



**iab.TECH LAB**

**Server-Side Ad Insertion (SSAI)  
Digital Video Ad Serving  
Template (VAST) Macro Guidance**

SSAI VAST Macro Guidance: Version 1.0

Released November 2020

Drafted by the Digital Video Technical Standards Working Group  
Please email [video@iabtechlab.com](mailto:video@iabtechlab.com) with any feedback.

The Digital Video Technical Standards Working Group consists of members from the following companies:

- A Million Ads
- A+E Networks
- ABC TV Network
- Activision Blizzard Media
- Ad-ID
- AdColony
- Adform
- AdGear
- ADLOOX
- Admixer EU Gmbh
- Adobe
- AdPushup
- Adswizz
- AdTheorent
- AI Music
- Amazon Advertising
- Anyclip
- Aotter
- AppNexus
- Audit Bureau of Circulations UK
- Axel Springer SE
- BARC India
- Bidstack
- Blue 449
- Bonzai
- BroadSign
- Browsi
- Cadent
- Canvas Worldwide
- CBS Interactive
- Centro
- Collectcent Digital Media Pvt. Ltd.
- Comcast
- Comcast Spotlight
- Connatix Native Exchange
- Conversant Media
- Conviva
- Cox Media Group
- Cyber Communications Inc.
- Cyberagent, Inc.
- Dailymotion
- Dentsu Aegis
- Digital Advertising Consortium Inc.
- Digitas LBI
- Disney Interactive
- Disney Streaming Services
- Display.io
- District M
- DoubleVerify
- EMX Digital
- ESPN
- Experian Marketing Services
- Extreme Reach
- Flashtalking
- Flipboard
- Forensiq
- FreeWheel
- Fyber
- Google
- GroupM
- GumGum
- Hearst
- Hulu
- HyperTV
- IAB
- IAB Europe
- IAB Germany
- IAB Russia
- IAB Ukraine
- Index Exchange
- InMobi
- Innovid
- Integral Ad Science
- Invisibly
- Jivox
- Jukin Media
- JW Player
- Kargo
- KERV Interactive
- Konduit
- Line
- Magnite
- MediaMath
- Meetrics
- Microsoft Advertising
- Mintegral
- NASCAR Digital Media
- NBCUniversal
- NEXD
- Nielsen
- Oracle Data Cloud
- Pagescience
- Pandora
- PGA TOUR
- Philo
- Picalate
- Pluto TV
- Powerinbox
- Premion
- Protected Media
- Publicis Media
- PubMatic
- PulsePoint
- Quantcast
- Roku
- Shanghai MetaX Software Inc.
- Sizmek by Amazon
- Smaato
- Smart AdServer
- Sony Pictures Television
- Sovrn
- Spark Foundry
- Spectrum Reach
- Spotify
- SpotX
- StackAdapt
- Strategus
- Taboola
- Teads
- Terragon Group
- The Media Trust Company
- The New York Times Company
- The Trade Desk
- TikTok
- Timehop
- Tremor Video
- TripleLift
- Triton Digital
- true[X]
- Twitch
- Twitter
- Unruly
- Verizon Media
- Videonow
- ViralGains
- VONWERSCH DIGITAL STRATEGIES GMBH
- WWE
- Xandr
- XUMO
- Yahoo Japan Corporation
- Yieldmo
- Yomedia Network
- YOSPACE
- ZEFR

The IAB Tech Lab Lead for this project was Mike Midden, Director of Product – Ad Experiences.

# Table of Contents

<b>VAST Player Macros Summary</b>	2
<b>Macro Replacement Responsibility</b>	2
<b>Macro Formatting</b>	2
<b>Marking Macro Values as Unknown or Unavailable</b>	3
<b>Recommended Macros</b>	3
Device & Environment Macros	4
Content Macro Passing	5
SSAI / Media Player Specific Macros	6

## VAST Player Macros Summary

Ad servers, measurement partners, and buyers need access to additional data from publishers to gain a clearer view into the environments they are serving into. In 2018, the IAB Tech Lab released a set of standardized VAST player macros to facilitate this transparency across any VAST environment.

