



AdTonos

Integration with AdTonos using VAST or DAAST

Version	Description	Author	Date
1.0	Initial version	Wojtek Lichota	02/22/2019
1.1	Advertising ID on mobile devices	Wojtek Lichota	09/07/2019
1.2	Additional parameters	Wojtek Lichota	12/03/2020
1.3	Additional parameters, List of supported Macros	Wojtek Lichota	02/17/2021
1.4	New parameter “volume”	Robert Kawecki	02/09/2022
1.5	New parameters for podcasts	Robert Kawecki	06/09/2022
1.6	Clarify “pricing” parameter values	Robert Kawecki	09/01/2022
1.7	New parameter “device”	Piotr Śmiałkowski	12/20/2022
1.8	New parameter “capacity”	Piotr Śmiałkowski	01/02/2023
1.9	New parameters for bid floor control	Robert Kawecki	04/26/2024

Introduction

AdTonos solution for online radio broadcast replaces on-air commercial break content with targeted advertisements changing spray-and-pray ATL ads into pay-per-play performance marketing. AdTonos uses server-side ad-stitching to put targeted ads into MP3 stream allowing integration without interfering with existing players. Although server-side ad-stitching service cannot access cookies used in traditional client-side tracking, AdTonos puts JavaScript code on web pages with players to take over this task. This JS code may be used to sync cookies, unique user identifiers or GDPR consents.

VAST and DAAST formats

VAST, or Video Ad Serving Template, is the most popular standard for exchanging information about ads. Although initially it was prepared for video but this standard has also been adopted for audio commercials.

DAAST, Digital Audio Ad Serving Template, is a format very similar to VAST but specialized to audio ads.

AdTonos supports both formats.

VAST tag

<https://play.adtonos.com/xml/><stationId>/vast.xml?&listenerId=<listenerId>&sessionId=<sessionId>&ver=4.0&adType=preroll&contentType=audio&targeting=on&cb=<random>

Parameter	Description	Required
stationId	The unique identifier of the station. This identifier is generated after adding the stream in the AdTonos dashboard. Pass a special constant value “ podcast ” to select podcast by URL if you cannot pass its ID directly.	Yes
podcast_url	An alternative way of selecting station ID, for podcasts only. Pass your podcast’s RSS URL.	Yes, if stationId is “podcast”
divisionId	For audio products which are divisible into parts like episodes/chapters, this identifies the part. For podcasts , this is the episode GUID from RSS.	No
listenerId	Unique listener ID (ULID). Parameter enabling identification of the returning listener (new session).	No*

idfa	iOS Identifier for Advertising	Yes, only on iOS devices
gaid	Google Advertising ID	Yes, only on Android devices
sessionId	Unique session ID. It should remain the same for a given listener during one session (e.g. open browser tab)	No
ver	The VAST protocol version to use. The default is 4.0	No
adType	Advertising purpose: preroll or midroll or bannerad	No
contentType	The type of requested ad: audio or video	No
targeting	Information whether the listener has consented to profiling ads (GDPR). Available values on and off. By default on.	No
ip	Listener IP address	No**
lang	Listener's preferred language	No**
userAgent	Listener's browser User Agent	No**
pricing	Force adding <Pricing model="CPV"> tag to VAST response. By default this tag is not present. Pass "1" or "true" to enable.	No
bidfloor	Minimum price acceptable for ads, expressed as CPM (per 1000), formatted as floating point. Example: 2.50	No
bidfloorcur	Currency in which to interpret bidfloor. If not passed, always defaults to USD .	No
speakerId	Unique smart speaker identifier. On Amazon Echo use context.System.device.deviceId on Google Home generate UUID4 and store it action storage .	Yes, only on smart speakers
iva	Allow ads with Your Truly "call-to-action" for virtual Assistants even when speakerId is not set and userAgent do not match smart speakers. Available values amazon and google.	No
referrer or referrer	Website URL that is requesting VAST.	No
bitrate	filter out media-files that have a different bitrate than the indicated one. Supported bitrates: 64, 128, 192, 256 and 320.	No

