Table of Contents

Get Started 4
- Publisher 4
- Advertiser 8
- SSP 12
- DSP 18
- Data Platform 22

Retrieve ID5 IDs 27
- ID5 Javascript Library 27
- ***Prebid*** 29
  - **Prebid.js** 29
    - ID5 User ID Module for Prebid.js 29
    - ID5 Prebid.js User ID ModuleTroubleshooting 40
    - ID5 Prebid.js Analytics Module - Identity Insights 42
    - Minimum Prebid.js Integration Requirements for ID5's Real Time Bid Enrichment Service 46
  - **Prebid Server Java** 49
    - ID5 User ID Module for Prebid Server Java 49
- ***Mobile In App*** 55
  - Mobile In-App Integration - Client-to-Server 55
  - Mobile In-App Integration - Server To Server 57
- CTV Integration 66
- Passing Signals to ID5 77
- Adobe Experience Cloud 82
- Amazon (APS) Integration 84
- Google Secure Signals 87
- Permutive 90
- Signal Obfuscation 92
- TrueLink Integration 93
- Guarded Publisher ID 97
- Publisher Provided ID (PPID) Provisioning 99
- ID5 JS API Lite 100

Data Onboarding 102
- ***ID5 Direct*** 102
  - Data Onboarding - Match Preview 102
- ***Snowflake*** 104
  - ID5 Match Preview - Snowflake Native Application 104

Cookie Sync with ID5 111
- Initiate Cookie Sync to ID5 111
- Receive Cookie Sync from ID5 114

Access Identity Graphs 118
- Overview of ID5 Identity Graphs 118
- Client Data Sharing Specification 121
- Partner Graph File Transfer 125
- Matching File Transfer 128

Privacy 131
- Read First: Privacy Policy Requirements 131
- DSAR - Data Deletion API 135

Mobile Opt-in/Opt-out API (deprecated) 141
User Rights Propagation 144
Opt-out API 148
ATT Signal Processing (iOS) 153
Campaign Measurement with ID5 155
ID5 Metadata 157
Audience Activation 159
Audience Activation in Google Ad Manager (GAM) 159
Enrich 159
Real Time Bid Enrichment Service 159
Enrich for Publishers (based on ID5 Graph) 164
Enrich for SSPs (Offline graph) 168
ID5 OpenRTB ID Provenance Support and Best Practices 172
Enriching Bid Requests with the ID5 ID 175

Publisher

12/04/2025 12:00 pm EST

Core Value Proposition

1. Prevent data leakage by permissioning user IDs to authorized partners only
2. Ensure compliance with data privacy regulations thanks to ID5's privacy-by-design technology
3. Identify authenticated and unauthenticated users in all digital advertising environments to increase monetization

ID5 ID Overview

The ID5 ID is a shared, neutral identifier that allows publishers and ad tech platforms to recognise users in environments where third-party cookies are unavailable. ID5 enables publishers to generate and distribute a common first-party identifier across the ecosystem. Ad tech partners integrated with ID5 can decrypt the ID5 ID, provided they have the appropriate user and publisher consent, to enhance their ability to recognise users. The ID5 ID is built to honour user privacy choices and publisher preferences at every stage of the advertising value chain.

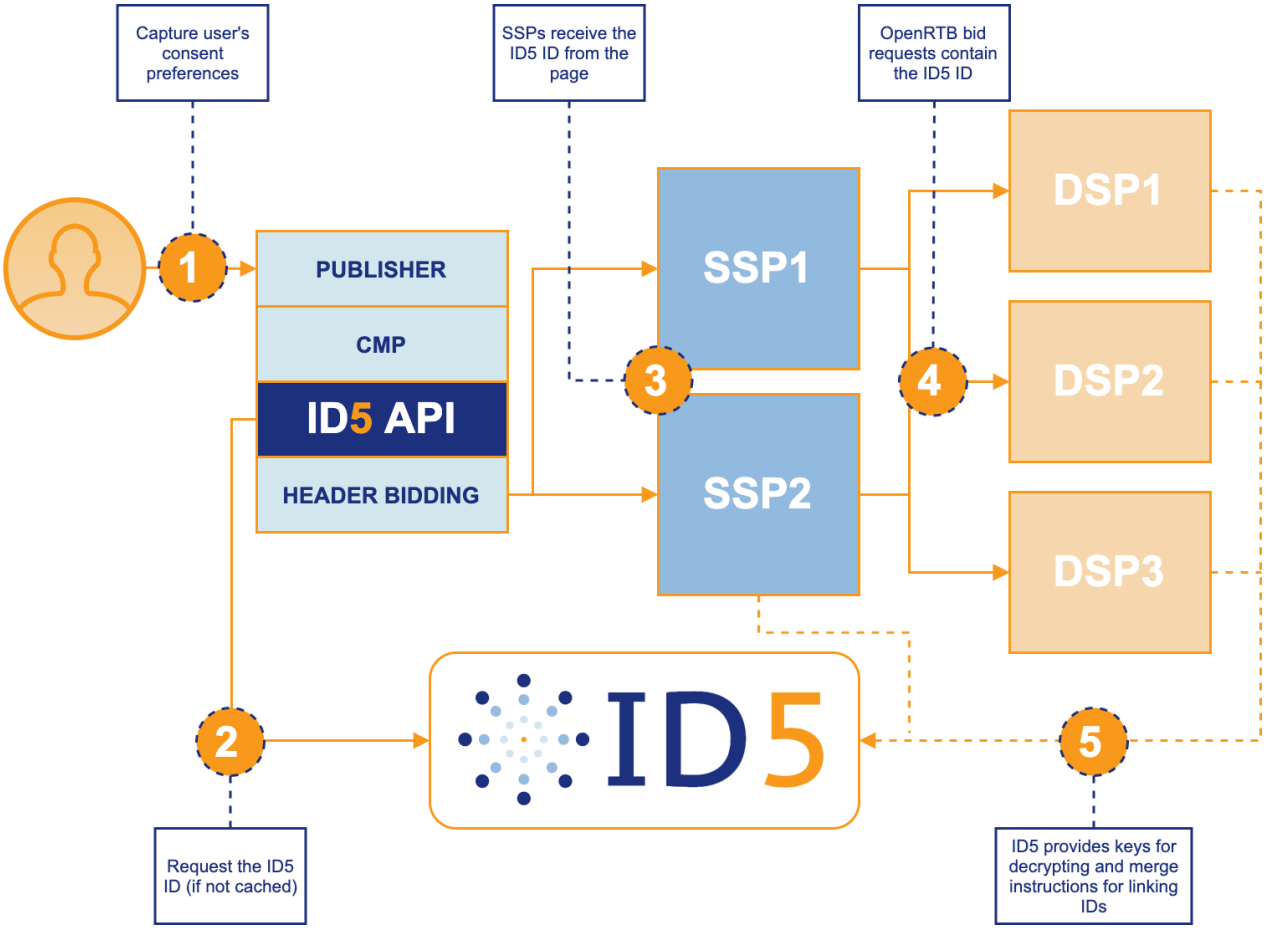
How does the ID5 ID work?

The ID5 ID provides a common, privacy-conscious identifier that replaces the need for ad tech platforms to maintain and sync their own proprietary user IDs. Instead of each platform speaking its own identity "language," the ID5 ID gives the entire ecosystem a shared way to recognise users, even where third-party cookies or device identifiers are not available.

Once implemented, publishers can expose the ID5 ID to partners to increase addressability and serves as a stable key for building, targeting, and activating audiences. Partners can pass the encrypted ID through their existing tags or pixels and in the bid stream enabling them to reliably identify users for audience creation, data activation, bidding, optimisation, and measurement across all environments.

The ID5 ID can be used alongside the ID5 Identity Graph, to support people-based advertising strategies..

Solution Overview



ID5 Integration Overview

Phase 1: Deploy & Distribute

- Deploy the ID5 ID across your user base using the Prebid.js User ID module, ID5 JS API, Mobile In App and CTV APIs or through one of our integration partners such as Google or Amazon Publisher Services. Ensure you share the ID5 ID in the user.eids field in OpenRTB bid requests using these instructions.
- Tell your platform partners that you have integrated with ID5 and the ID5 ID is now available for all of your users.

Phase 2: Measure

- Deploy the Prebid Analytics module alongside AB Testing and work with your Account Manager to analyse the value of ID5 on your business.

Phase 3: Read & Use

- Publishers with many properties can decrypt the ID5 ID to access the persistent identifier and use it as a basis for building and targeting audiences, optimising campaigns as well as measurement and attribution across properties.
- Encourage your platform partners to decrypt the ID5 ID and use it as a basis for building and targeting audiences, executing programmatic deals, optimising campaigns as well as measurement and attribution. Refer to our integration guides for SSPs, DSPs and Data Platforms.

Deploying the ID5 ID

Contract

Before getting started with deploying the ID5 ID, we need to make sure you have signed a Site Agreement and have been issued an ID5 Partner Number. If you are not already integrated with ID5, reach out to contact@id5.io or sign the agreement and we'll get you set up right away.

Deployment

There are multiple ways to deploy the ID5 ID across your properties as outlined below. Where possible, we recommend passing additional signals such as hashed email addresses and MAIDS in requests to ID5 to enhance the accuracy of reconciling consented users across domains and devices.

Prebid.js User ID Module

If you are using Prebid.js, you can deploy the ID5 ID by including both the `userId` and `id5IdSystem` modules, in addition to the other modules you normally include in your prebid configuration. Full instructions can be found in our Prebid documentation.

ID5 JS API

If you are not using Prebid.js, you can install the ID5 JS API directly on your page after your CMP (if applicable), but as high in the `<head>` as possible. Full instructions can be found on our GitHub page.

Mobile App Deployment

ID5 can be deployed in the mobile in app environment. For a complete guide to the specifics of integrating in-app, please see our Mobile In-App Integration page.

CTV Deployment

The ID5 ID can be deployed in CTV environment as outline here.

Inform your partners

Once the ID5 ID has been deployed across your sites, we recommend you reach out to your ad tech partners to let them know that you have integrated with ID5 and have the ID5 ID deployed. Please contact your ID5 Account Manager if you would like assistance with content for this communication.

For an optimal integration, your:

- **SSP partners** should be retrieving the ID using the Prebid.js Bid Adapter and/or the ID5 JS API and sending them to DSPs via OpenRTB bid requests.
- **Data Platform partners** should be using the ID as a basis for building audience segments and syncing with your SSP and DSP partners.
- **DSP partners** should be decrypting the ID5 IDs they are receiving in OpenRTB bid requests and using it in their bidding logic as well as to communicate identity with their other platform partners including data management platforms.

Measure the Value of the ID

To help publishers better understand the value of working with ID5, we have launched an Analytics Module for Prebid. With just a few additional lines of configuration, publishers can use ID5's analytics platform to dig into the data without having to build their own reporting tools.

Read and Use the ID5 ID

Publishers operating across multiple environments may wish to use the ID5 ID combined with the ID5 graph to build audience segments and activate omnichannel campaigns. To enable this, the ID5 ID must be **decrypted** so that the underlying persistent identifier can be accessed. Once decrypted, the ID5 ID can be used in conjunction with the ID5 graph which aggregated ID5 ID into individuals and household to support omnichannel audience creation, campaign optimisation, measurement, and attribution.

