**2013**

# Video Multiple Ad Playlist (VMAP 1.0)

**June 2013**

# Acknowledgements

**These guidelines were produced through IAB Australia's Standards and Guidelines Council on which these company sit.**

# Table of Contents

## Executive Summary

The IAB Video Multiple Ad Playlist (VMAP) specification is an XML template that video content owners can use to describe the structure for ad inventory insertion when they don't control the video player or the content distribution outlet.

In order to effectively monetize video content with in-stream insertion advertising, video content owners must carefully manage the structure and use of ad inventory opportunities available within their content. When the content owner controls the content distribution outlet, the content owner can easily manage ad placement within the content they play. However, when the video content airs in a video player that they do not control (such as when the content is syndicated using a video aggregation service), managing ad placement becomes complicated.

Business arrangements between the video content owner and a distribution outlet may grant the owner the right to manage its video ad inventory, but without control of the video player or the distribution outlet, the video content owner is effectively prevented from exercising this right.

Content owners and distribution outlets have looked to the IAB Video Ad Serving Template (VAST) to help solve this problem. While VAST 3.0 provides some additional controls over the use of video ad inventory (such as support for ad pods), it lacks the ability to define ad breaks or the timing of those ad breaks within the video content entertainment timeline.

With VMAP, video content owners can exercise control over the ad inventory displayed in their content when they can't control the video player, to capitalize on advertising while maintaining the integrity of their program content. VMAP enables the content owner to define the ad breaks within their content, including the timing for each break, how many breaks are available, what type of ads and how many are allowed in each break.

What VMAP cannot do is define the ads themselves. VMAP was designed to accept VAST 3.0 ad responses to fill ad breaks, but can also accept ad responses in other formats. VMAP is complementary to VAST and only useful in cases where content owners have no control over the video players, but have rights to control the advertising experiences within their content.

Video players that support VMAP provide a mechanism to honor business agreements that give advertising control to video content owners. VMAP specifications were defined with creativity and innovation in mind and should not limit video player design. As with all IAB guidelines and specifications, this document will be updated as video advertising progresses and new ad formats become more widely adopted.

## Intended Audience

VMAP is defined for a specialized use case and is intended for anyone that is part of a business agreement where one party is afforded the control of ad breaks within video program content they either own or distribute. Specifically, this document is designed for the engineers who design the video players in which VMAP is accepted as well as for those who develop the systems that serve VMAP. While VMAP is defined for this special use case, other uses may be found and anyone who is party to other use cases will benefit from being familiar with VMAP specifications.

## IAB Guidelines

The incredible growth of digital video has been accompanied by a steep rise in video advertising spend. To facilitate this spend, the IAB Digital Video Committee has brought together publishers, agencies and vendors to create a set of video advertising specifications that establish a common framework for communication between ad servers and video players. Six sets of IAB guidelines have been developed to help improve video advertising:

- **Video Ad Measurement Guidelines (VAMG)**: Outlines the set of events that should be tracked when a video ad is played back.
- **Video Ad Serving Template (VAST):** Enables the common structure of a video ad response sent from an ad server to a video player.
- **Video Player Ad Interface Definition (VPAID)**: Establishes the communication protocol between an interactive ad and the video player that is rendering it.
- **Video Multiple Ad Playlist (VMAP)**: Enable a structure for a playlist of video ads sent from an ad server to a video player.
- **Digital Video Ad Format Guidelines and Best Practices**: Outlines the general format and best practices that video ads should adhere to for the best advertising experience.
- **Digital Video In-Stream Ad Metrics Definitions**: Defines industry-accepted metrics for measuring video ad effectiveness.

The following diagram explains the relationship between these guidelines and the video ad serving process.



# 1  General Overview

The process of placing ads into video content involves two basic steps:

1) Defining the structure of the ad inventory such as the opportunities within the video content for serving ads (or ad breaks), their quantity, their position in the content timeline, the number of ads allowed, and so on.
2) Defining the ads to fill the ad inventory.

VMAP defines the structure for the ad inventory (1), and VAST defines the ads that fill the inventory (2).

