

Unified ID Adoption Guidelines for SSPs

October 2019

Contents

- A Device ID Equivalent Designed for the Open Web 2
- Solution Overview..... 2
- TDID for SSPs Without Header Wrapper Support..... 3
 - Set-Up Procedure..... 3
 - Step 1 - Adsrvr.org Setup 3
 - Step 2 - Publisher Page Setup..... 3
 - Step 3 - Bid Request Flow 5
- TDID for SSPs With Header Wrapper Support 6
 - Realtime ID Service Overview 6
 - JSONP Format 6
 - JSON Format 7
 - Set-Up Procedure..... 7
 - Step 1 - Adsrvr.org Setup 7
 - Step 2 - Publisher Page Setup..... 8
 - Step 3 - Bid Request Flow 8
- General Note on Disabled Third-Party Cookies 9
- Privacy and Compliance..... 9
 - Opt-Out..... 9
 - GDPR..... 10
- Changes to this Specification 10

A Device ID Equivalent Designed for the Open Web

Adserver.org and its sister ad-serving domain Adsrvr.org were created to enable ad tech companies to collaborate on the foundation of anonymous identity that our industry relies on to deliver a personalized advertising experience to customers and great results to advertisers. The Trade Desk, Inc. (NASDAQ:TTD) has been using this domain for eight years. We write an ID called TDID into the top-level cookie in this domain along with the metadata describing the status of mapping that cookie to partners. In H2 2017, TTD started making the TDID available for other parties to use it as their primary ID to better reflect the collaborative intent of the domain. The implementation is identical, and TTD continues to own and operate the domain. It is a Google AdX-approved fourth-party domain, it is included in the National Advertising Initiative (NAI) opt-out tool, and it has a privacy policy to which all adopters must adhere.

Solution Overview

SSPs currently sync cookies with Adsrvr.org at various priority levels. Given the concentration of demand that is driven from Adsrvr.org, at a minimum SSPs should make it their top cookie mapping priority and fire the pixel from anywhere they can on their footprint. JavaScript pixels should be used whenever possible so that mapping can be made in parallel.

Currently, any partner set up to do cookie mapping with TTD issues their own user ID and fires cookie sync pixels across their footprint. The result of the cookie mapping is that they can store a map of their user ID to the TDID in a server-side match table and/or a cookie. For SSPs, this allows the TDID to be looked up at bid time and included in the bid request.

Going forward, TTD will start allowing the use of the TDID the primary user ID for any partners that would like to participate in this initiative. Partners will be able to use the existing cookie mapping endpoints that they're already using to extract the TDID and then simply use it as their own ID. When a partner identifies a new user, they can redirect the user to Adsrvr.org's cookie mapping endpoint which will either extract the existing TDID from Adsrvr.org cookies or issue a new TDID and redirect back to the partner's configured endpoint. Essentially, partners will use the Adsrvr.org endpoint as an ID-issuing service.

For SSPs supporting header wrappers, Adsrvr.org now has a new endpoint for supporting multiple consortium IDs, such as Advertising ID and others. A consortium ID can be fetched simultaneously with a TDID in real time, and before the SSP makes a bid request from the user's browser. This is to further improve the percentage of requests that have a user ID and are thus targetable by buyers. This endpoint can either be used by SSPs pre-bid to get the user IDs or may also be included in header bidding tags such that there's a single call to this endpoint at the top of the header bidding tag and the user IDs will then be made available to any SSPs participating in that tag for that impression.

TDID for SSPs Without Header Wrapper Support

This section describes how to use the TDID for SSPs that do not support header wrappers.

Set-Up Procedure

Step 1 - Adsrvr.org Setup

All adopting parties will need to be configured as a partner in Adsrvr.org system. This is the same setup for a standard cookie mapping to Adsrvr.org.

(If you are already set up with a standard cookie mapping to Adsrvr.org, you can skip this section and move to Step 2.)

Adopting parties will provide URLs (`http` and `https`) to their server with two required query string parameters, one for the TDID and another for the TTL. These will be stored internally in the database in Adsrvr.org with macros that are substituted with the values before being redirected back to the partner's endpoint.

Here is an example of one such URL.

```
http(s)://partnername.com/pixel?tdid=%TDID%&ttdl=%EXPIRATION_UNIX%
```

where:

- `%TDID%` is substituted with the TDID generated by Adsrvr.org, and
- `%EXPIRATION_UNIX%` is substituted with timestamp of the expiration time, at which point the adopting party will need to refresh the TDID. The expiration time is in terms of the seconds elapsed since the Unix epoch at 01/01/1970.

As mentioned above, Adsrvr.org supports the federation of alternative consortium's user IDs. If Adsrvr.org has a consortium user ID available at the time it receives the call from a partner's cookie sync pixel, it will be added onto the partner's URL as another query string parameter; for example, `appnexus_id`. Any additional user IDs that we federate in the future will be available as additional query string parameters. We will provide a table of these query string parameter names, so that code may be added at the partner's server to parse them out and use them as appropriate.