Privacy & Regulations

Privacy-by-Design

ID5 has built a privacy-by-design shared ID service for publishers and ad tech vendors. Our service leverages the IAB's Transparency and Consent Framework (TCF) and US Privacy Framework to capture the user's privacy preferences.

As a shared ID provider, ID5 acts as a Controller of the ID5 ID, and thus, we must receive a valid legal basis to process requests. When we receive a request for the ID5 ID, we check that we have a legal basis to store our user ID in a cookie before proceeding; if we don't have one, we do not read our cookie or write to it as part of the HTTP response.

When ID5 returns an ID to the page, the value is encrypted in such a way that only platforms that have authorization to process data (based on the consumer's and publisher's privacy preferences) are able to decrypt the string back to a stable ID. By doing so, ID5 enforces privacy preferences and regulations, ensuring that no downstream party can understand the ID without the proper legal basis to do so. When the ID is non-decryptable, the request is truly anonymized, preventing any personal data from being retrieved or processed.

Privacy Policy

For further information please visit our Platform Privacy Policy, please visit https://id5.io/platform-privacy-policy and Privacy API documentation.

Advertiser

01/09/2025 9:22 am EST

Core Value Proposition

1. Address users in quality environments and reduce reliance on dominant media platforms
2. Apply key campaign strategies in cookieless environments and get ready to transition to the post-cookie world
3. Ensure that consumers' privacy preferences are respected and enforced in the advertising value chain

ID5 ID Overview

The ID5 ID is a shared, neutral identifier that publishers and ad tech platforms can use to recognize users even in environments where 3rd party cookies are not available or blocked. ID5 enables publishers to create and distribute a shared 1st party identifier to the entire ecosystem. Ad tech platforms that connect with ID5 can decrypt the ID5 ID and improve their user recognition capabilities. The ID5 ID is designed to respect users' privacy choices and publishers preferences throughout the advertising value chain.

How does the ID5 ID work?

By using the ID5 ID, ad tech platforms can eliminate the need to sync their platform-specific IDs with their partners - the equivalent of needing a translator to help two people speaking different languages understand each other. When all platforms are using the ID5 ID to transact against, it's like they're all speaking the same, common language. After deploying the ID5 ID across your user base, you can make the ID5 ID available to any partners via a single Javascript variable. Your partners pass the ID to their platforms via their existing tags/pixels and can use the ID5 ID to identify the user for data collection/aggregation, bidding, optimization, etc., even when third party cookies are not supported.

Solution Overview

1. The Advertiser first loads its CMP and captures the user's consent preferences. This is essential before any IDs are requested or delivered.
2. The Advertiser calls ID5 (via the ID5 JS API, a header bidding identity module, or server-to-server) to request the ID5 ID, which can then be placed in cache (in the user's browser or the advertiser's server) to avoid unnecessary http requests on future page views.
3. The DSP's tags and pixels retrieve and log the ID5 ID alongside any data.
4. The data platform's tag on the advertiser's page retrieves the ID5 ID and passes it, along with any other data signals they normally use, to their servers for processing
5. The data platform pushes the data to the DSP and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass

6. The Publisher first loads its CMP and captures the user's consent preferences.

7. If consent is given, the user is assigned an ID5 ID.

8. Via the publisher's header bidding wrapper, or through a direct integration, SSPs receive the ID5 ID in the ad request to their servers

9. SSPs pass the ID5 ID into the OpenRTB bid request that they send to their DSP partners. Outside of the RTB process, DSPs receive keys from ID5 so they can properly decrypt the value they receive in bid requests. DSPs should also initiate and receive cookie sync requests to ID5. DSPs should then leverage the ID5 ID in the same way they use their proprietary platform user ids; for segment building, audience targeting, campaign optimization, measurement and attribution.

ID5 Integration Overview

Phase 1: Deploy

Deploy the ID5 ID across your user base using the ID5 JS API. When available, we recommend passing deterministic signals such as hashed email addresses and mobile ad ids into requests to ID5 to enhance the accuracy of reconciling consented users across domains and devices.

Phase 2: Read, Share & Use

Encourage your platform partners to integrate with ID5

- **DSPs**
    - Initiate to and receive cookie sync requests from ID5
    - Deploy and retrieve the ID5 ID by integrating the ID5 JS API in any tags or tracking pixels
    - Decrypt the ID5 ID received in bid requests and use it in the same way as their proprietary platform user id, to inform campaign targeting, optimisation, attribution and measurement on cookieless impressions and all impressions when third party cookies are deprecated
    - Sync with their Data platform partners using the ID5 ID
- **Data Platforms**
    - In any data collection tags on publisher and advertiser pages
    - Decrypt the ID5 ID and use it as a basis for building segments and pushing segments to partner platforms such as your DSP
- **SSPs**
    - Read the ID5 ID from ad requests (Prebid.js, tags, server-to-server, etc.)
    - Share the ID5 ID as-received in bid requests to demand partners
    - Enrich all bid requests with an ID5 ID (via server-to-server integration with ID5)
    - Ingest data with an ID5 ID for use with publisher audience building and selling
    - Support PMP creation with segments derived from ID5 IDs
- **Ad Server**
    - Deploy and retrieve the ID5 ID by integrating the ID5 JS API or using the client or server side fetch-end point in any ad server tags or pixels on advertiser pages

- Decrypt the ID5 ID and use it in the same way as their proprietary platform user id, to inform campaign targeting, optimisation, attribution and measurement

Phase 3: Measure

Measure the value of the ID5 ID to your business by working with your platform partners to understand how leveraging ID5 IDs in place of proprietary platform user IDs impacts:

- The size of your addressable audience broken down by browser type
- Overall ROI

Deploying the ID5 ID

Contract

Before getting started with deploying the ID5 ID, we need to make sure you have signed a Site Agreement and have been issued an ID5 Partner Number. If you are not already integrated with ID5, reach out to contact@id5.io or sign the agreement and we'll get you set up right away.

Deployment

There are multiple ways to deploy the ID5 ID across your properties as outlined below. Where possible, we recommend passing additional signals such as hashed email addresses and MAIDS in requests to ID5 to enhance the accuracy of reconciling consented users across domains and devices.

ID5 JS API

If you are not using Prebid.js, you can install the ID5 JS API directly on your page after your CMP (if applicable), but as high in the `<head>` as possible. Full instructions can be found on our GitHub page.

Custom Integrations

The primary benefit of using the Prebid.js User ID Module or ID5 JS API implementations is that they handle all of the steps required to properly integrate beyond just handling the request/response, such as caching, refreshing, and ID storage. If you cannot (or do not want to) use these integration options, you may integrate directly with our platform using the one of the following:

- Client-side Fetch Endpoint | Used to retrieve an ID5 ID for a single user from the user's browser
- Server-side Fetch Endpoint | Used to retrieve an ID5 ID for a single user server-side from an advertiser's server

In-app Deployment

ID5 can be deployed in the mobile in app environment using using one of the custom integrations above. For a complete guide to the specifics of integrating in-app, please see our Mobile In-App Integration page.

Inform your partners

Once the ID5 ID has been deployed across your sites, we recommend you reach out to your ad tech partners to let them know that you have integrated with ID5 and have the ID5 ID deployed. Please contact your ID5 Account Manager if you would like assistance with content for this communication.

For an optimal integration, your:

- **SSP partners** should be retrieving the ID using the Prebid.js Bid Adapter and/or the ID5 JS API and sending them to DSPs via OpenRTB bid requests.
- **Data Platform partners** should be using the ID as a basis for building audience segments and syncing with your SSP and DSP partners.
- **DSP partners** should be decrypting the ID5 IDs they are receiving in OpenRTB bid requests and using it in their bidding logic as well as to communicate identity with their other platform partners including data management platforms.

Read and Use the ID5 ID

Advertisers with multiple domains might want to consider using the ID5 ID to build segments and target campaigns across their properties. To achieve this, the ID5 ID must be decrypted to access the persistent identifier. ID5 encrypts the ID5 ID in order to enforce the privacy preferences of the consumer and the publisher. To learn how to decrypt the ID5 ID, please visit Decrypting the ID5 ID (login required). The decrypted ID5 ID can then be leveraged by your technology providers such as your ad server and data platform to achieve cross domain segment building, campaign optimisation, measurement and attribution. Please contact your Account Manager to guide you through this process.

Privacy & Regulations

Privacy-by-Design

ID5 has built a privacy-by-design shared ID service for publishers and ad tech vendors. Our service leverages the IAB's Transparency and Consent Framework (TCF) and US Privacy Framework to capture the user's privacy preferences.

As a shared ID provider, ID5 acts as a Controller of the ID5 ID, and thus, we must receive a valid legal basis to process requests. When we receive a request for the ID5 ID, we check that we have a legal basis to store our user ID in a cookie before proceeding; if we don't have one, we do not read our cookie or write to it as part of the HTTP response.

When ID5 returns an ID to the page, the value is encrypted in such a way that only platforms that have authorization to process data (based on the consumer's and publisher's privacy preferences) are able to decrypt the string back to a stable ID. By doing so, ID5 enforces privacy preferences and regulations, ensuring that no downstream party can understand the ID without the proper legal basis to do so. When the ID is non-decryptable, the request is truly anonymized, preventing any personal data from being retrieved or processed.

Privacy Policy

For our Platform Privacy Policy, please visit https://id5.io/platform-privacy-policy.

SSP

03/06/2026 10:19 am EST

Core Value Proposition

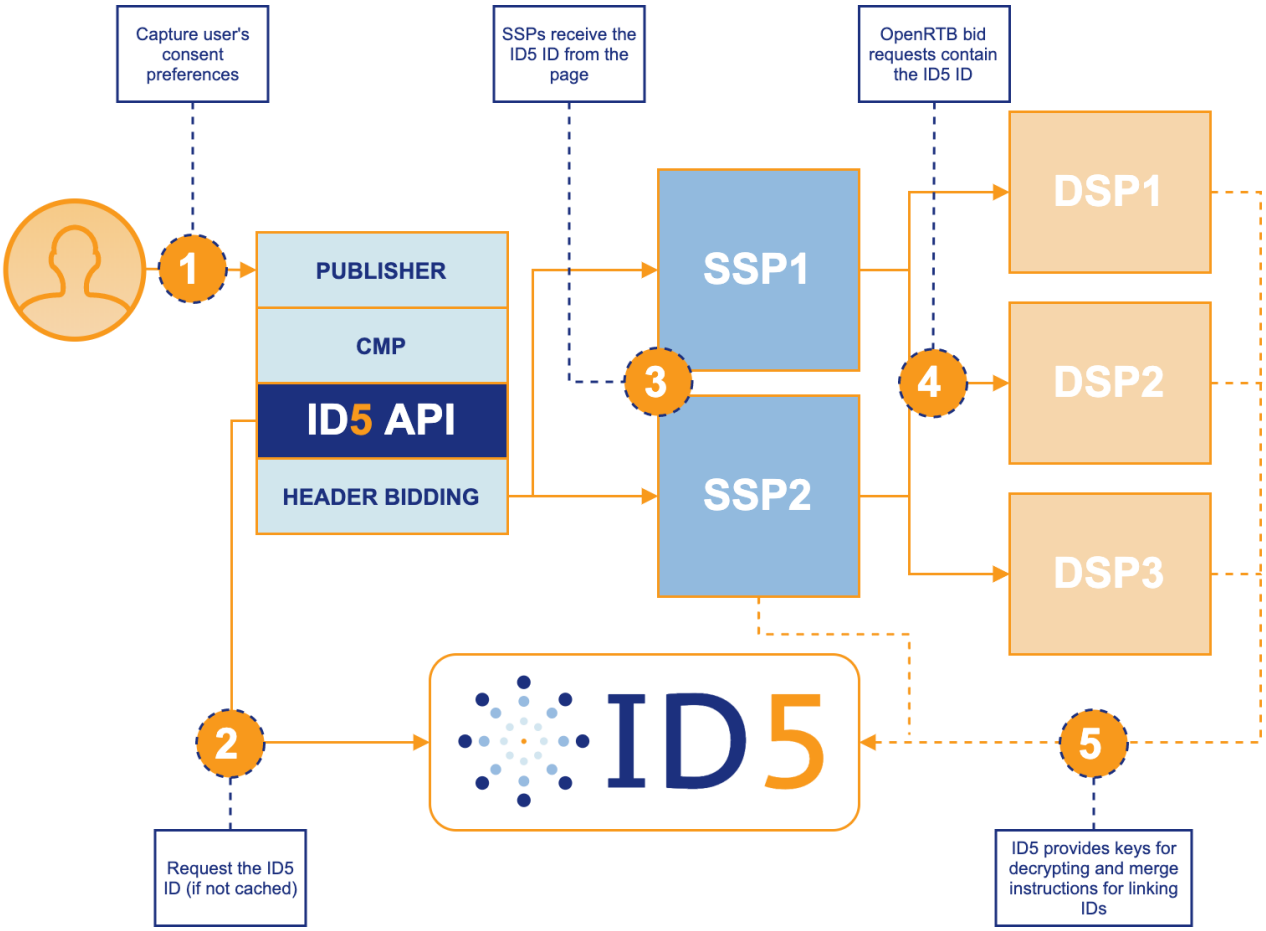
1. Send more identifiable bid requests to buyers to increase CPMs and overall publisher revenue
2. Deliver strategic value to publishers by supporting their transition to the cookieless future
3. Reduce operational costs by outsourcing identity infrastructure to specialized solution provider