VAST player macros enable publisher's media players to provide additional data points into ad requests. They cover passing details such as the user agent and app that the VAST ad is serving into. They are compatible with any VAST 2.0+ environment.

## Macro Replacement Responsibility

VAST player macros are to be filled by the party that performs VAST HTTP requests. Most commonly, for both VAST URLs and tracking pixel URLs, this would be the video player that's executing the ad or the SSAI server on behalf of the video player.

**Note VAST player macros differ from ad server macros that Ad Operations teams enable through ad servers.**

- Ad server macros are proprietary to specific ad server platforms and are replaced from within the ad server
- VAST macros, on the other hand, are standardized across the industry and replaced by the media player or SSAI platform (whichever initiates the VAST request)

## Macro Formatting

Note: Details here are summarized from the [VAST 4.2 specification in section 6.1](#)

All macro names are surrounded by square brackets, e.g.: [EXAMPLE]. These macros are to be replaced with a corresponding value. The entire macro, including brackets, needs to be replaced. Macros can be placed in both the VAST URL as well as the individual tracking URLs within a VAST tag.

For example, if you'd want to replace the [EXAMPLE] macro in the URL `https://mydomain.com/something?test=[EXAMPLE]` with the value `somevalue`, you would get <https://mydomain.com/something?test=somevalue>.

Some macros must be populated as a series of values rather than a single value. These macros use the `Array<T>` data type. This is a list of `T` values where `T` is another data type like `string` or `integer`. When replacing a macro with such a list, the value should be rendered as a set of values separated by a comma (","), and no spacing. For example: the values `stringA` and `stringB` would be encoded as `stringA,stringB`.

When replacing macros, make sure to apply `encodeURIComponent` to any value, to avoid creating invalid URLs. However, note that encoding should be applied to individual values only, not the entire macro replacement string (i.e. unencoded commas should separate distinct values).

For example, to encode the values `abc/def` and `y=z`, you'd replace the macro with `abc%2Fdef,y%3Dz`

Note that each individual value is encoded, but the comma between values is not.

## Marking Macro Values as Unknown or Unavailable

For any macros that are marked as optional or deprecated and where the actual macro is not provided, the following special values must be inserted into the macro to indicate the reason for not providing the information:

If the macro value is...	Then replace macro with...
Value is unknown, but would be shared if it was known.	-1
Value is known, but information can't be shared because of policy or other circumstances (ie no value currently exists for nonlinear AdType).	-2

## Recommended Macros

While the IAB Tech Lab has an official list of 50+ macros, it's technically not possible to support all macros in any given HTTP request simply due to length limitations. To help guide SSAI providers and the industry toward an optimal set of macros, the below list has been curated for sharing basic user agent, environment, and content details that are critical to verification partners and buyers.

Information that may be available via HTTP headers is not always consistent, and often publishers do not have control over what values are forwarded. Hence providing macros such as `[DEVICEUA]`, `[PAGEURL]` and `[SERVERUA]` are recommended. These macros are also increasingly important as the industry moves away from VPAID and HTTP headers.

For a full list of macros, please refer to the IAB VAST macro list page:

<http://interactiveadvertisingbureau.github.io/vast/vast4macros/vast4-macros-latest.html>

## Device & Environment Macros

Device & environment macros are critical for identifying where an ad is being served into. These are most commonly used by measurement partners for verification against fraud and buyer's brand safety rules.

Note example values are shown unencoded; per the VAST spec values should be encoded.