## 1.1  Relationship between VMAP and VAST

VMAP is not a replacement for VAST. VMAP is complementary to VAST and VAST can be served with or without the use of VMAP.

- VMAP is used to express the structure the ad inventory as a set of timed ad breaks within a publisher's video content.
- VAST (either contained within the VMAP response or by referencing another source) is used to express the ad or ads that are to fill a particular ad break.

Using VAST with VMAP or VAST alone depends on the integration scenario. The parties involved in controlling the video player, the content or content distribution, and the ad server should negotiate details based on the business and policy arrangements between them.

In general, VMAP is recommended when a content owner or distributor cannot control the video player but is granted the right to control the structure or allocation of ad inventory for the video content to be played in the video player. Without this control, VAST alone can support ad servers' needs.

VMAP is recommended when the following conditions are true:

- The ad server generally has first right to all ad inventory within the video program. Often this party is the producer of, or owns rights to the video content.
- The ad server has the right to control the structure of the video ad inventory within the video content, including the number and placements of ad breaks and the use of each ad break (i.e. number of ads allowed within the ad break, the allocation of video ad inventory to advertisers or other parties for resale, etc.).
- The ad server does not have the right to control the inventory within the video content, but does have the right to prescribe how the video ad inventory is used (again, number of ads allowed in each ad break, allocation of ad breaks to advertisers or other parties, etc.)

Other conditions for using VMAP are possible, including use by the video player owner to build playlists instead of using proprietary code.

## 1.2  Terminology

The following terminology is used throughout this document.

**Ad Break**: A location or point in time where one or more ads may be scheduled for delivery. Specifically, an "Ad Break" is often used in this document to represent the technical format and expectations that define an Ad Break in VMAP 1.0.

**Ad Pod**: A sequence of linear ads played back-to-back, like a commercial break with multiple ad spots on TV. Specifically, "Ad Pod" is often used in this document to represent the compliance format for Ad Pods in VAST 3.0 and includes all the technical format and expectations that define an Ad Pod in VAST 3.0.

**Ad Response**: A formatted ad definition that provides details for ad display and is returned when an ad (or ads) is requested from another system, usually the publishing organization that supplies digital space for ad display. An ad response formatted using VAST is called a VAST response.

**Ad Response Template**: The formal guidelines that define how to structure the ad creative details provided in an ad response, setting up expectations that enable the interactive advertising community

to design technology that can receive ads served from a variety of systems that may otherwise be incompatible.

**Content Video:** The entertaining content video that a publisher provides for its audience. Content video is often monetized by providing ads at timed points along the content video timeline.

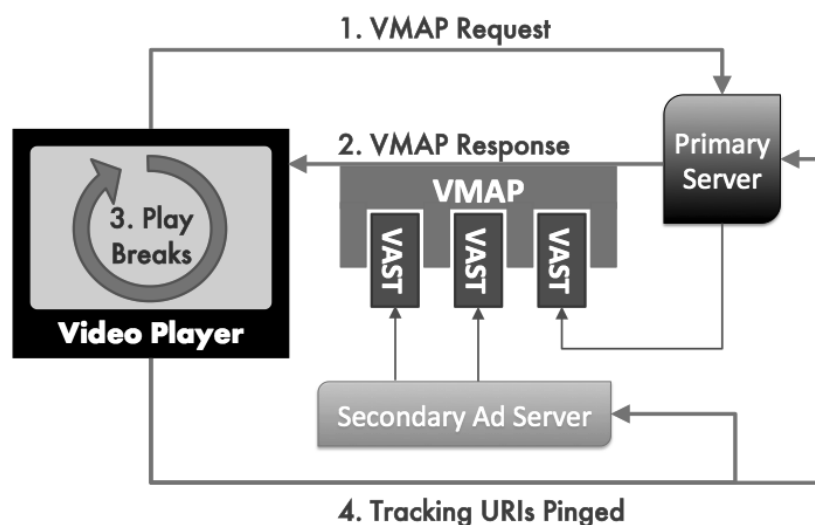**Playlist:** A structured list of ad breaks within video content.