samplingrate	filter out media-files that have a different sampling rate than the indicated one. Supported sampling rates: 44100 and 48000.	No
cb	Cache Busting - random number (at least 10 digits) to avoid caching responses in the listener's browser	No
content_language	The language of the content where ads will be played. Two-letter code that corresponds to ISO 639-1 language codes.	No
content_country	Country Code of the listener's location. Two-letter code corresponds to ISO 3166-1 alpha-2 country codes.	No
content_type	Content type as defined by the IAB Tech Lab Content Taxonomy V2. Comma-separated list of integer values, each in the 1-698 range.	No
gdpr_consent	IAB TCFv2 consent string. If <code>gdpr_consent</code> does not contain the user's consent to the processing of data by Radio Net Media Ltd., the returned VAST will always be empty.	Yes, if GDPR applies to user
volume	Audio volume setting (%) currently active on the listener's device, in numeric form. Range 0-100. Example: 0 (audio is muted). This prevents serving ads which cannot be heard.	No
device	Bypass client device detection . Available values: desktop, mobile, "smart speaker" or out-of-home	No
capacity	Required ads block duration in seconds.	No

HTTP Headers

Header	Description	Required
X-Forwarded-For or X-Real-Ip	Listener IP address	No**
X-Device-User-Agent or User-Agent	Listener's browser User Agent	No**
X-Device-Accept-Language or Accept-Language	Listener's preferred language	No**

Referer	URL of the page where the player is embedded	No*
---------	--	-----

* Parameter / header is not required in case the VAST request comes directly from the listener's browser or mobile application. The parameter / header is required when the VAST tag is used in the Server-to-Server approach.

** The ip, lang and userAgent parameters or the X-Forwarded-For, X-Device-AcceptLanguage and X-Device-User-Agent headers are not required when the VAST request comes directly from the listener's browser. If the VAST tag is used in a Server-to-Server approach, it is required to pass a pair of parameters or a pair of headers.

DAAST tag

DAAST tag uses the same parameters as VAST tag.

`https://play.adtonos.com/xml/<stationId>/daast.xml?&listenerId=<listenerId>&sessionId=<sessionId>&ver=4.0&adType=preroll&contentType=audio&targeting=on&cb=<random>`

Macros

Impressions and tracking events may contain macros - for example

[http://example.com?bb=\[CACHEBUSTING\]](http://example.com?bb=[CACHEBUSTING]). AdTonos support the most popular VAST 4.x macros:

- [\[CACHEBUSTING\]](#)
- [\[TIMESTAMP\]](#)
- [\[ASSETURI\]](#)
- [\[PODSEQUENCE\]](#) with alias [ABPOS]
- [\[ADCOUNT\]](#) with alias [COUNTER]
- [\[PAGEURL\]](#) with alias [URL]

JS code (Web only)

When the player that uses VAST or DAAST tag is embedded on the web page it is necessary to embed JavaScript code on the website containing the player. This code is used to profile ads and synchronize Unique listener ID (ULID).

Embedding involves inserting the JS code at the end of the page, just before the closing tag `</body>`.

```
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

where {name} is variable provided but AdTonos employee.

Additionally, it is possible to pass the listener's disagreement on profiling. If a website uses TCF 2.0 popup to gather user consents it's not required to use `window.adtonosOptOuts` variable. To do this it should be done before (above) mentioned JS code:

```
<script>
window.adtonosOptOuts = ['*'];
</script>
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

Example JS code when another variable specifying the consent for profiling is available:

```
<script>
if (!gdprAllowPersonalizedAds) {
  window.adtonosOptOuts = ['*'];
}
</script>
<script async defer src="https://play.adtonos.com/attc-{name}.min.js"></script>
```

Advertising ID (Mobile only)

When the player that use VAST or DAAST tag is embedded in mobile application it is necessary to pass `idfa` or `gaid` parameter.

iOS Identifier for Advertising (idfa)

On iOS 6+ devices use:

```
NSString *idfaString = [[[ASIdentifierManager sharedManager]
advertisingIdentifier] UUIDString];
```

Note that the user can opt-out from the iOS Device Advertising ID, so the application must handle cases where this ID is not available and send empty ID.

<https://developer.apple.com/documentation/adsupport/asidentifiermanager>

Google Advertising ID (gaid)

Google Advertising ID can be retrieved from Google Play services. Please follow official documentation:

<https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient>

Note that the user can opt-out from the Google Device Advertising ID, so the application must handle cases where the ID is not available and send empty ID.

Help

In case of problems or questions regarding integration, please contact us by email: support@adtonos.com.