ID5 ID Overview

The ID5 ID is a shared, neutral identifier that publishers and ad tech platforms can use to recognise users even in environments where 3rd party cookies are not available or blocked. ID5 enables publishers to create and distribute a shared 1st party identifier to the entire ecosystem. Ad tech platforms that connect with ID5 can decrypt the ID5 ID and improve their user recognition capabilities. The ID5 ID is designed to respect users' privacy choices and publishers preferences throughout the advertising value chain.

Solution Overview

RTB Process



1. The Publisher first loads its CMP and captures the user's consent preferences. This is essential before any IDs or ads are requested or delivered

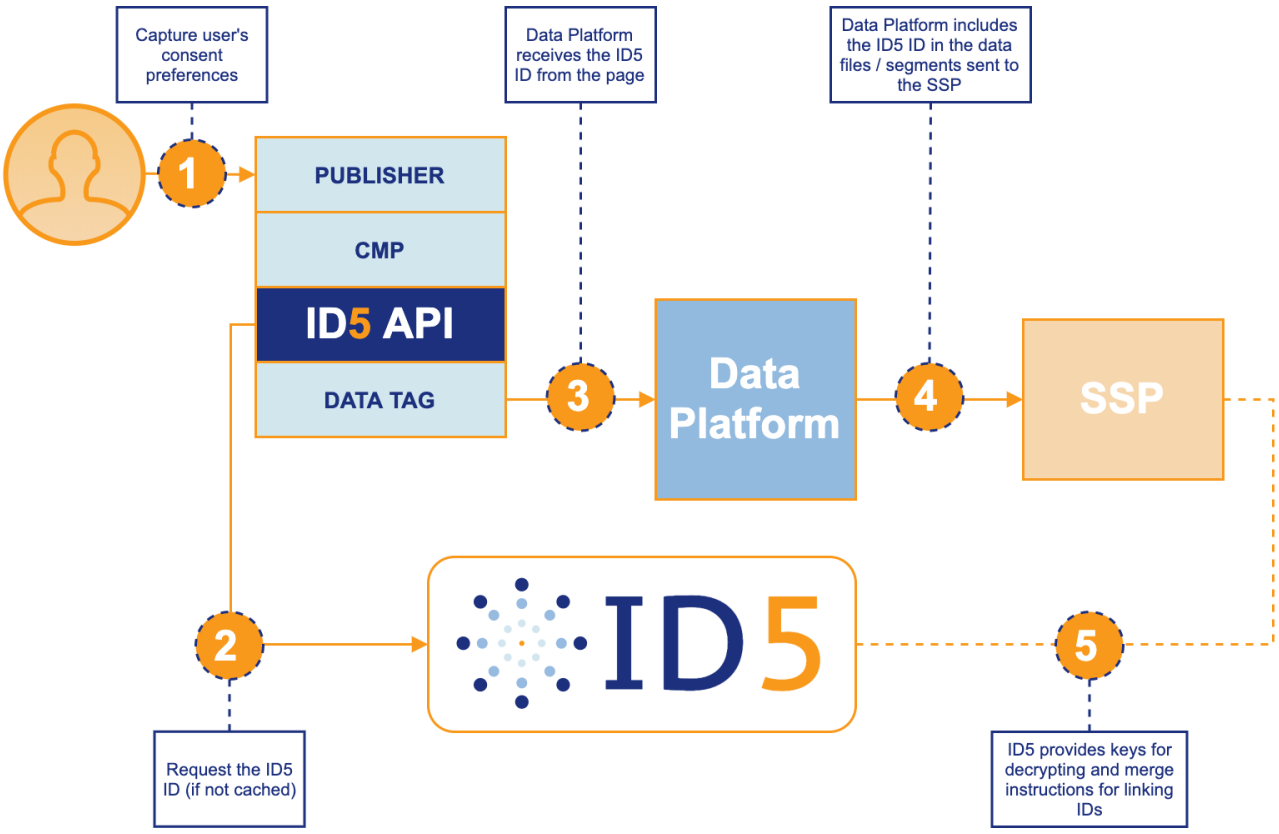
2. The Publisher calls ID5 (via the ID5 API, a header bidding identity module, or server-to-server) to request the ID5 ID, which can then be placed in cache (in the user's browser or the publisher's server) to avoid unnecessary http requests on future page views.

3. Via the publisher's header bidding wrapper, or through a direct integration, SSPs receive the ID5 ID in the ad request to their servers

4. SSPs pass the ID5 ID into the OpenRTB bid request that they send to their DSP partners

5. (Optional) Outside of the RTB process, SSPs may choose to receive keys from ID5 so they can decrypt the value they receive from publishers (only necessary if the SSP is selling audience segments against ID5 IDs).

Data Integrations



1. Publisher first loads its CMP and captures the user's consent preferences. This is essential before any IDs or ads are requested or delivered

2. The Publisher calls ID5 (via the ID5 API or server-to-server) to request the ID5 ID, which can then be placed in cache (in the user's browser or the publisher's server) to avoid unnecessary http requests on future page views.

3. The data platform's tag on the publisher's page retrieves the ID5 ID and passes it, along with any other data signals they normally use, to their servers for processing

4. The data platform pushes the data to the SSP and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass

5. Outside of the data ingestion process, SSPs receive keys from ID5 so they can properly decrypt the value they receive in bid requests. SSPs will also receive merge instructions from ID5 to link publisher-specific IDs together.

ID5 Integration Overview

Phase 1: Read & Share

- Read the ID5 ID from ad requests (Prebid.js, tags, server-to-server, etc.)
- Share the ID5 ID as-received in bid requests to demand partners

Phase 2: Enrich & Use

- Enrich all bid requests with an ID5 ID (via server-to-server integration with ID5)
- Ingest data with an ID5 ID for use with publisher audience building and selling
- Support PMP creation with segments derived from ID5 IDs

ID5 Integration Details

From Prebid.js Bid Adapter

If the ID5 ID is available, the `bidRequest` object, which is available in the bid adapter module, will contain both a `userId` object and a `userIdAsEids` array. The bidder adapter code should listen to this object and pass it along to its server in the ad call.

> We strongly recommend adapters leverage the `bidRequest.userIdAsEids`, rather than retrieving from `bidRequest.userId`, as it contains a pre-formatted version of the eids array needed to pass on to DSPs.

> NOTE: ID5 provides extra data about the user id in the extension object which is important to send to DSPs.

`userIdAsEids` Example

An example implementation in the bid adapter using `bidRequest.userIdAsEids` might look like this:

```
const SUPPORTED_EIDS = [ 'id5-sync.com' ];
if (bidRequest.userIdAsEids) {
    adCallData.eids = bidRequest.userIdAsEids.filter(eid => {
        return SUPPORTED_EIDS.indexOf(eid.source) !== -1;
    });
}
```

`userId` Example

An example implementation in the bid adapter using `bidRequest.userId` might look like this:

```
if (bidRequest.userId && bidRequest.userId.id5id && bidRequest.userId.id5id.uid) {
    adCallData.id5Id = bidRequest.userId.id5id.uid;
    adCallData.id5IdExt = bidRequest.userId.id5id.ext || {};
}
```

Additional information about retrieving the ID5 ID can be found in the Prebid documentation.

From the ID5 JS API

If the publisher has integrated the ID5 API, the SSP can retrieve the ID5 ID directly. This can be used by the SSP in other header bidding integrations or where the SSP has tags directly on page.

```
adCallData.id5Id = id5Status.getUserId();
adCallData.id5IdLinkType = id5Status.getLinkType();
```

For more details about the ID5 API or how to retrieve the ID5 ID, take a look at our documentation on Github.

Sending the ID5 ID to DSPs

OpenRTB Bid Requests

The process of passing the ID5 ID through the OpenRTB bid request depends on the version of the specification adopted by the DSP. The IAB aimed to standardize how Universal IDs, like ID5, are handled by formally incorporating the `eids` structure.

The `eids` array is available within the `user` object. Depending on the OpenRTB version supported by the DSP, it might be included either as part of the `user` object or as an extended field.

OpenRTB 2.5 (user.ext.eids)

```
{
   "user": {
      "id": "SSP_UID",
      "buyeruid": "DSP_UID",
      "ext": {
         "eids": [
            {
               "source": "id5-sync.com",
               "uids": [
                  {
                     "id": "ID5*ecJr2ylVAdiHp2mLG0Mm-LAU6rS-7EW-ySi5jBt1rXz6Y7la7t2k3-",
                     "ext": {
                        "linkType": 2
                     }
                  }
               ]
            }
         ]
      }
   }
}
```

OpenRTB 2.6 (user.eids)

```
{
  "user": {
    "id": "SSP_UID",
    "buyeruid": "DSP_UID",
    "eids": [
      {
        "source": "id5-sync.com",
        "uids": [
          {
            "id": "ID5*ecJr2ylVAdiHp2mLG0Mm-LAU6rS-7EW-ySi5jBt1rXz6Y7la7t2k3-",
            "ext": {
              "linkType": 2
            }
          }
        ]
      }
    ]
  }
}
```

ATTENTION
It's recommended to double-check the DSP's integration specs to confirm the location of the eids array inside the user object.

ID5 provides extra data about the user ID in the `ext` object. For more information, see our metadata documentation.

ID5 ID Decryption

ID5 encrypts the ID in order to enforce the privacy preferences of the consumer and the publisher. To learn more about how to decrypt the ID, please visit Decrypting the ID5 ID (login required).