Note: Adopting parties will receive a Cookie Mapping Partner ID, created in the Adsrvr.org system. This ID is required in step 2.

Step 2 - Publisher Page Setup

You may already be deploying JavaScript or pixels on a publisher page that kick off cookie mapping with Adsrvr.org and other partners.

Typically, JavaScript or a pixel on such a publisher page hits the SSP's server that can detect if the user is a new user. If so, the SSP needs to redirect the new user's browser to the Adsrvr.org cookie

mapping URL for Adsrvr.org to issue a TDID (if it is a new user for Adsrvr.org) or to simply return the existing TDID (which will be read by the Adsrvr.org server from a cookie in the Adsrvr.org domain).

Format the cookie mapping URL for Adsrvr.org like this:

```
http(s)://match.adsrvr.org/track/cm/generic?ttd_pid=  
<cookiePartnerId&ttd_tpi=1
```

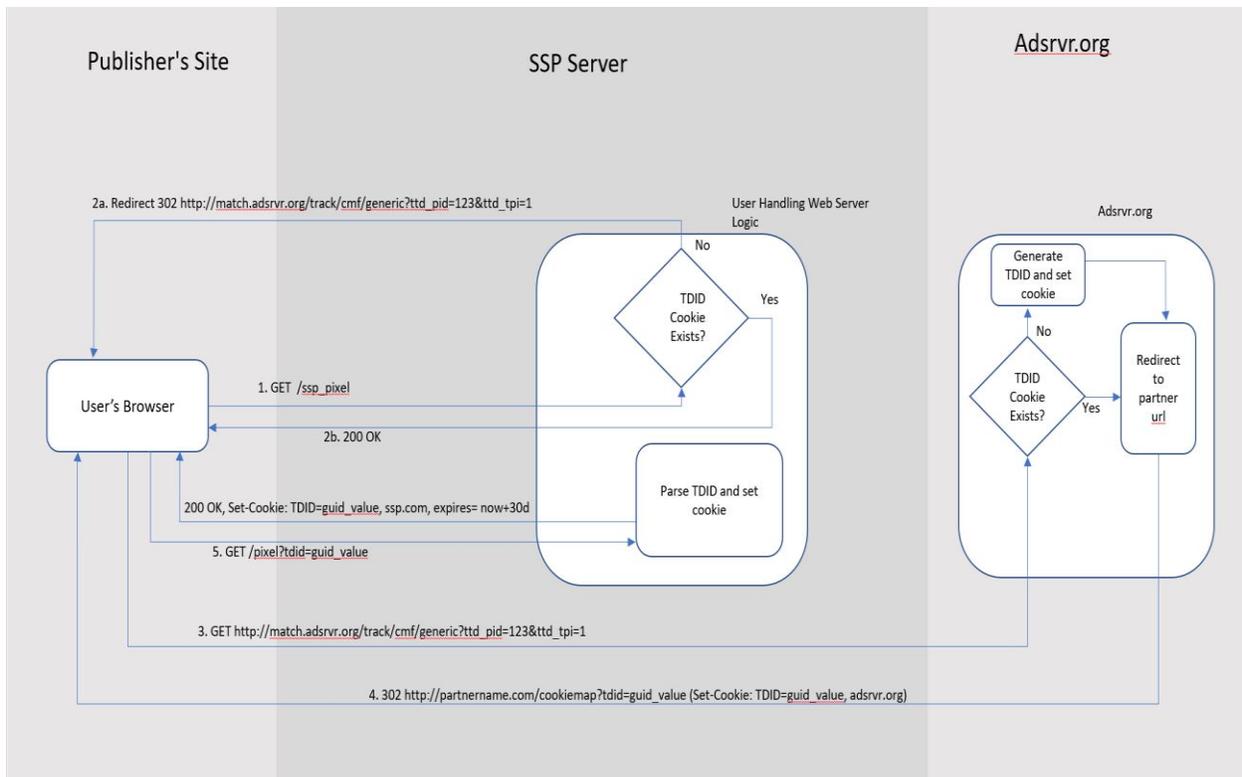
where:

- `<cookiePartnerId>` is to be substituted with the Cookie Mapping Partner ID received at the time of Adsrvr.org setup, and
- The `ttd_tpi` query parameter indicates that this is a cookie mapping request initiated by the partner. Adsrvr.org will then redirect the user's browser to the partner URL provided at the time of setup and substitute the `%%TDID%%` macro with the TDID for the user and the `%%EXPIRATION_UNIX%%` macro with the expiration time, as described in Step 1.