**Primary Ad Server**: The first ad server to receive a request for ads from either a browser or video player.

**Secondary Ad Server**: An ad server to which a primary ad server redirects a video player or browser to retrieve ads when the primary ad server does not directly manage the ad or ads to be served.

## 1.3  How VMAP Works

VMAP represents a playlist structure that wraps one or more ad responses. This structure defines the ad breaks within a video program, identifying details such as how many ad breaks, which ad types to display, and when each ad break should occur. VMAP cannot provide ads directly. Instead, a VMAP response must contain separate ad responses that define which ads to display. While any ad response format can be used, VMAP was designed to accept VAST ad responses.

The diagram below illustrates a simplified example of the VMAP serving process.



1. **VMAP Request:** Video player requests a VMAP response from the primary server.
2. **VMAP Response:** The primary server returns a VMAP response that contains a playlist of ad breaks. Each ad break references a VAST ad response that provides the video player with ads to fill specific ad breaks.
3. **Play Breaks:** The video player executes (displays) VAST ads at the time within the video program specified by VMAP and in accordance with VAST guidelines.
4. **Tracking URIs Pinged:** At the appropriate times, the video player sends requests the tracking URIs provided in VMAP. These tracking URIs represent VMAP-level events such as the start and end of ad break.

## 1.4  What VMAP is Not

To avoid confusion it is important to understand what VMAP is not.

1. **Not a replacement for VAST:** VMAP is not a replacement for or redundant with VAST. VMAP is a complementary guideline that helps solve a use case that VAST was not intended to address. VMAP provides a way to unambiguously represent a playlist of ad breaks but VAST provides the ads to be played in those ad breaks.

2. **Not required:** VMAP support is not required for VAST compliance.

3. **Not for requesting ads:** The VMAP guideline does not recommend an ad request format. As with VAST, each ad server is free to implement its own interface to represent an ad request.

4. **Not for everyone:** VMAP is not intended for use by all parties in the video advertising ecosystem. VMAP is intended for use by an ad server controlled by a party with the primary right to manage the video ad inventory (typically the party creating the video entertainment) in some video entertainment content and by a video player that distributes that content. For example, ad networks and rich media vendors don't usually produce video entertainment content (other than ads) or serve as content distribution outlets so they may never need to support VMAP. However, such parties are not prevented from implementing VMAP.

5. **Not an execution technology:** VMAP is not the video player execution of the VMAP response. As with VAST, VMAP does not "do" anything. It is simply a method of conveying information from an ad server to a video player. The VMAP guideline provides guidance as to how a video player should act in response to a VMAP document, but the implementation of the correct behavior is the responsibility of the party developing the player.
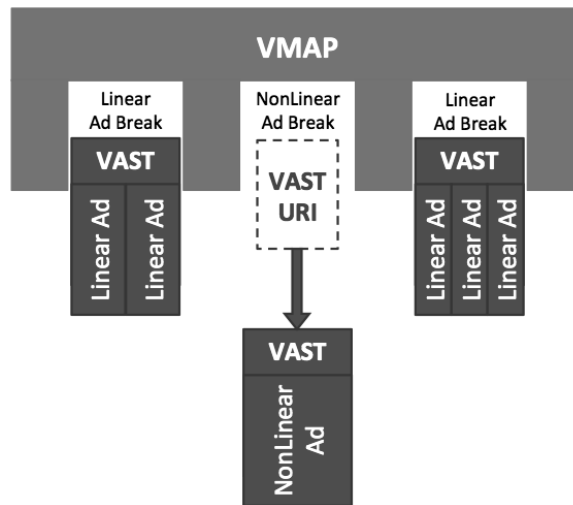
## 1.5  Live or Simulcast Content

While VMAP can conceivably be used to insert advertising into live or simulcast streaming content, the VMAP guideline does not include any special provisions for use in this environment or make any recommendations as to how to apply VMAP to live streams.