ID5 Partner Creation

Before getting started with the using ID5 ID (Integration Phase 2), we need to make sure you have an ID5 Partner account. If you are not already integrated with ID5, reach out to contact@id5.io and we'll get you set up right away.

Privacy & Regulations

Privacy-by-Design

ID5 has built a privacy-by-design shared ID service for publishers and ad tech vendors. Our service leverages the IAB's Transparency and Consent Framework (TCF) and US Privacy Framework to capture the user's privacy preferences.

As a shared ID provider, ID5 acts as a Controller of the ID5 ID, and thus, we must receive a valid legal basis to process requests. When we receive a request for the ID5 ID, we check that we have a legal basis to store our user ID in a cookie before proceeding; if we don't have one, we do not read our cookie or write to it as

part of the HTTP response.

When ID5 returns an ID to the page, the value is encrypted in such a way that only platforms that have authorization to process data (based on the consumer's and publisher's privacy preferences) are able to decrypt the string back to a stable ID. By doing so, ID5 enforces privacy preferences and regulations, ensuring that no downstream party can understand the ID without the proper legal basis to do so. When the ID is non-decryptable, the request is truly anonymized, preventing any personal data from being retrieved or processed.

Privacy Policy

For our Platform Privacy Policy, please visit https://id5.io/platform-privacy-policy.

DSP

11/25/2025 8:24 am EST

Core Value Proposition

1. Bid on more impressions, track conversion attribution to ultimately improve bidding strategies and campaign performance in all digital advertising environments with the ID5 ID
2. Enable clients to reach and engage with more consumers effectively across devices and to generate better ROI from their campaigns with the Cross-device Graph
3. Reduce privacy compliance risks and infrastructure costs by outsourcing identity to a specialized solution provider

ID5 ID Overview

The ID5 ID is a shared, neutral identifier that publishers and ad tech platforms can use to recognise users even in environments where 3rd party cookies are not available or blocked. ID5 enables publishers to create and distribute a shared 1st party identifier to the entire ecosystem. Ad tech platforms that connect with ID5 can decrypt the ID5 ID and improve their user recognition capabilities. The ID5 ID is designed to respect users' privacy choices and publishers preferences throughout the advertising value chain.

Solution Overview

RTB Process

1. The Publisher first loads its CMP and captures the user’s consent preferences. This is essential before any IDs or ads are requested or delivered.
2. The Publisher calls ID5 (via the ID5 API, a header bidding identity module, or server-to-server) to request the ID5 ID, which can then be placed in cache (in the user’s browser or the publisher’s server) to avoid unnecessary http requests on future page views.
3. Via the publisher’s header bidding wrapper, or through a direct integration, SSPs receive the ID5 ID in the ad request to their servers.
4. SSPs pass the ID5 ID into the OpenRTB bid request that they send to their DSP partners.
5. Outside of the RTB process, DSPs receive keys from ID5 so they can properly decrypt the value they receive in bid requests. DSPs will also receive merge instructions from ID5 to link publisher-specific IDs together.

Data Integrations

1. Publisher first loads its CMP and captures the user's consent preferences. This is essential before any IDs or ads are requested or delivered.

2. The Publisher calls ID5 (via the ID5 API or server-to-server) to request the ID5 ID, which can then be placed in cache (in the user's browser or the publisher's server) to avoid unnecessary http requests on future page views.

3. The data platform's tag on the publisher's page retrieves the ID5 ID and passes it, along with any other data signals they normally use, to their servers for processing

4. The data platform pushes the data to the DSP and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass.

5. Outside of the data ingestion process, DSPs receive keys from ID5 so they can properly decrypt the value they receive in bid requests. DSPs will also receive merge instructions from ID5 to link publisher-specific IDs together.

ID5 Integration Overview

Phase 1: Listen & Log

- Listen to the ID5 ID in the bidstream
- Log IDs for analytics purposes
- Measure potential uplift v. proprietary IDs, broken down by browser type (Chrome v. Safari/Firefox), countries, devices, etc.

Phase 2: Read & Use

- Add ID5 ID to ID graph and implement real-time decryption
- Enable decisioning on ID5 ID in bidding engine
- Use ID5 ID to optimize campaign delivery (frequency capping, reach optimization, sequencing)

Phase 3: Enrich & Measure

- Ingest user profile from data providers keyed off ID5 ID
- Integrate calls to ID5 API in data collection & conversion tags
- Use ID5 ID for cookie-less audience (re)targeting, performance measurement & attribution, and CPA-based bid optimization

ID5 Integration Details

ID5 Partner Creation

Before getting started with the ID5 ID, we need to make sure you have an ID5 Partner account. If you are not already integrated with ID5, reach out to contact@id5.io and we'll get you set up right away.

Receiving the ID5 ID from SSPs

SSPs include the ID5 ID in the OpenRTB bid request for DSPs to log, look up users, and use in bidding. The

IAB aimed to standardize how Universal IDs, like ID5, are handled by formally incorporating the `eids` structure.

The `eids` array is available within the `user` object. Depending on the OpenRTB version supported, it might be included either as part of the `user` object or as an extended field.

OpenRTB 2.5 (user.ext.eids)

```
{
  "user": {
    "id": "SSP_UID",
    "buyeruid": "DSP_UID",
    "ext": {
      "eids": [
        {
          "source": "id5-sync.com",
          "uids": [
            {
              "id": "ID5*ecJr2ylVAdiHp2mLG0Mm-LAU6rS-7EW-ySi5jBt1rXz6Y7la7t2k3-",
              "ext": {
                "linkType": 2
              }
            }
          ]
        }
      ]
    }
  }
}
```

OpenRTB 2.6 (user.eids)

```
{
  "user": {
    "id": "SSP_UID",
    "buyeruid": "DSP_UID",
    "eids": [
      {
        "source": "id5-sync.com",
        "uids": [
          {
            "id": "ID5*ecJr2ylVAdiHp2mLG0Mm-LAU6rS-7EW-ySi5jBt1rXz6Y7la7t2k3-",
            "ext": {
              "linkType": 2
            }
          }
        ]
      }
    ]
  }
}
```

ID5 ID Decryption

ID5 encrypts the ID in order to enforce the privacy preferences of the consumer and the publisher. To

learn more about how to decrypt the ID, please visit Decrypting the ID5 ID (login required).

Privacy & Regulations

Privacy-by-Design

ID5 has built a privacy-by-design shared ID service for publishers and ad tech vendors. Our service leverages the IAB's Transparency and Consent Framework (TCF) and US Privacy Framework to capture the user's privacy preferences.

As a shared ID provider, ID5 acts as a Controller of the ID5 ID, and thus, we must receive a valid legal basis to process requests. When we receive a request for the ID5 ID, we check that we have a legal basis to store our user ID in a cookie before proceeding; if we don't have one, we do not read our cookie or write to it as part of the HTTP response.

When ID5 returns an ID to the page, the value is encrypted in such a way that only platforms that have authorization to process data (based on the consumer's and publisher's privacy preferences) are able to decrypt the string back to a stable ID. By doing so, ID5 enforces privacy preferences and regulations, ensuring that no downstream party can understand the ID without the proper legal basis to do so. When the ID is non-decryptable, the request is truly anonymized, preventing any personal data from being retrieved or processed.

Privacy Policy

For our Platform Privacy Policy, please visit https://id5.io/platform-privacy-policy.

Data Platform

01/09/2025 10:49 am EST

Core Value Proposition

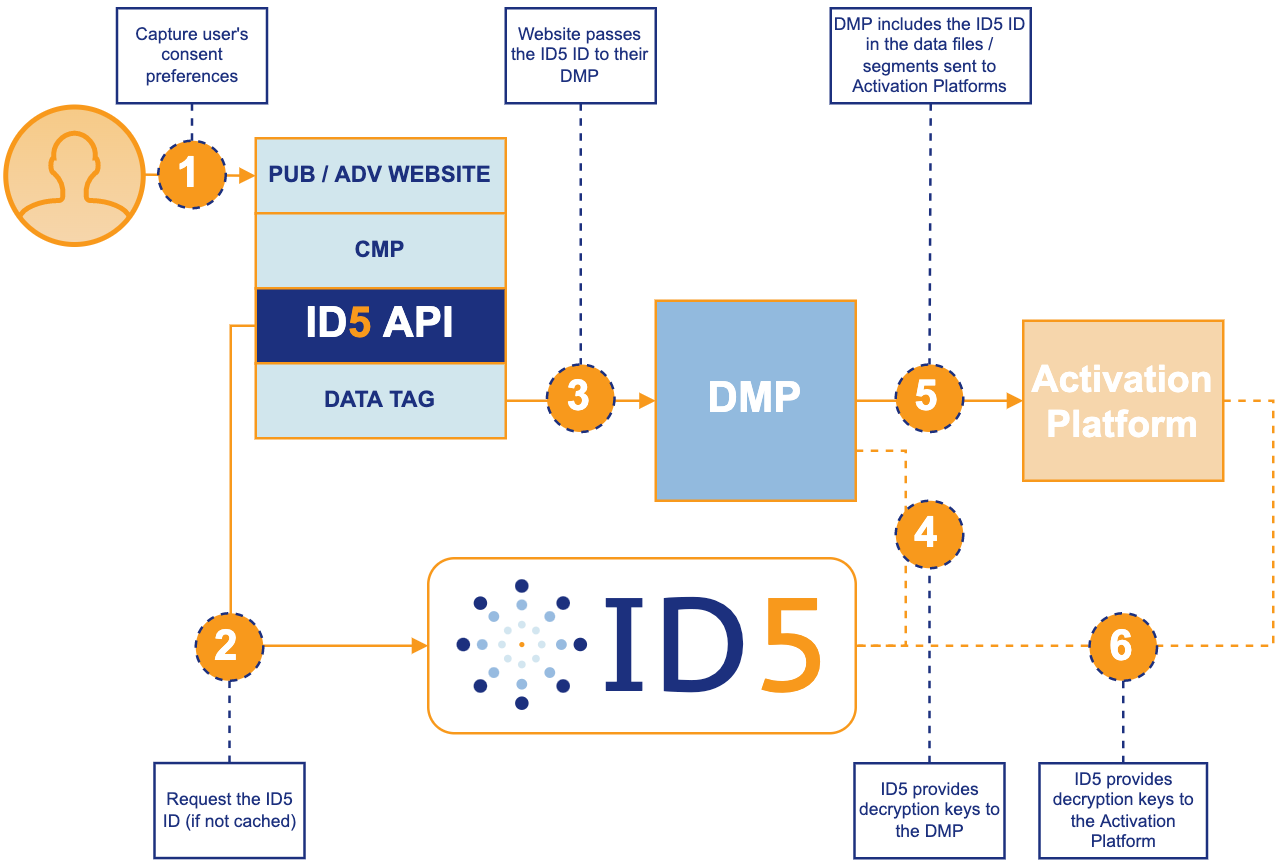
1. Improve user identification with the Partner Graph and maximize the availability of first and third-party data in activation platforms such as Ad Servers, SSPs and DSPs
2. Future-proof your business with the ID5 ID by enabling clients to activate audiences in cookieless environments
3. Improve efficiency and privacy compliance by outsourcing identity infrastructure to a specialized solution provider

ID5 ID Overview

The ID5 ID is a shared, neutral identifier that publishers and ad tech platforms can use to recognise users even in environments where 3rd party cookies are not available or blocked. ID5 enables publishers to create and distribute a shared 1st party identifier to the entire ecosystem. Ad tech platforms that connect with ID5 can decrypt the ID5 ID and improve their user recognition capabilities. The ID5 ID is designed to respect users' privacy choices and publishers preferences throughout the advertising value chain.