Note: If you are a partner SSP already using a customized version of the cookie mapping URL to Adsrvr.org, that will mapping continue to function as before. We do not require you to switch to the format outlined in the example above. If you do use a customized version, Adsrvr.org will infer the `<cookiePartnerId>` from the path of the endpoint itself.

The result of this mapping will be the user's browser being redirected to the partner URL. On the server, the SSP partner can then parse the TDID value received in the query string. Typically, at this point in the flow SSP partners have stored a map of their own user ID to the TDID in a user match table. As part of the migration to using TDID as the primary user ID, SSP partners are encouraged to start writing the TDID into their own cookie. This will make it so that the TDID is always available at the time of the bid request and will not require any extra lookups in the match table.

SSP partners are encouraged to re-sync the ID with Adsrvr.org at the TTL interval specified on the query string. Re-syncing helps minimize drift resulting from the user deleting their cookies or some such case that results in Adsrvr.org issuing a new TDID to the user, after the partner has received the originally issued TDID. The following diagram explains the flow described above.



Step 3 - Bid Request Flow

At this point in the implementation, as an SSP, you would have setup the Adsrvr.org configuration and deploying any necessary JavaScript or pixels on the publisher page such that you are getting a TDID for every user that you see, new or existing. Also, you would have started writing the TDID into a cookie in your domain.

The final step of the flow is to now start passing the TDID on every bid request to every DSP with which you are integrated. We recommend passing all the IDs that you received in the redirect back to your server from Adsrvr.org on the publisher page, on the bid request to the DSPs. This will likely mean using an extension object to make these IDs available for the DSP to choose which ID they want to use to transact RTB.

Note: This final step is critical in order to propagate the flow of the TDID across the RTB fabric.

TDID for SSPs With Header Wrapper Support

This section describes how to use the TDID for SSPs that support header wrappers.

Currently, header wrappers themselves do not have any built-in support for establishing the user ID of the user on the publisher page. This is left to each of the SSPs participating in the header wrapper by firing their own pixels and mapping with any partners in the priority order that they've established.

In order to increase the availability of TDID across the RTB fabric, Adsrvr.org has implemented a new endpoint, the Real Time ID Service, that may be used by header wrappers. Code in the wrapper will make a call to this new service, which will return the TDID and any other consortium IDs. The header wrapper code can then make these IDs returned in the response to the SSPs participating on the header wrapper. Any SSP that's attempting to map with Adsrvr.org will no longer need to do this because the TDID will always be available. SSPs may still need to fire cookie mapping pixels in order to map with any of their partners that are not using one of the IDs returned by the real time ID service.

Realtime ID Service Overview

Adsrvr.org now has a new endpoint that can be accessed directly from the user's browser on the publisher's site, in order to fetch the TDID (and alternative participating consortium IDs, such as AppNexus' user ID, etc.) before the bid request is made. The goal with this endpoint is faster propagation of users amongst adopting parties, thus improving the coverage and match rate for all parties involved. There are formats of responses that are supported as described below. A header wrapper may choose one of these in order to integrate the Realtime ID Service.

JSONP Format

The endpoint will need to be included via a script tag on the page where an SSP needs to be able to access the user ID, like this:

```
<script src=
"http://match.adsrvr.org/track/rid?v=1.0.0&ttd_pid=partnerid"
type=javascript />
```

where

- The `v` query string parameter is a version identifier allowing for updating the responses returned in newer versions.
- The `ttd_pid` query string parameter is a Cookie Mapping Partner ID that you will receive as part of the Adsrvr.org setup, described below.

The response from the endpoint will be JavaScript that will invoke a JavaScript function with the argument being a JSON of the user IDs that are currently known to Adsrvr.org like this:

```
OpenID.getID({"ttdid": "D6885E90-2A7A-4E0F-87CB-7734ED1B99A3",
"appnexus_id": "1234"});
```

Consumers of this endpoint will need to implement a JavaScript function named `OpenID.getID` so that the response from the endpoint will invoke it and pass along the IDs as shown above. Note that the actual values are just examples and the formats of the IDs can change at any time. Clients should not make any assumptions about the format other than the IDs are strings.

JSON Format

The endpoint can be invoked via an XHR request from JavaScript code. If you use this format, it is critical that you implement some caching mechanism on the page where you are consuming this endpoint, so as not to make a call to this service for every execution of the header wrapper on a publisher page. Please make sure that the XHR request has “withCredentials” enabled (<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/withCredentials>) so that cookies can be properly synced. You may use your own cookie or local storage to cache the response from the endpoint. You should cache the response for 30 days before making a subsequent request to this endpoint again. The URL to be used for invoking this format will be:

```
http://match.adsrvr.org/track/rid?ttd_pid=partnerid&fmt=json
```

where:

- The `ttd_pid` query string parameter is the same as described in the section above, the Cookie Mapping Partner ID that you will receive as part of the Adsrvr.org setup.
- The `fmt` query string parameter is used to indicate the chosen JSON response format. Versioning is not supported on this response format since the response will always be a simple JSON with key/values for each of the IDs returned by this endpoint.