# 2  VMAP Implementation Details

## 2.1  General VMAP Document Structure

VMAP's purpose is to describe the structure of video ad inventory within a video program. VMAP does this by declaring one or more ad breaks, each with a specified time offset and type (such as linear or non-linear). These ad breaks comprise a playlist. The ads to be displayed within each ad break must be served using VAST or some other ad serving template.

The diagram below illustrates the conceptual structure of a VMAP response.



In this example, VMAP specifies a linear ad break followed by a nonlinear ad break and then another linear ad break. The linear ads are served using an InLine VAST repsonse, but the nonlinear ad is served initially as a URI that references a NonLinear VAST response from another server. URIs can be used to reference both linear and nonlinear VAST ads to fill appropriate ad breaks in a VMAP response.

## 2.2  Using VMAP with VAST and Other Ad Server Templates

VMAP is designed to use VAST 3.0 ad responses to supply ads for specified ad breaks but other ad serving templates can be used. Video players that are compliant with VMAP guidelines are only required to support VAST ad responses. Other ad serving template responses can be used, but the video player may choose to ignore non-VAST responses.

The VAST ads in a VMAP response are not wrapped in CDATA blocks. Since URIs and other potentially dangerous resources are already wrapped in VAST, eliminating CDATA blocks in VMAP simplifies the response by: making the response easier for the video player to parse, maintaining a simplified human-readable format, and avoiding all the embedded CDATA delimiters that would be needed otherwise. VAST is given this special treatment because VMAP was designed to accept VAST ads, but any non-VAST ads used in a VMAP response must be wrapped in a CDATA block.

Because VMAP wraps VAST XML-formatted ad responses, the dual XML document structure requires the use of an XML namespace in the VMAP response. For brevity, this guideline excludes the namespace in its examples, but VMAP should use the namespace URI: http://www.iab.net/videosuite/vmap with a prefix of "vmap" in front of each element name.

The following sample VMAP response uses a namespace:

```
<vmap:VMAP xmlns:vmap="http://www.iab.net/videosuite/vmap"
version="1.0">
   <vmap:AdBreak>
   …
   </vmap:AdBreak>
</vmap:VMAP>
```
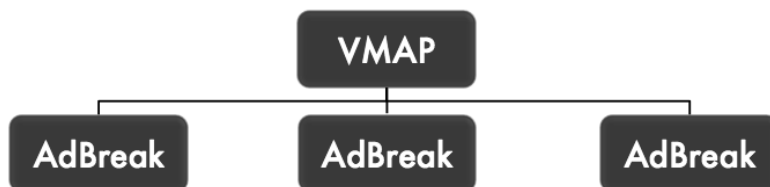
## 2.3  VMAP Document Structure Details

Each VMAP document is rooted at a `<VMAP>` element. The `<VMAP>` element represents the entire playlist of ad breaks to be scheduled during the video program. VMAP is declared with the `<VMAP>` element and includes an attribute for identifying the version, which is currently 1.0.  The example below declares the VMAP response:

```
<VMAP version="1.0">
   <!--VMAP details-->
</VMAP>
```

### 2.3.1  The <AdBreak> Element

The `<VMAP>` element contains zero or more `<AdBreak>` child elements as illustrated in the following diagram:



Each `<AdBreak>` element represents a single ad break, but may allow for multiple ads. Each `<AdBreak>` includes the following attributes:

- **timeOffset:** required to represent the timing for the ad break. Values for this attribute can be represented in one of four ways:
    - **time:** in the format `hh:mm:ss` or `hh:mm:ss.mmm` where `.mmm` is milliseconds and is optional. The time value is offset from the start of the video content to the placement of the ad break in the video content timeline.
    - **percentage:** if the duration of the video content is unknown, a percentage (in the format `n%` where "n" is a value from 0-100) can be entered and represents a percentage of the total video content duration from the start up to the point where the ad break should be entered.
    - **start/end:** for ad breaks that are inserted at the very start or end of the video content, the values "`start`" or "`end`" can be entered.
    - **position:** In cases where the timing of ad breaks is unknown (such as with live content), positional values can be entered in the format `#m` where "m" is an integer of 1 or greater and represents the ad break opportunity. For example, an ad break to be inserted at the first opportunity for an ad break would enter the value #1. Position values can only be honored if no other offset values are provided.