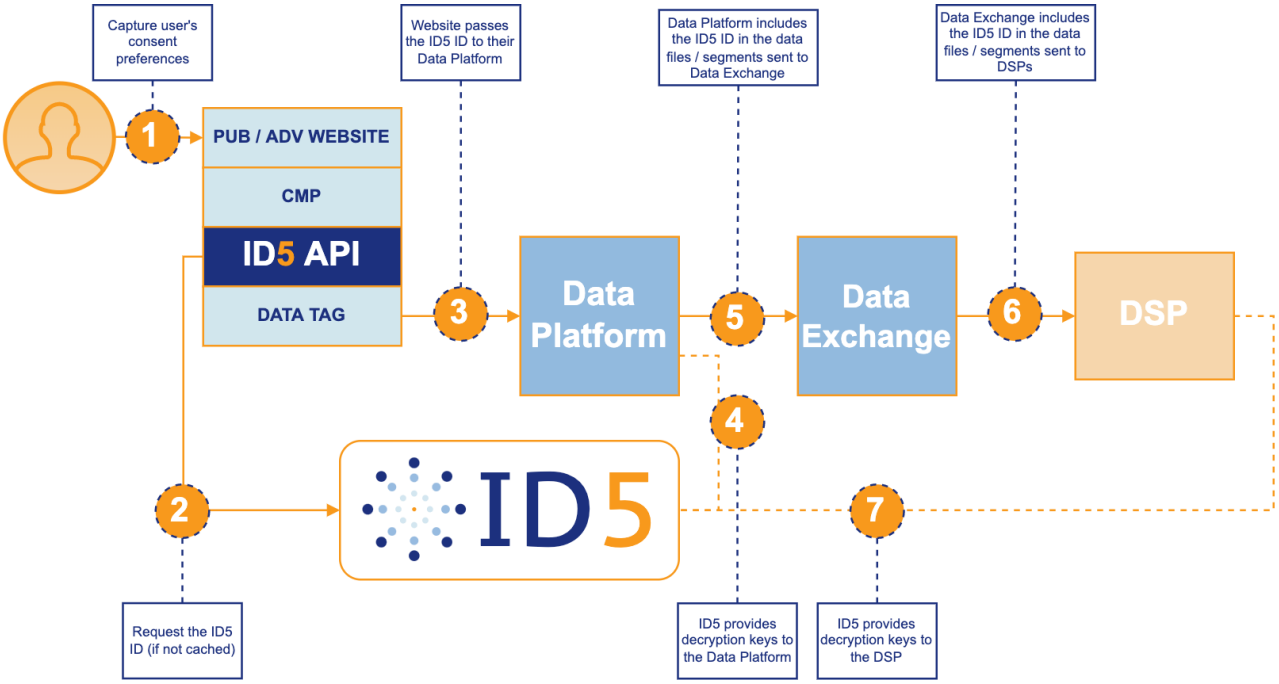
Solution Overview

Data Provider to DMP Integration



1. Publisher first loads its CMP and captures the user's consent preferences. This is essential before any IDs or ads are requested or delivered
2. The ID5 API checks in cache (local storage, 1P or 3P cookies) for an ID5 ID and ensures it is still fresh. If necessary, a request to ID5 is made for a new/refreshed ID, which is then placed in cache to avoid unnecessary http requests on future page views.
3. The DMP tag on the publisher's page retrieves the ID5 ID from the API and passes it, along with any other data signals they normally send, to the DMP servers for processing
4. In order to aggregate data and build segments, the DMP must decrypt the ID5 ID with keys provided by ID5, which happens outside of the data ingestion process.
5. The DMP pushes the generated data segments to Activation Platforms (Data Exchanges, DSPs, SSPs, etc) and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass
6. Outside of the data ingestion process, the Activation Platform decrypts the ID5 ID and matches the ID5 ID to their internal ID

Data Exchange Integration



1. Publisher first loads its CMP and captures the user's consent preferences. This is essential before any IDs or ads are requested or delivered
2. The ID5 API checks in cache (local storage, 1P or 3P cookies) for an ID5 ID and ensures it is still fresh. If necessary, a request to ID5 is made for a new/refreshed ID, which is then placed in cache to avoid unnecessary http requests on future page views.

3. The DMP tag on the publisher's page retrieves the ID5 ID and passes it, along with any other data signals they normally use, to its servers for processing

4. In order to aggregate data and build segments, the DMP must decrypt the ID5 ID with keys provided by ID5, which happens outside of the data ingestion process.

5. The DMP pushes the generated data segments to Data Exchange and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass

6. The Data Exchange pushes the DMP's data segments to DSPs and includes the ID5 ID in addition to, or instead of, the normal user IDs they pass

7. Outside of the data ingestion process, DSPs decrypt the ID5 ID and match the ID5 ID to their internal ID

ID5 Integration Overview

Phase 1: Collect & Log

- Integrate calls to ID5 API in data collection tags
- Listen to the ID5 ID in calls to your platform
- Log IDs for analytics purposes

Phase 2: Read & Measure

- If applicable, pass through the still-encrypted ID5 ID in raw data feeds to other platforms
- Measure potential uplift v. proprietary IDs, broken down by browser type (Chrome v. Safari/Firefox), countries, devices, etc.

Phase 3: Use & Share

- Add ID5 ID to ID graph and implement real-time decryption
- Enable analytics and audience building with the ID5 ID
- Use ID5 ID (in addition to existing IDs) when sending data segments to other platfor

Integration Details

ID5 Partner Creation

Before getting started with the ID5 ID, we need to make sure you have an ID5 Partner account. If you are not already integrated with ID5, reach out to contact@id5.io and we'll get you set up right away.

Passing or Receiving the ID5 ID in Data Integrations

Data integrations between platforms are custom with no industry standards to go by. As such, you will need to work with your inbound or outbound data partner to determine the best way to receive or pass the ID5 ID with them. However, we've experienced some of these integrations and have the following tips and best practices to share.

If you'd like to hear more about how other platforms have addressed this or would like to seek advice, reach out to ID5 at contact@id5.io and we'd be more than happy to provide you some best practices and

consulting to help make your integration work.

JSON Object

If data is passed using JSON, rather than just pass a field called “id” (which could contain the inbound or outbound partner's UID), we recommend using (or creating) a user ids array that can contain multiple ids: the receiving partner's, the sending partner's, and shared IDs like ID5. An example may look like this:

```
{
  "user_ids": [
    {
      "source": "RECEIVING_PARTNER",
      "uids": [
        {
          "id": "RECEIVING_PARTNER_UID"
        }
      ]
    },
    {
      "source": "id5-sync.com",
      "uids": [
        {
          "id": "ID5-abc123"
        }
      ]
    }
  ],
  ... // additional data fields
}
```

Flat Files

If data is passed using flat files, typically it is agreed ahead of time whether to use the receiving partner’s or the outbound partner’s UID for all rows of data received, along with a file format for passing data. We suggest adding a field in the file format to accept a shared ID like ID5, in addition to, or instead of, the existing UID being used.

ID5 ID Decryption

ID5 encrypts the ID in order to enforce the privacy preferences of the consumer and the publisher. To learn more about how to decrypt the ID, please visit Decrypting the ID5 ID (login required).

Privacy & Regulations

Privacy-by-Design

ID5 has built a privacy-by-design shared ID service for publishers and ad tech vendors. Our service leverages the IAB’s Transparency and Consent Framework (TCF) and US Privacy Framework to capture the user’s privacy preferences.

As a shared ID provider, ID5 acts as a Controller of the ID5 ID, and thus, we must receive a valid legal basis to process requests. When we receive a request for the ID5 ID, we check that we have a legal basis to store

our user ID in a cookie before proceeding; if we don't have one, we do not read our cookie or write to it as part of the HTTP response.

When ID5 returns an ID to the page, the value is encrypted in such a way that only platforms that have authorization to process data (based on the consumer's and publisher's privacy preferences) are able to decrypt the string back to a stable ID. By doing so, ID5 enforces privacy preferences and regulations, ensuring that no downstream party can understand the ID without the proper legal basis to do so. When the ID is non-decryptable, the request is truly anonymized, preventing any personal data from being retrieved or processed.

Privacy Policy

For our Platform Privacy Policy, please visit https://id5.io/platform-privacy-policy.

ID5 Javascript Library

02/05/2026 9:08 am EST

ID5 Javascript API Workflow

Here's a step-by-step guide to integrating ID5 effectively into your webpage. After configuring your ID5 JS API to retrieve the ID5 ID, you can share it with partners on your page using a single JavaScript variable. Partners can then transmit the ID5 ID to their platforms via existing tags or pixels to communicate user identity.

The ID5 ID serves as a key to build and activate audiences, optimize campaigns, and measure performance, even in cookie-less environments. This helps publishers future-proof their user addressability and sustain advertising revenue streams.

1. Register with ID5

The ID5 ID is free to distribute, but requires a simple registration with us. If you don't already have an account with ID5, please visit our website to sign up and request your ID5 Partner Number.

NOTICE: TERMS OF SERVICE & BINDING AGREEMENT

ID5 provides this software and the ID5 ID Service free of charge, subject to strict data usage and privacy rules.

By copying, installing, or initializing the ID5 API (or any ID5 Code Snippet), you and the legal entity you represent AGREE TO BE BOUND by the ID5 ID Agreement.

If you do not agree to the ID5 ID Agreement, you are NOT AUTHORIZED to use this code or the ID5 Service.

Commercial Use: If you are a Platform, Intermediary, or intend to use ID5 Data for commercial purposes beyond the standard publisher integration, you must contact contact@id5.io to execute a commercial license.

2. Integrate the ID5 JS API

We recommend that you monitor our releases in order to stay up-to-date with any changes to the library.

The ID5 JS API is an open-source library, available on GitHub: https://github.com/id5io/id5-api.js. All documentation for building and installing the library will be maintained in GitHub, but please reach out to support@id5.io if you have any questions or need help.

3. Optimize ID5 JS API Configuration

Configuring the pd parameter

To maximise addressability and produce the highest quality ID5 ID, publishers and advertisers must send additional signals such as Hashed Email, First Party user IDs in the Partner Data (pd) parameter when available. To ensure this information is shared in a secure way, please review the guidance here.

4. Integrate TrueLink (Recommended)

TrueLink is an additional client-side integration method that can complement a standard ID5 JS API or Prebid (version 9.2.0 and above) implementation. With a TrueLink integration, ID5 can create a cross-domain signal called the TrueLink ID for a single user within a specific browser. The TrueLink ID is produced independently of third party cookies by redirecting the user through an ID5 operated domain and setting the ID as a first party cookie.

This TrueLink signal is utilized by ID5 for cross domain user reconciliation and the generation of a high-quality ID5 ID. Publishers can optionally access a publisher first-party user identifier called the ID5 Guarded Publisher ID (GPID). The GPID is a publisher specific version of the ID5 ID which remains unique for a user across their owned and operated properties within a given browser environment. It's quality is implicitly linked to the provision of signals such as hashed emails and TrueLink signals. The GPID can be used to facilitate use cases such as cross-domain audience building and activation, all without relying on third-party cookies. It may also be used as a PPID within Google Ad Manager.