Macro	Description	Examples
[APPBUNDLE]	For app ads, a platform-specific application identifier, bundle or package name and should not be an app store ID such as iTunes store ID. Where required and applicable, but unknown or unavailable, -1 or -2 must be set. When required and not applicable (such as in a web context), 0 must be set.	<code>com.example.myapp</code>
[STOREID]	Platform-specific app store ID such as iTunes store ID. Could be the same as app bundle on some platforms.	iOS/tvOS: 886445756 Android: <code>com.tubitv</code> Roku: 41468
[STOREURL]	The app store URL for the installed app	<code>https://apps.apple.com/us/app/tubi-watch-moviestv-shows/id886445756</code>
[DOMAIN]	Domain of the top level page where the end user will view the ad.	<code>www.mydomain.com</code>
[ADTYPE]	Indicates whether the ad's intended use case was video, audio, or hybrid, as defined in the <code>adType</code> attribute of the VAST element.	Video Audio
[PAGEURL]	The full URL of the top level page where the end user will view the ad. Applicable for web environments	<code>https://www.mydomain.com/article/page</code>
[DEVICEIP]	IP address of the device that is displaying the ad to the end user. Only relevant when another device (server) is making the request on behalf of that client.	IPv4: 8.8.8.8  IPv6: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

[DEVICEUA]	User-Agent of the device that is rendering the ad to the end user. Only relevant when another device (server) is making the request on behalf of that client.	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
		Chrome/51.0.2704.103 Safari/537.36

Note: When considering [PAGEURL] vs. [DOMAIN] it is generally accepted that [PAGEURL] provides a preferably broader depth of information. Yet in some instances, publishers may be required to only transmit top level data due to privacy regulations, in which case [DOMAIN] is an appropriate fallback.

## Content Macro Passing

Content macros help verification partners categorize and evaluate the content against a buyer's brand safety rules.

Macro	Description	Examples
[CONTENTURI]	The URI of the main media content asset into which the ad is being loaded or inserted.	<code>https://mycontentserver.com/video.mp4</code>
[CONTENTID]	<p>The publisher-specific content identifier for the content asset into which the ad is being loaded or inserted.</p> <p>The format of this field is similar to the [UNIVERSALADID] macro, consisting of a registry identifier and a registry-specific content identifier.</p> <p>If you are not using a public registry, you can use your own domain name as the registry identifier.</p> <p>The provided content identifier should uniquely define a specific content asset.</p> <p>Format: <code>registry_id id_value</code> with a space separating <code>registry_id</code> and <code>id_value</code>.</p>	<code>my-domain.com my-video-123</code>

## SSAI / Media Player Specific Macros

SSAI / Media Player Specific Macros help identify the type of player technology that is utilized.

[SERVERUA]	<p>User-Agent of the server making the request on behalf of a client. Only relevant when another device (server) is making the request on behalf of that client.</p> <p>The goal is to allow ad servers to identify who is making the request, so don't use generic HTTP server names like Apache, but rather identify the company and product or service making the request.</p> <p>Suggested format: {service name}/{version} ({URL to vendorinfo})</p>	<p>MyServer/3.0 (+https://myserver.com/contact)</p> <p>AdsBot-Google (+http://www.google.com/adsbot.html)</p>
[SERVERSIDE]	<p>Value indicating if a URL is requested from a client device or a server. This value may be set differently on the request versus tracking URLs, as the request may be made from a server (value of 1 or 2) while tracking URLs may be fired from the client (value of 0)</p> <p>0 -Client fires request or tracking call without server intermediary. 1 -Server fires request or tracking call on behalf of a client. The client told the server to act on its behalf. 2 -Server fires request or tracking call on behalf of another server, unknown party, or based on its own decision, without an explicit signal from the client.</p> <p>0 is the default value if macro is missing.</p>	1
[CLIENTUA]	<p>An identifier of the player and VAST client used. This will allow creative providers to identify the client code used to process and render video creatives.</p> <p>If player name is not available, use "unknown"</p> <p>Suggested format: {player name}/{player version} {plugin name}/{plugin version}</p>	<p>MyPlayer/7.1 MyPlayerVastPlugin/1.1.2</p>



[PLAYBACKMETHODS ]	The value indicating attributes of the inventory.	1 : initiates on Page Load with Sound On 2 : initiates on Page Load with Sound Off by Default
		3 : initiates on Click with Sound On
[PLAYERSTATE]	List of options indicating the current player state.	muted: to indicate the player is currently muted. fullscreen: to indicate the player is currently fullscreen.