An ad break may contain an identical time offset as another ad break and is common when a linear ad is followed by a nonlinear ad. Also, a VMAP response can contain a mix of offset value types; however, when a mix is values is provided, any position values can be ignored.

- **breakType:** required to identify whether the ad break allows "`linear`", "`nonlinear`" or "`display`" ads. Display break types map to VAST companion ads. If more than one type is allowed, they can be entered using a comma between each (no spaces). For example "`linear,nonlinear`" can be entered. This attribute ensures that only intended ad types are accepted, that the video player displays ad breaks appropriate for viewer controls and that the video player can optimize video content playback dependent on the ad types being displayed (such as pausing content at the start of a linear ad to ensure precise timing). Please see section 2.4 for more information on handling ad breaks.
- **breakId:** An optional string identifier for the ad break.
- **repeatAfter:** An option used to distribute ad breaks equally spaced apart from one another along a linear timeline. If used, the value is time in the format `hh:mm:ss` or `HH:MM:SS.mmm` and indicates that the video player should repeat the same `<AdBreak>` break (using the same `<AdSource>`) at time offsets equal to the duration value of this attribute. Should a conflict occur where the duration of an ad break overlaps with a repeating ad break, the ad break scheduled to play first should take precedence while the overlapping ad break is ignored. Since an `<AdSource>` can be a VAST Wrapper to an ad server or ad network, the ads played in a repeated ad break may not be the same at each point.

For details on handling each of the break types see section 2.4.

Each `<adBreak>` element may contain any of the following elements:

- **<AdSource>:** Identifies the ads to be displayed in an ad break, either with an ad response inline or by referencing an ad response in another system. Please see section 2.3.2 for details.
- **<TrackingEvents>:** Provides the tracking URIs for events specific to VMAP. Please see section 2.3.3 for VMAP tracking details.
- **<Extensions>:** Can be used to express additional information not supported in the VMAP specification. Please see section 2.7 for more information.

### 2.3.2 The <AdSource> Element

Each `<AdBreak>` may contain up to one `<AdSource>` element, but is not required. The `<AdSource>` element provides the player with either an inline ad response or reference to an ad response.

The `<AdSource>` element includes the following attributes:

- **id**: an identifier for the ad source.
- **allowMultipleAds**: an optional Boolean value that indicates whether a VAST ad pod or multiple buffet of ads can be served into an ad break. If not specified, the video player accepts playing multiple ads in an ad break. The video player may choose to ignore non-VAST ad pods.
- **followRedirects**: an optional Boolean value that indicates whether the video player should honor redirects within an ad response. If not specified, the video player may choose whether it will honor redirects.

If the `<adSource>` element is provided, it must define the source for ads with one of the following elements:

- **<VASTAdData>:** indicates that a VAST ad response is embedded within the VMAP response. This element is reserved for VAST data so that it can be embedded without a CDATA block, but a namespace for VAST data should be declared.
- **<AdTagURI>:** a URI that references an ad response from another system. The URI must be contained within a CDATA block.
- **<CustomAdData>:** Arbitrary string data that represents non-VAST ad response. The data must be contained within a CDATA block.

To help the video player identify the ad response template used to serve the ad source, the attribute `templateType` is available for the `<AdTagURI>` and `<CustomAdData>` elements. (The `<VASTAdData>` is reserved for VAST ad responses and so template identification is not needed.)

The values accepted for `templateType` are: "vast", "vast1", "vast2", "vast3", or any string identifying a proprietary template. The numbers in the "vast" values represent the VAST version number (i.e. "vast3" represents VAST 3.0.)

### 2.3.3 VMAP Tracking

The `<TrackingEvents>` element is used to track the start and end of an ad break and whether an error occurred during the ad break. An `<AdBreak>` may contain one `<TrackingEvents>` element. The `<TrackingEvents>` element contains `<Tracking>` elements, each of which provides a tracking URI that are used to provide tracking URIs for the three tracking events available in VMAP.