The response is JSON that looks like this:

```
{ "ttdid": "D6885E90-2A7A-4E0F-87CB-7734ED1B99A3",  
  "appnexus_id": "1234" }
```

Set-Up Procedure

Step 1 - Adsrvr.org Setup

All adopting parties will need to be configured as a partner in Adsrvr.org system. This is the same setup for a standard cookie mapping to Adsrvr.org.

Adopting parties will provide URLs (`http` and `https`) to their server with two required query string parameters, one for the TDID and another for the TTL. These will be stored internally in the database in Adsrvr.org with macros that are substituted with the values before being redirected back to the partner’s endpoint.

Here is an example of one such URL.

```
http(s)://partnername.com/pixel?ttdid=%TDID%&ttdl=%EXPIRATION_UNIX%
```

where:

- `%TDID%` is substituted with the TDID generated by Adsrvr.org, and

- %%EXPIRATION_UNIX%% is substituted with timestamp of the expiration time, at which point the adopting party will need to refresh the TDID. The expiration time is in terms of the seconds elapsed since the Unix epoch at 01/01/1970.

As mentioned above, Adsvr.org supports the federation of alternative consortiums user IDs. If Adsvr.org has a consortium user ID available at the time it receives the call from a partner's cookie sync pixel, it will be added onto the partner's URL as another query string parameter; for example, `appnexus_id`. Any additional user IDs that we federate in the future will be available as additional query string parameters. We will provide a table of these query string parameter names, so that code may be added at the partner's server to parse them out and use them as appropriate.

Note: Adopting parties will receive a Cookie Mapping Partner ID, created in the Adsvr.org system. This ID is required in step 2.

Step 2 - Publisher Page Setup

On the publisher page, the header wrapper should make a call to the Realtime ID Service, using either the JSONP format or JSON format. The result of calling this endpoint will be that the wrapper code will receive the TDID along with any other IDs that are available to Adsvr.org at the same the call is made. The wrapper should then pass the IDs to the SSP bidder adapters that are configured as part of the header wrapper.

Step 3 - Bid Request Flow

SSPs that are participating as part of the header wrapper and are onboarded onto Unified ID will now be required to start passing the TDID on every bid request to every DSP that you are integrated with. We recommend passing all the IDs that you received in the redirect back to your server from Adsvr.org on the publisher page, on the bid request to the DSPs. This means using an extension object to make these IDs available for the DSP to choose which ID they want to use to transact RTB.

Note: This final step is critical in order to propagate the flow of the TDID across the RTB fabric.

Please use the following extension within the user property to pass the additional IDs, like this:

```
{
  "user": {
    "id": "W13bZtHM5KUAAEQRVFsAAAAAL",
    "ext": {
      "eids": [{
        "source": "adserver.org",
        "uids": [{
          "id": "43f888e1-2941-4b9f-8dc9-1fffb631b7786",
          "ext": {
            "rtiPartner": "TDID"
          }
        }]
      }]
    }
  }
}
```

General Note on Disabled Third-Party Cookies

If third-party cookies are disabled in a user's browser, as they are by default in Safari, the cookies set by an SSP, DSP, or DMP will be rejected because these are set on the response to a pixel firing in a page that's running in a different top level domain, such as the publisher's or the advertiser's domain. If cookies can't be set and used at ad serving time, it is impossible to have any reliable notion of identity.

Privacy and Compliance

The EULA (www.thetradedesk.com/general/uid-eula) sets out requirements to which adopters must adhere.

Opt-Out

All adopters must honor user preferences. Adopters should check for the opt-out state at least every 14 days.

A list of opted-out IDs will be available via service, details are TBD.

The opt-out state is also available on resync; an opted-out client will return os on resync.

Once an opt-out is received, it should be honored accordingly. This includes no further use of the opted-out ID, and may include passing it on to other parties, depending on your circumstances.

If the ID has changed between re-syncs, you must update the ID. You must not tie together data with different IDs for the same device from the ID service across deletion or opt-out events. For example,

if a user clears cookies, and subsequently gets a new ID, you should not try to merge data from the pre-cookie clearing ID with data from the post-cookie clearing ID.

You should treat data associated with expired IDs according to your retention policies and other legal requirements.

GDPR

The ID issuing service supports the IAB Transparency & Consent Framework (TCF) and will expect to receive an appropriate TCF legal basis signal with a GDPR-applicable ID request. When you receive IAB TCF signals you should honor them according to IAB TCF policy.

Changes to this Specification

The service and this specification may be updated from time to time. Upon such updates, adopters' integrations must be updated to conform within a commercially reasonable time.