPE=AdBegin,DURATION=20,DATA="CCC..."
#EXTINF:10,
s5.ts
#EXT-X-MARKER:ID="m5",TYPE=AdEnd,OFFSET=10,DURATION=0,
  DATA="CCC..."
#EXT-X-MARKER:ID="m6",TYPE=PodEnd,OFFSET=10,DURATION=0,
  DATA="AAA..."
#EXTINF:10,
s6.ts
#EXT-X-DISCONTINUITY
#EXTINF:10,
s7.ts
#EXTINF:10,
s8.ts
#EXT-X-ENDLIST
```

## 9.3. Example playlist with complex intervals

This playlist contains a preroll interval that spans s1.ts. The preroll interval contains a single ad interval that spans s1.ts. The playlist also contains 3 midroll intervals. The first midroll interval spans s2.ts and s3.ts and contains a single ad interval that spans s2.ts and s3.ts. The second midroll interval spans s5.ts and contains 2 ad intervals. The first ad interval spans the first 3 seconds of s5.ts. The second ad interval spans the remaining 7 seconds of s5.ts. The third midroll interval spans s7.ts and contains a single ad interval that spans s7.ts. The remaining segments (s4.ts) contain primary content.

```
00  #EXTM3U
01  #EXT-X-ADOBE:1
02  #EXT-X-TARGETDURATION:10
03  #EXT-X-MEDIA-SEQUENCE:1
04  #EXT-X-MARKER:ID="m1",TYPE=PrerollPodBegin,DURATION=10,
      DATA="AAA..."
05  #EXT-X-MARKER:ID="m2",TYPE=AdBegin,DURATION=10,DATA="BBB..."
06  #EXTINF:10,
07  s1.ts
08  #EXT-X-DISCONTINUITY
09  #EXTINF:10,
10  s2.ts
11  #EXT-X-DISCONTINUITY
12  #EXT-X-MARKER:ID="m3",TYPE=AdEnd,OFFSET=10,DURATION=0,
      DATA="CCC..."
13  #EXT-X-MARKER:ID="m4",TYPE=PodEnd,OFFSET=10,DURATION=0,
      DATA="DDD..."
14  #EXTINF:10,
15  s3.ts
16  #EXT-X-DISCONTINUITY
17  #EXTINF:10,
18  s4.ts
19  #EXT-X-DISCONTINUITY
20  #EXT-X-MARKER:ID="m5",TYPE=PodBegin,DURATION=10,DATA="EEE..."
21  #EXT-X-MARKER:ID="m6",TYPE=AdBegin,DURATION=3,DATA="FFF..."
22  #EXT-X-MARKER:ID="m7",TYPE=AdBegin,DURATION=7,DATA="GGG..."
23  #EXTINF:10,
24  s5.ts
25  #EXT-X-DISCONTINUITY
26  #EXTINF:10,
27  s6.ts
28  #EXT-X-DISCONTINUITY
29  #EXT-X-MARKER:ID="m8",TYPE=PodBegin,DURATION=0,DATA="HHH..."
30  #EXT-X-MARKER:ID="m9",TYPE=AdBegin,DURATION=10,DATA="III..."
31  #EXTINF:10,
32  s7.ts
```

## 9.4. Example media playlist with AAXS segments

This playlist contains 2 AAXS segments that use the same DRM metadata (line 5) and local key delivery (line 6).

```
01  #EXTM3U
02  #EXT-X-ADOBE:1
03  #EXT-X-TARGETDURATION:10
04  #EXT-X-MEDIA-SEQUENCE:1
05  #EXT-X-FAXS-CM:AAAA...
06  #EXT-X-KEY:METHOD=AES-128,URI="faxs://example.com"
07  #EXTINF:10,
08  s1.ts
09  #EXTINF:10,
10  s2.ts
11  #EXT-X-ENDLIST
```

## 9.5. Example playlist with multiple DRM metadatas

This playlist contains 2 AAXS segments that use the DRM metadata specified by line 5, and 2 AAXS segments that use the DRM metadata specified by line 11. All segments use remote key delivery.

```
#EXTM3U
#EXT-X-ADOBE:1
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-FAXS-CM:AAA...
#EXT-X-KEY:METHOD=AES-128,URI="https://keyserver.com"
#EXTINF:10,
s1.ts
#EXTINF:10,
s2.ts
#EXT-X-FAXS-CM:BBB...
#EXT-X-KEY:METHOD=AES-128,URI="https://keyserver.com"
#EXTINF:10,
s4.ts
#EXTINF:10,
s5.ts
#EXT-X-ENDLIST
```

## 9.6. Example playlist with key rotation

This playlist contains 4 AAXS segments that use the same DRM metadata (line 5). The first 2 AAXS segments use a different rotation key from the next 2 AAXS segments.

```
01  #EXTM3U
02  #EXT-X-ADOBE:1
03  #EXT-X-TARGETDURATION:10
04  #EXT-X-MEDIA-SEQUENCE:1
05  #EXT-X-FAXS-CM:AAA...
06  #EXT-X-KEY:METHOD=AES-128,URI="https://example.com"
07  #EXTINF:10,
08  s1.ts
09  #EXTINF:10,
10  s2.ts
11  #EXT-X-KEY:METHOD=AES-128,
      URI="https://keyserver.com&EncryptedRK=0x0123456789A"
12  #EXTINF:10,
13  s3.ts
14  #EXTINF:10,
15  s4.ts
16  #EXT-X-ENDLIST
```

## 9.7. Example signed Primetime media playlist

```
#EXTM3U
#EXT-X-ADOBE:1
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-FAXS-CM:AAA....
#EXT-X-FAXS-PACKAGINGCERT:BBB...
#EXT-X-KEY:METHOD=AES-128,URI="faxs://keyserver.com"
#EXTINF:10,
s1.ts
#EXTINF:10,
s2.ts
#EXT-X-FAXS-SIGNATURE:VERSION=1,SignatureValue="CCC..."
#EXT-X-ENDLIST
```

## Normative References

[DPI]

Adobe Systems Incorporated, "Primetime Digital Program Insertion Signaling Specification v1.1", December 2013.

[FIPS.180-2.2002]

National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-2, August 2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

[I-D.pantos-http-live-streaming]

Pantos, R., "HTTP Live Streaming", draft-pantos-http-live-streaming-12 (work in progress), October 2013.

[PKCS.12.1997]

"Personal Information Exchange Standard", Version 1.0 Draft", PKCS 12, April 1997.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.

[RFC3280]

Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[VAST3]

IAB, "Video Ad Serving Template (VAST): Version 3.0", July 2012.

[VMAP1]

IAB, "Video Multiple Ad Playlist (VMAP): Version 1.0", July 2012.

## Informative References

[AAXS-OVERVIEW]

Adobe Systems Incorporated, "Adobe Access Version 4.0: Overview", 2013, http://www.adobe.com/support/adobeaccess/pdfs/server/AdobeAccess_4_Overview.pdf.