VMAP tracking events:

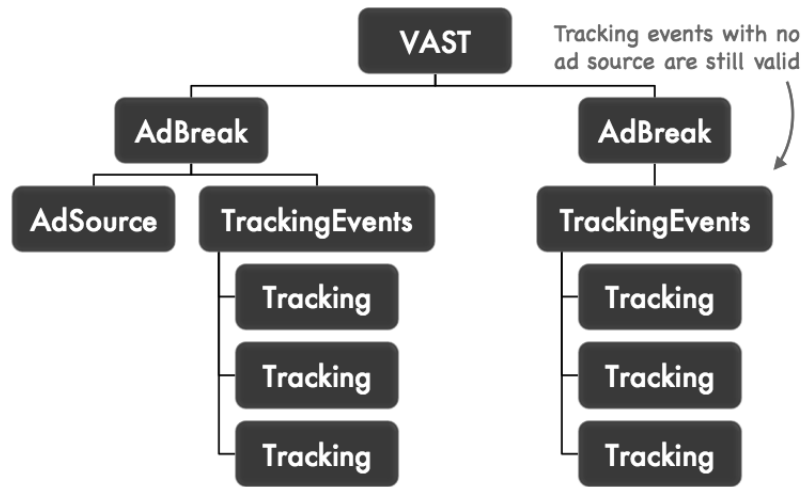- breakStart
- breakEnd
- error

Tracking events are identified using an event attribute for the `<Tracking>` element. The following example demonstrates a tracking event for "breakStart".

```
<TrackingEvents>
   <Tracking event="breakStart">
      <![CDATA[
           http://adserver.com/breakstart.gif
      ]>
   </Tracking>
</TrackingEvents>
```

Any URI or potentially harmful code should be wrapped in a CDATA block as shown in the example above.

When the associated event occurs during playback, the video player must send a request to the tracking URI provided within the `<Tracking>` elements. When multiple `<Tracking>` elements for the same event are included, the video player must send requests to all event tracking URIs when the associated event occurs. When an ad break contains tracking events but no ad source, the tracking events are still valid and tracking resources must be requested when associated events occur.

The following diagram illustrates the hierarchy of a VMAP response with tracking events.



The `<TrackingEvents>` and `<Tracking>` elements are intentionally named the same as tracking elements in VAST because their purpose and structure is the same. However, VMAP and VAST tracking elements track different events and belong to their respective specifications, so they should be distinguished using an XML namespace in VMAP.

## 2.4  Handling VMAP Ad Breaks

A video player should expect ad types identified by the `breakType` attribute. For example, if "`linear`" is indicated, then the video player should expect a linear ad and can ignore any other ad types offered.

A single ad break may specify multiple values for its break type, indicating that any of the types listed can be accepted with preference given to the first `breakType` listed. For example, a break type of "`linear,nonlinear`" indicates that a linear ad is preferred, but an overlay (nonlinear) is acceptable if the secondary ad server has no linear ad. The video player should attempt to interpret the VAST document in the context of a linear ad break, and if there is no ad, then attempt to interpret the document in the context of a nonlinear ad break.

The video player should attempt to execute the ad break as close to the specified time as possible, within the video player's capabilities. Nonlinear ad breaks can overlap if a second nonlinear ad break occurs before a nonlinear ad has completed playback. In this case the video player may disregard the later nonlinear ad break.

`Display` ad breaks map to companion ads that display outside of the video player and most likely are provided to accompany linear or nonlinear ads. These ad breaks should only be executed for display ad spaces that are currently visible. For example, if the video player is in full-screen mode, the display ad space isn't visible and the ad breaks may be ignored. A player may execute the most recent display ad break should the display ad space become visible (e.g., the player exits full-screen mode).

Some vendor may allow an ad break to be skipped. Whether or not an ad break can be skipped and how it should be handled should be negotiated between publisher and vendor.

## 2.5 Error Handling

The error event should be triggered whenever an error occurs. Errors fall into two categories:

- VMAP execution errors: occur when VMAP is processed, such as when extracting the response payloads, calling the secondary ad server, or parsing VMAP data.
- Ad response execution errors: occur when individual ad responses are processed.

Whenever an error of either type occurs any "error" tracking events associated with the `<AdBreak>` being executed should be called. A video player must replace `[ERROR_CODE]` and `[ERROR_MESSAGE]` macros, if provided, with appropriate values. Error messages are at the video player's discretion and special characters must be appropriately percent-encoded.

For ad response execution errors in VAST ads, the video player should report the same `[ERROR_CODE]` in VMAP that was reported for the VAST ad. For non-VAST ad execution errors, the video player may report at its discretion or use the error code 900 (Undefined error).

When an ad response execution occurs, the video player should execute as much of the as response as possible. For example, if there is an error with one ad in a sequenced ad pod preventing its playback, the player should attempt to move on to the next ad rather than aborting playback of the entire pod.

For VMAP execution errors, the video player should choose an error code from the table below.

| Code | Description |
|------|-------------|
| 900 | Undefined error |
| 1000 | VMAP schema error |
| 1001 | VMAP version of response not supported |
| 1002 | VMAP parsing error |
| 1003 | AdBreak type not supported |
| 1004 | General ad response document error |
| 1005 | Ad response template type not supported |
| 1006 | Ad response document extraction or parsing error |
| 1007 | Ad response document retrieval timeout |
| 1008 | Ad response document retrieval error (e.g., HTTP server responded with error code) |

## 2.6 No Ad Break

A VMAP response that is absent of any ad breaks, the response should contain only the root VMAP element, as shown in the following example.

```
<vmap:VMAP xmlns:vmap="http://www.iab.net/videosuite/vmap" version="1.0"/>
    …
</vmap:VMAP>
```

## 2.7 Extensions

VMAP supports the use of extensions, which may be used to express information that VMAP cannot.

Each `<AdBreak>` element may have an optional `<Extensions>` element. The `<Extensions>` element may contain `<Extension>` elements that wrap proprietary XML data. Each extension should express all of its XML data in a unique namespace and use the type attribute for each that identifies the extension. A video

player is not required to support any extensions. A video player may ignore `<Extension>` elements for any extension type that it does not understand.

Each VMAP response may also have a top-level `<Extensions>` element to express custom policies that may not be ad break-dependent.

## 2.8  XML Comments

XML comments are allowed anywhere within the VMAP document. VMAP XML parsers should anticipate the inclusion of XML comments.

```
<!--This is an xml comment.-->
```

# 3  Compliance

## 3.1  Ad Servers

In order to be considered compliant with VMAP 1.0 an ad server must:

- Adhere to the VMAP 1.0 schema
- Support cross-origin retrieval of VMAP responses using industry standard methods (crossdomain.xml file for Adobe Flash, Access-Control-Allow-Origin HTTP header for CORS-enabled user agents, clientaccesspolicy.xml file for Microsoft Silverlight)

## 3.2  Video Players

In order to be considered compliant with VMAP 1.0 video players must adhere to features described within this document with the following exceptions and special notes:

- The video player must be VAST 3.0 compliant. The video player need not implement support for any ad response template other than VAST 3.0.
- The video player must be compliant with the VAST 3.0 Ad Pod format.
- The video player must support VMAP ad breaks of all three types to the fullest extent of its capabilities and that match its VAST 3.0 formats compliance. For example, if the technology on which a video player is built precludes it from supporting non-linear ads, the video player can still be considered VMAP compliant even if it does not support the non-linear ad break type.

# 4 Human-Readable VMAP 1.0 Schema

| Element | Attributes | Values | Req'd | Notes |
|---|---|---|---|---|
| **VMAP** | | **Root Node** | **Yes** | |
| | Version | String | Yes | Current version is 1.0 |
| AdBreak | | | Yes | Represents each ad break. |
| | timeOffset | String | Yes | hh:mm:ss.mmm, "start", "end", n% (n is an integer from 0-100), #m (m represents sequence and is an integer > 0) |
| | breakType | String | Yes | Suggested hint to the player. |
| | breakId | String | No | ID of the ad break. |
| AdSource | | | No | Represents the ad data that will be used to fill the ad break. |
| | id | | No | |
| | allowMultipleAds | | No | Whether or not the player should honor VAST ad pods or other multiple ad formats in the ad response document. If not specified, left to the video player's discretion. Non-VAST ad pods may be ignored. |
| | followRedirects | | No | Whether or not the player should follow wrappers/redirects in the ad response document. If not specified, left to the video player's discretion. |
| VASTData | | VAST 3.0 document | No | A VAST document that comprises the ad response document. Not contained within a CDATA section. |
| CustomAdData | | String (in CDATA) | No | An ad response document (included inline) that is not VAST 3.0. |
| | templateType | vast1, vast2 or proprietary. | Yes | The ad response template employed by the ad response document. |
| AdTagURI | | URI | No | A URL to a secondary ad server that will provide the ad response document. |
| | templateType | vast1, vast2, vast3 or proprietary. | Yes | The ad response template employed by the ad response document. |
| Tracking Events | | | No | |
| Tracking | | URI | No | URI to track for specified event type. |
| | Event | breakStart, breakEnd, error | Yes | The name of the VMAP ad break level event to track. |
| Extensions | | | No | Container for Extensions that provide ability to express information not supported by VMAP. |
| Extension | | Any (XML) | No | The XML content of the Extension. Extension XML must use its own namespace. |

| Element | Attributes | Values | Req'd | Notes |
|---|---|---|---|---|
| | type | String | Yes | The type of the extension. The type value must be globally unique. A URI is recommended. |
| Extensions | | | No | Container for Extensions that provide ability to express information not supported by VMAP. |
| Extension | | Any (XML) | No | The XML content of the Extension. Extension XML must use its own namespace. |
| | type | String | Yes | The type of the extension. The type value must be globally unique. A URI is recommended. |

# 5  Example VMAP 1.0 Response

VAST ad response documents have been removed for brevity.

```xml
<vmap:VMAP xmlns:vmap="http://www.iab.net/vmap-1.0" version="1.0">
   <vmap:AdBreak breakType="linear" breakId="mypre" timeOffset="start">
      <vmap:AdSource allowMultipleAds="true" followRedirects="true" id="1">
         <vmap:VASTData>
            <VAST version="3.0" xsi:noNamespaceSchemaLocation="vast.xsd">
               ...
            </VAST>
         </vmap:VASTData>
      </vmap:AdSource>
      <vmap:TrackingEvents>
         <vmap:Tracking event="breakStart">
            http://MyServer.com/breakstart.gif
         </vmap:Tracking>
      </vmap:TrackingEvents>
   </vmap:AdBreak>
   <vmap:AdBreak breakType="linear" breakId="myid" timeOffset="00:10:23.125">
      <vmap:AdSource allowMultipleAds="true" followRedirects="true" id="2">
         <vmap:VASTData>
            <VAST version="3.0" xsi:noNamespaceSchemaLocation="vast.xsd">
               ...
            </VAST>
         </vmap:VASTData>
      </vmap:AdSource>
      <vmap:TrackingEvents>
         <vmap:Tracking event="breakStart">
            http://MyServer.com/breakstart.gif
         </vmap:Tracking>
      </vmap:TrackingEvents>
   </vmap:AdBreak>
```

```
    <vmap:AdBreak breakType="linear" breakId="mypost" timeOffset="end">
       <vmap:AdSource allowMultipleAds="true" followRedirects="true" id="3">
          <vmap:VASTData>
             <VAST version="3.0+" xsi:noNamespaceSchemaLocation="vast.xsd">
                ...
             </VAST>
          </vmap:VASTData>
       </vmap:AdSource>
       <vmap:TrackingEvents>
          <vmap:Tracking event="breakStart">
             http://MyServer.com/breakstart.gif
          </vmap:Tracking>
       </vmap:TrackingEvents>
    </vmap:AdBreak>
</vmap:VMAP>
```