To integrate TrueLink, follow the instructions here.

5. Use ID5's GPID as a first party identifier or as a PPID (Optional)

ID5 can optionally provide publisher and advertisers with access to a partner-specific version of the ID5 ID, known as the Guarded Publisher ID (GPID). The GPID can be used for various publisher or advertiser purposes, including cross-domain audience building, audience activation, measurement, and attribution. The GPID can also be used as a Publisher Provided Identifier (PPID) in ad servers like Google Ad Manager or AppNexus. Google's PPID, for example, enables publishers to:

- Apply audience-based ad delivery controls, such as frequency capping and sequential ad rotation.
- Enhance audience segmentation and targeting across devices.
- Potentially increase CPMs from Google's buying stack.

You can find out more about the GPID and how to get it enabled here.

ID5 User ID Module for Prebid.js

07/08/2026 7:09 am EDT

ID5 Prebid Integration Workflow

Here's a step-by-step overview of integrating ID5 into your Prebid configuration for any webpage. Once your Prebid.js setup is configured to fetch the ID5 ID, your demand partners in Prebid can access this ID and pass it to their server-side RTB partners (typically DSPs) as a means to communicate user identity. This enables DSPs to recognize and target users, manage frequency and recency caps, and apply additional data, even in environments where cookies are unsupported. This in turn, helps publishers futureproof user addressability and generate sustainable advertising revenue.

1. Register with ID5

The ID5 ID is free to use, but requires a simple registration with us. If you don't already have an account with ID5, please visit our website to sign up and request your ID5 Partner Number.

NOTICE: TERMS OF SERVICE & BINDING AGREEMENT

ID5 provides this software and the ID5 ID Service free of charge, subject to strict data usage and privacy rules.

By copying, installing, or initializing the ID5 API (or any ID5 Code Snippet), you and the legal entity you represent AGREE TO BE BOUND by the ID5 ID Agreement.

If you do not agree to the ID5 ID Agreement, you are NOT AUTHORIZED to use this code or the ID5 Service.

Commercial Use: If you are a Platform, Intermediary, or intend to use ID5 Data for commercial purposes beyond the standard publisher integration, you must contact contact@id5.io to execute a commercial license.

2. Build Prebid.js with the User ID Module

Below are step by step instructions for installing and configuring the Prebid.js User ID Module with the ID5 ID. The instructions below assume a basic understanding of building Prebid.js and editing its page-level configuration; for more detailed instructions, getting started guides, and more, please visit the Prebid.org website.

When building Prebid.js, include both the `userId`, `id5IdSystem`, `id5Id5AnalyticsAdapter` modules, in addition to the other modules you normally include.

```
gulp build –modules=userId,id5IdSystem,id5AnalyticsAdapter
```

You may also use the Prebid Download page to build your version of Prebid.js by selecting the **User ID Module: ID5 ID, Analytics Adapter: ID5**.

We recommend that you monitor Prebid Releases in order to stay up-to-date with any changes to the implementation of the ID5 ID in Prebid.

3. Configure the User ID Module

ATTENTION
If you or your monetization partners are deploying multiple Prebid wrappers on your websites, you should make sure you add the ID5 ID User ID module to every wrapper. Only the bidders configured in the Prebid wrapper where the ID5 ID User ID module is installed and configured will be able to pick up the ID5 ID. Bidders from other Prebid instances will not be able to pick up the ID5 ID.

Within the `pbjs.setConfig()` function, add the following configuration before making a request for bids:

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: 'id5Id',
      params: {
        partner: 173,           // change to the Partner Number you received from ID5
        externalModuleUrl: 'https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js',
        pd: 'MTIiNTBjY...'      // see below for a link to how to generate the pd string
        abTesting: {            // required if using ID5 Analytics Adapter
          enabled: true,        // false by default
          controlGroupPct: 0.1  // valid values are 0.0 - 1.0 (inclusive)
        }, canCookieSync: true, // optional, has effect only when externalModuleUrl is used  gamTargetingPrefix: "id5" // optional, when set the ID5 module will set gam targeting paramaters with this prefix
      },
      storage: {
        type: 'html5',          // 'html5' is the required storage type
        name: 'id5id',          // 'id5id' is the required storage name
        expires: 90,            // storage lasts for 90 days
        refreshInSeconds: 7200  // refresh ID every 2 hours to ensure it's fresh
      }
    }],
    auctionDelay: 250           // 250ms maximum auction delay, applies to all userId modules
  }
});
```

From **Prebid.js v4.13.0**, ID5 requires `storage.type` to be `"html5"` and `storage.name` to be `"id5id"`. Using other values will display a warning today, but in an upcoming release, it may prevent the ID5 module from loading. This change is to ensure the ID5 module in Prebid.js interoperates properly with the ID5 API and to reduce the size of publishers' first-party cookies that are sent to their web servers.

Configuration Parameters

Name	Required	Type	Description	Example
partner	Required	Number	This is the ID5 Partner Number obtained from registering with ID5.	173

Name	Required	Type	Description	Example
externalModuleUrl	Optional	String	URL to the external ID5 module. Highly recommended for the best integration possible.	`https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js`
pd	Optional	String	Publisher-supplied data, such as hashed email address or publisher user id, used for linking ID5 IDs across domains. Omit the parameter or leave as an empty string if no data to supply. Learn how to populate this field.	"MT1iNTBjY..."
abTesting	Required if using ID5's Analytics Adapter	Object	Allows publishers to easily run an A/B Test. If enabled and the user is in the Control Group, the ID5 ID will NOT be exposed to bid adapters for that user. See below for more details.	Disabled by default
abTesting.enabled	Required if using ID5's Analytics Adapter	Boolean	Set this to true to turn on this feature	true

Name	Required	Type	Description	Example
abTesting.controlGroupPct	Required if using ID5's Analytics Adapter	Number	Must be a number between 0.0 and 1.0 (inclusive) and is used to determine the percentage of users that fall into the control group (and thus not exposing the ID5 ID). For example, a value of 0.20 will result in 20% of users without an ID5 ID and 80% with an ID.	0.1
disableExtensions	Optional	Boolean	Set this to true to force turn off extensions call. Default false	true
provider	Optional	String	An identifier provided by ID5 to technology partners who manage API deployments on behalf of their clients. Reach out to ID5 if you have questions about this parameter	providerName
canCookieSync	Optional	Boolean	Set this to true to enable cookie syncing with other ID5 partners. See our documentation for details. Default false	true
gamTargetingPrefix	Optional	String	When this parameter is set the ID5 module will set appropriate GAM pubads targeting tags	"id5"

From **Prebid.js v8.33.0** you should provide the `externalModuleUrl` parameter and set it to the latest available module version at `https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js` to ensure you are benefiting from the latest version of ID5's identity resolution alogrithms and functionality on page. While the configuration of the parameter is optional and not technically required we **strongly**

recommend implementing it.
If you have any questions, please reach out to us at prebid@id5.io.

Configuring the pd parameter

To maximise addressability and produce the highest quality ID5 ID, publishers and advertisers must send additional signals such as Hashed Email, First Party user IDs in the Partner Data (pd) parameter when available. To ensure this information is shared in a secure way, please review the guidance here.

The below is an example for how to Passing Signals to ID5 demonstrating how your configuration could look in Prebid:

```
params: {
   partner: 173,   // change to the Partner Number you received from ID5
   pd: 'MTIiNTBjYTA4MjcxNzk1YThlN2U0MDEyODEzZjIzZDUwNTE5M2Q3NWMwZjJlMmJiOTlhYWE2M2FhODIyZjY2ZWQzIjU9bSV
},
```

Configuring the User ID module for A/B Testing (Optional)

Publishers may want to test the value of the ID5 ID with their downstream partners. While there are various ways to do this, A/B testing is a standard approach. Instead of publishers manually enabling or disabling the ID5 User ID Module based on their control group settings (which leads to fewer calls to ID5, reducing our ability to recognize the user), we have baked this into our module directly.

To turn on A/B Testing, simply edit the configuration (see details above) to enable it and set what percentage of users you would like to set for the control group. The control group is the set of users where an ID5 ID will not be exposed to bid adapters or in the various user id functions available on the pbjs global. A common control group percentage used is 5%.

An additional value of `ext.abTestingControlGroup` will be set to `true` or `false` and can be used to inform reporting systems that the user was in the control group or not. It's important to note that the control group is user-based, and not request-based. In other words, from one page view to another, a user will always be in or out of the control group without changing.

The `abTesting` feature is available in version `4.20.0` of Prebid.js and later.

Google Ad Manager (GAM) Targeting Tag Support in ID5 UserId Module (Since v10.6.0)

Starting from version **10.6.0**, the **ID5 UserId module** supports setting **Google Ad Manager (GAM) targeting keys**, provided that the feature is enabled via the `params.gamTargetingPrefix` parameter.

Key Configuration Requirements

- The value of `params.gamTargetingPrefix` must be unique per property.
- If multiple Prebid integrations are present on a page, each must use a distinct prefix.
- If the same prefix is reused across integrations, the GAM targeting tags will be overwritten by the last initialized ID5 module instance.

Targeting Tags Set by the Module

When a non-empty `params.gamTargetingPrefix` is configured **and** the ID5 module has successfully initialized, the module sets the following GAM targeting keys:

Key	Description
`{prefix}_id`	Set to `"y"` if a valid `id5id` is available. The key is **not set** if no ID is present.
`{prefix}_ab`	Set when A/B testing is enabled: `"n"` – Normal group (ID5 ID returned) `"c"` – Control group (no ID5 ID returned)
`{prefix}_enrich`	Set when **bid enrichment** is enabled: `"y"` – Enriched IDs returned `"s"` – Enrichment attempted, no enriched IDs found `"c"` – Control group (no enrichment, used for uplift measurement)

GDPR Support

The ID5 ID is a privacy-by-design implementation of a shared ID and fully supports the GDPR. When the ID5 ID is requested by Prebid in a GDPR-relevant country, ID5 will ensure the user has consented to processing by ID5 for the "Information storage and access" purpose (Purpose 1). If not, ID5 will not attempt to read or write our 3P cookie and we will not deliver an ID.

To enable GDPR support within Prebid, you will need to include the GDPR Consent Management module when you build Prebid:

```
gulp build –modules=userId,id5IdSystem,id5AnalyticsAdapter,consentManagement
```

You will also need to ensure you add the appropriate configuration to your `setConfig()` function to include `consentManagement`:

```
// JavaScript
pbjs.setConfig({
 userSync: {
  userIds: [{
   name: 'id5Id',
   params: {
    partner: 173,           // change to the Partner Number you received from ID5
    externalModuleUrl: 'https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js',
    pd: 'MTIiNTBjY...'      // optional, see below for a link to how to generate the pd string
    abTesting: {            // required if using ID5 Analytics Adapter
     enabled: true,         // false by default
     controlGroupPct: 0.1   // valid values are 0.0 - 1.0 (inclusive)
    }, canCookieSync: true // optional, has effect only when externalModuleUrl is used
   },
   storage: {
    type: 'html5',          // 'html5' is the required storage type
    name: 'id5id',          // 'id5id' is the required storage name
    expires: 90,            // storage lasts for 90 days
    refreshInSeconds: 7200  // refresh ID every 2 hours to ensure it's fresh
   }
  }],
  auctionDelay: 250         // 250ms maximum auction delay, applies to all userId modules
 }
});
```

ID5 User ID Module Response

After configuring your Prebid.js setup to pull the ID5 ID, your demand partners configured in Prebid can retrieve the ID and pass it on to their server side RTB partners (typically DSPs). This allows the DSP to target the user, manage frequency and recency capping, and apply additional data using the ID5 ID.

The ID5 User ID module provides following eids:

```
[
  {
    "source": "id5-sync.com",
    "uids": [
      {
        "id": "some-random-id-value",
        "atype": 1,
        "ext": {
          "linkType": 2,
          "abTestingControlGroup": false
        }
      }
    ]
  },
  {
    "source": "gpid.id5-sync.com",
    "uids": [
      {
        "id": "some-gpid",
        "atype": 1
      }
    ]
  },
  {
    "source": "euid.eu",
    "uids": [
      {
        "id": "some-euid-id",
        "atype": 3,
        "ext": {
          "provider": "id5-sync.com"
        }
      }
    ]
  }
]
```

The ID from `id5-sync.com` should be always present (though the id provided will be '0' in case of no consent or optout).

The ID from `euid.eu` will be available if the partner that is used in ID5 user module has the EUID2 integration enabled (it has to be enabled on the ID5 side).

The id from `gpid.id5-sync.com` will be available if the publisher has enabled Guarded Publisher ID (if you are an ID5 partner you can learn more at here).

ID5 ID Encryption

The ID5 ID delivered to Prebid is encrypted by ID5 with a rotating key to avoid unauthorized usage and to enforce privacy requirements. Only platforms that have the necessary rights to process user data will be able to decrypt the ID and use it for targeting, frequency capping, measurement, etc. Therefore, we strongly recommend setting

`storage.refreshInSeconds` to 2 hours (`7200` seconds) or less to ensure all demand partners receive an ID that has been encrypted with the latest key, has up-to-date privacy signals, and allows them to transact against it.

4. Configure the ID5 Prebid Analytics Module for Identity Insights

> Identity Insights is enabled through ID5's Analytics Adapter for Prebid, available for Prebid.js version 7.27.0 or higher

To help publishers better understand the value of working with ID5, we have launched Identity Insights - an ID5 analytics adapter for Prebid. With just a few additional lines of configuration, ID5 is able to capture prebid event data to provide publishers with additional insights about the value of the ID5 ID to their business. For now, insights will be provided in the form of an adhoc report shared by the ID5 publisher support team. Identity Insights is currently free for publishers to use.

To enable ID5 Identity Insights within Prebid, you will need to include the ID5 analytics adapter when you build Prebid:

```
gulp build –modules=userId,id5IdSystem,id5AnalyticsAdapter
```

In addition to including the ID5 Analytics Adapter in your prebid build, you will also need to ensure you enable analytics (this is in addition to the configuration described above to enable the ID5 User ID module):

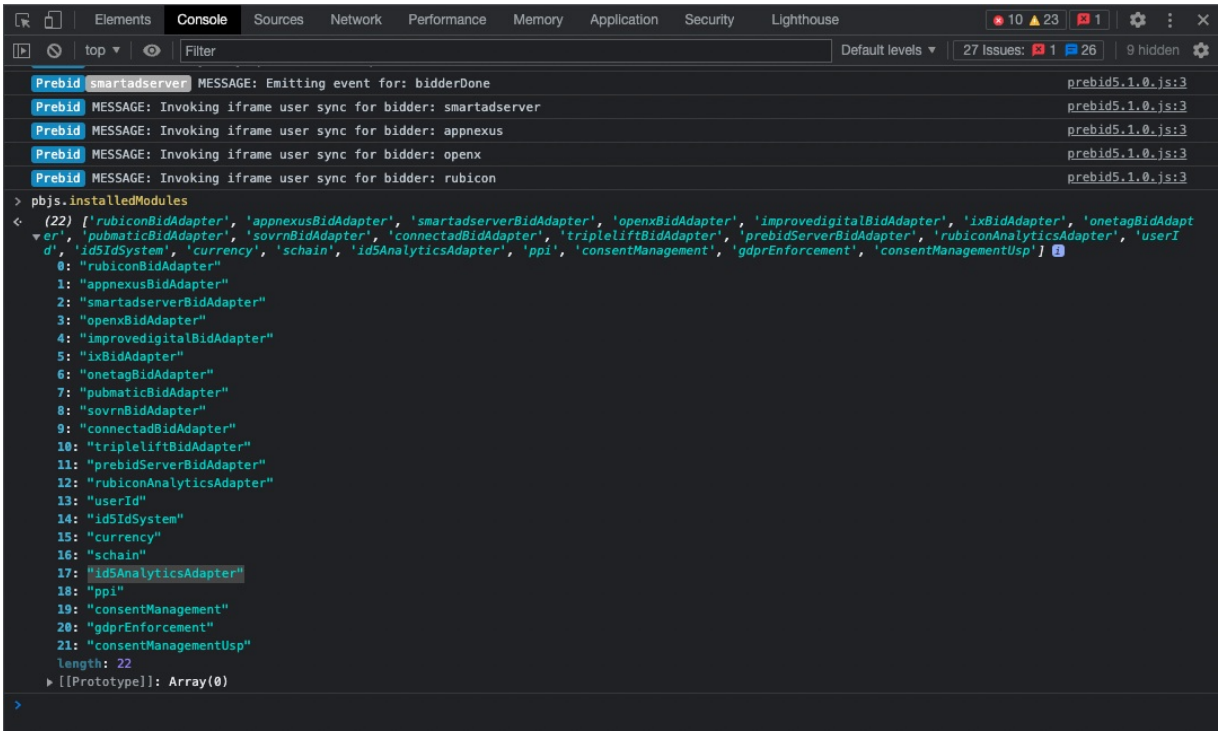
```
pbjs.enableAnalytics({
  provider: 'id5Analytics',
  options: {
    partnerId: 173   // change to the Partner Number you received from ID5
  }
});
```

To analyze the results of the ID5 ID, the adapter uses the A/B Testing feature described in the Prebid.js User ID Module. This enables us to evaluate key KPIs such as average bid CPM, average winning bid CPM, bid response rate, and bid density when the ID5 ID is present versus when it is not (control group).

How to Check if the ID5 Analytics Module is installed correctly

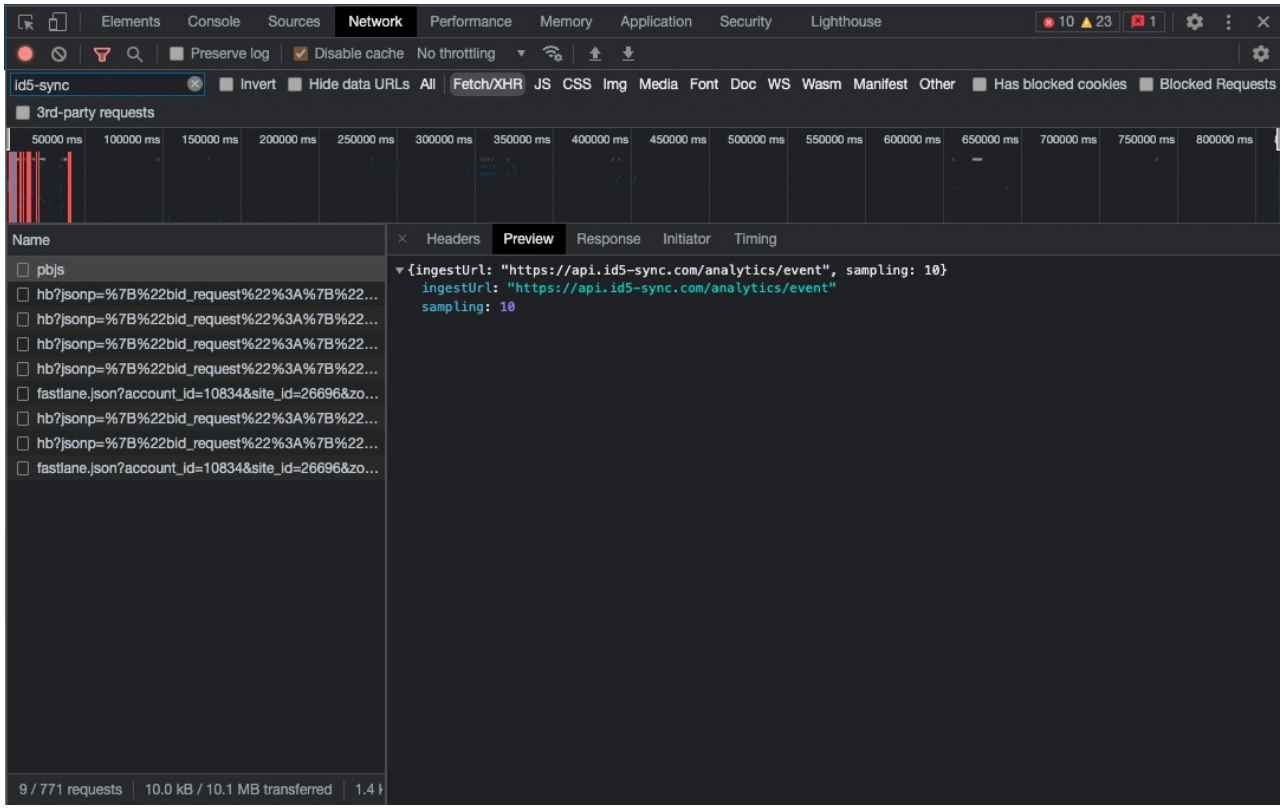
The following steps describe how to check your ID5 prebid integration:

1. Open a fresh incognito Chrome page, block 3rd party cookies and open the Developer Tools (Right Click -> Inspect)
2. Load a web page where Prebid with the ID5 module is setup;
3. Move to the “Console” and type: `pbjs.installedModules` . This will return all the modules from your prebid configuration;
4. Among all modules returned you should be able to see `id5AnalyticsAdapter` . Seeing the id5AnalyticsAdapter will mean that the ID5 Identity Insights were correctly installed:



How to Check if the ID5 Analytics Adapter Is Configured Correctly

1. Open a fresh incognito Chrome page, block 3rd party cookies and open the Developer Tools (Right Click -> Inspect);
2. Load a web page where Prebid is configured with the ID5 module setup;
3. On the “Network” tab filter the calls on type XHR and/or `id5-sync` word.
4. You should now see a call named `{your_partner_id}.json` and another one named `pbjs`. The call named `{your_partner_id}.json` should retrieve the ID5 ID. The call named `pbjs` is made by the analytics module;
5. Clicking on the call and after clicking on “Preview” will allow you to see the response from ID5: `{ingestUrl: "https://api.id5-sync.com/analytics/event", sampling: 10}`
6. Seeing the `pbjs` call and the response from point 6 will mean that the implementation is correct.



5. Integrate TrueLink (Recommended)

TrueLink is an additional client-side integration method that can complement a standard ID5 JS API or Prebid (version 9.2.0 and above) implementation. With a TrueLink integration, ID5 can create a cross-domain signal called the TrueLink ID for a single user within a specific browser. The TrueLink ID is produced independently of third party cookies by redirecting the user through an ID5 operated domain and setting the ID as a first party cookie.

This TrueLink signal is utilized by ID5 for cross domain user reconciliation and the generation of a high-quality ID5 ID. Publishers can optionally access a publisher first-party user identifier called the ID5 Guarded Publisher ID (GPID). The GPID is a publisher specific version of the ID5 ID which remains unique for a user across their owned and operated properties within a given browser environment. It's quality is implicitly linked to the provision of signals such as hashed emails and TrueLink signals. The GPID can be used to facilitate use cases such as cross-domain audience building and activation, all without relying on third-party cookies. It may also be used as a PPID within Google Ad Manager.

To integrate TrueLink, follow the instructions here.

6. Add the ID5 tag to your Webpage (Recommended)

While third-party cookies are still in use, ID5 can synchronize with ad tech vendors to enhance user recognition, improving audience addressability and monetization.

Follow the steps outlined here.

7. Use ID5's GPID as a first party identifier or PPID (Optional)

ID5 can optionally provide publisher and advertisers with access to a partner-specific version of the ID5 ID, known as the Guarded Publisher ID (GPID). The GPID can be used for various publisher or advertiser purposes, including cross-domain audience building, audience activation, measurement, and attribution. The GPID can also

be used as a Publisher Provided Identifier (PPID) in ad servers like Google Ad Manager or AppNexus. Google's PPID, for example, enables publishers to:

- Apply audience-based ad delivery controls, such as frequency capping and sequential ad rotation.
- Enhance audience segmentation and targeting across devices.
- Potentially increase CPMs from Google's buying stack.

You can find out more about the GPID and how to get it enabledhere.

ID5 Prebid.js User ID ModuleTroubleshooting

07/08/2026 7:09 am EDT

Prebid troubleshooting: step-by-step guide

To assist publishers in troubleshooting their Prebid integration, we have created this step-by-step guide, to help them review their integration.

Step 1: Enable Prebid Debug Mode

Make sure to use a new private navigation window.
Enable Prebid debugging mode:

- **Option 1**: Append `pbjs_debug=true` to the URL.
- **Option 2:** Use Prebid debug extension, Professor Prebid - Ensure the extension is enabled in private navigation mode.

Step 2: CMP implementation

If your website operates in a country subject to Data Protection laws, you must have a Consent Management Platform (CMP) on your website properties to collect user consent choices and make them available for any adapters.

On the website's page when the CMP pops up, you can load the Prebid library but you mustn't load any adapters or modules or send bid requests to bidders before obtaining the user's consent.

Step 3: Required modules

To distribute ID5 ID, ensure to include both the Consent Management module (if applicable) and the id5IdSystem submodule, in addition to the other modules you normally include in your prebid library.

To check your modules:

- **Option 1:** Type in your browser Console tab `pbjs.installedModules`
- **Option 2:** Open Professor Prebid, and go to the *Config* tab.

Step 4: ID5 settings inside the Prebid config

All the instructions to set the id5idSystem module into your Prebid Config can be found on ID5 Prebid User ID Module page.

To check Prebid settings:

- **Option 1:** Type in your browser Console tab `pbjs.getConfig()`
- **Option 2:** Open Professor Prebid, go to the *UserID* tab, and then click on *Config*

Step 5 - Call the Prebid elements in the right order

To ensure everything is properly implemented and to send an id5 id on every bid request you send to

bidders, you must respect a certain order when you initialize elements within your Prebid wrapper.

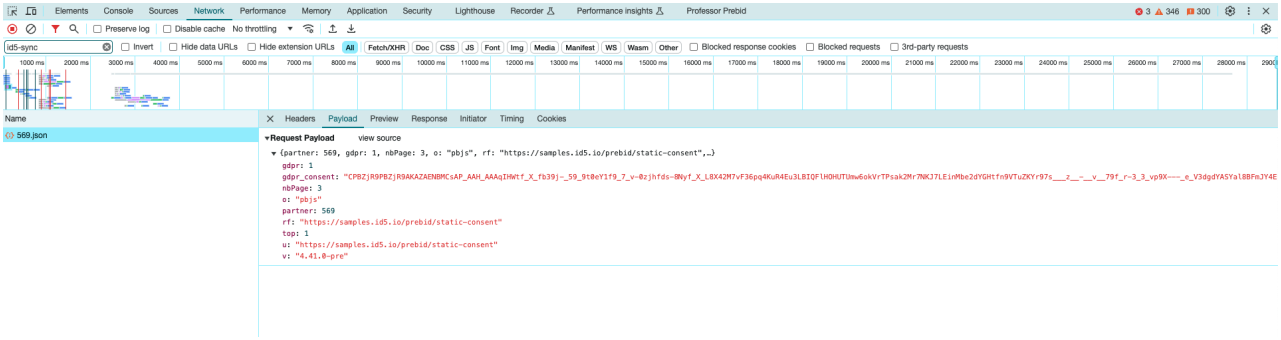
To inspect the order,in a new private navigation window with the browser console open, you should append `&pbjs_debug=true` at the end of the URL, to enable Prebid debug messages in the console.

Please find below the correct order

1. CMP appears - wait for the user's consent
2. Initialization of the `consentManagement module` which fetches user's consent
3. Initialization of the `id5idSystem module` - user consent is passed by the Consent Management module
4. Sending bid requests to bidders with ID5 ID

Step 6: Inspect the ID5 request call

In the Network tab of your browser, search for 'id5-sync' and inspect the request that is named `v3` (on older versions of prebid the request will be named with your ID5 Partner ID).
To check if an ID5 ID was return, you should inspect the reesponse of the request by going to the `Preview` tabe and check out the `universal_uid` paramer.
You can also inspect one of your bidder requests to see if the encrypted ID5 id is correctly passed.



ID5 Prebid.js Analytics Module - Identity Insights

07/08/2026 7:10 am EDT

ID5 Prebid Identity Insights

> Identity Insights is enabled through ID5's Analytics Adapted for Prebid, available for Prebid.js version 7.27.0 or higher

To help publishers better understand the value of working with ID5, we have launched Identity Insights - an ID5 analytics adapter for Prebid. With just a few additional lines of configuration, ID5 can capture prebid event data to provide publishers with insights into the value of the ID5 ID to their business. For now, insights will be provided in the form of a report shared by the ID5 publisher support team.

Enabling Identity Insights

To enable ID5 Identity Insights within Prebid, you will need to include the ID5 analytics adapter when you build Prebid:

```
gulp build –modules=userId,id5IdSystem,id5AnalyticsAdapter
```

In addition to including the ID5 Analytics Adapter in your prebid build, you will also need to ensure you enable analytics (this is in addition to the configuring the ID5 Prebid User ID module):

```
pbjs.enableAnalytics({
  provider: 'id5Analytics',
  options: {
    partnerId: 173    // change to the Partner Number you received from ID5
  }
});
```

Configuring the ID5 User ID Module for Identity Insights

The ID5 Analytics Adapter requires the ID5 User ID submodule to be configured in Prebid.js to capture event data effectively.

You'll need to enable A/B testing in the ID5 User ID submodule to compare performance metrics (e.g., bid CPM, response rate) with and without the ID5 ID.

Below is a simplified configuration for the ID5 User ID submodule, including A/B testing, to work with Identity Insights (ensure this is set before enabling analytics or requesting bids):

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: 'id5Id',
      params: {
        partner: 173,           // Required: your ID5 Partner Number
        abTesting: {            // Required for Identity Insights
          enabled: true,        // Enable A/B testing
          controlGroupPct: 0.1   // 10% of users in control group (no ID5 ID)
        }
      },
      storage: {
        type: 'html5',          // Required
        name: 'id5id',          // Required
        expires: 90,            // Storage duration in days
        refreshInSeconds: 7200   // Refresh ID every 2 hours
      }
    }],
    auctionDelay: 250           // Optional: max auction delay in ms
  }
});
```

Notes:

- A/B Testing is needed for using Identity Insights, as it allows you to directly compare performance between users who receive an ID5 ID and a control group that does not.
- The `partnerId` in the `pbjs.enableAnalytics` call must match the **partner** number provided in the `userIds` configuration.

Checking Your Implementation

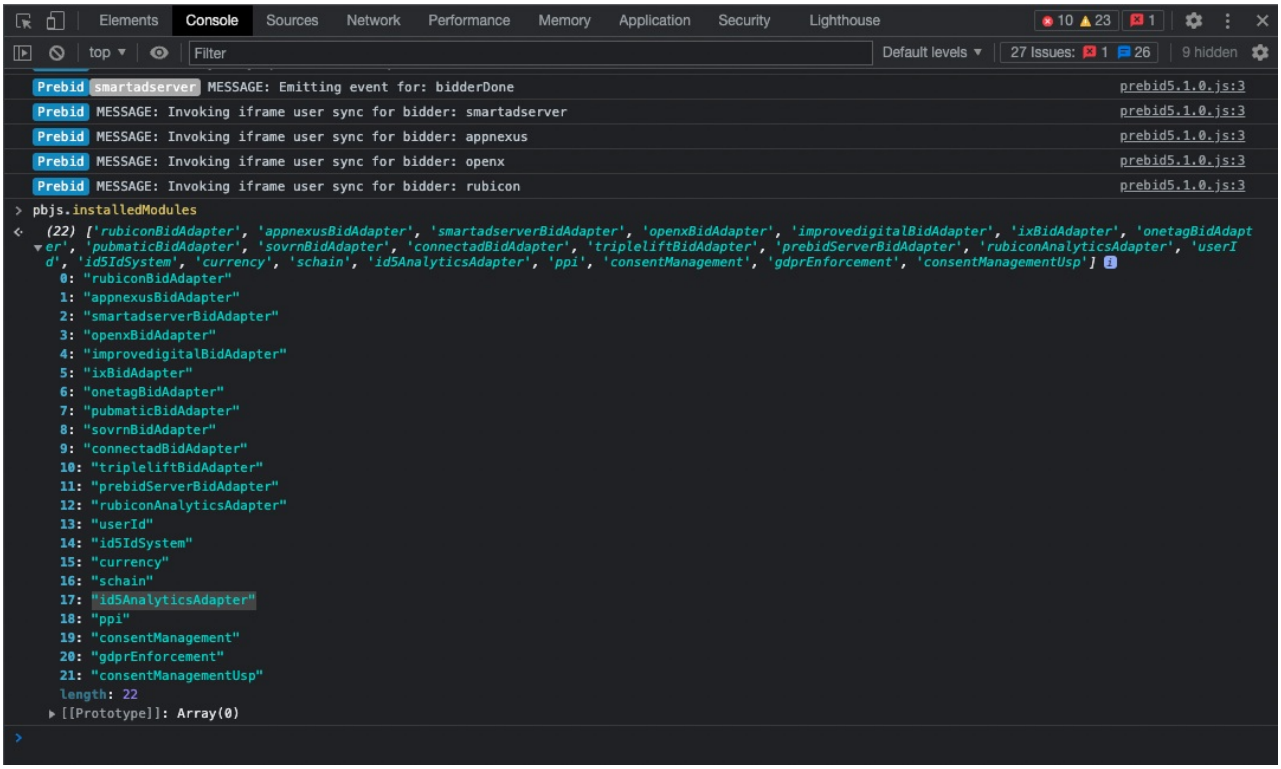
How to Check if the Prebid Modules Are Installed Correctly

1. Open a fresh incognito Chrome page, block 3rd party cookies and open the Developer Tools (Right Click -> Inspect)

2. Load a web page where Prebid with the ID5 module is setup;

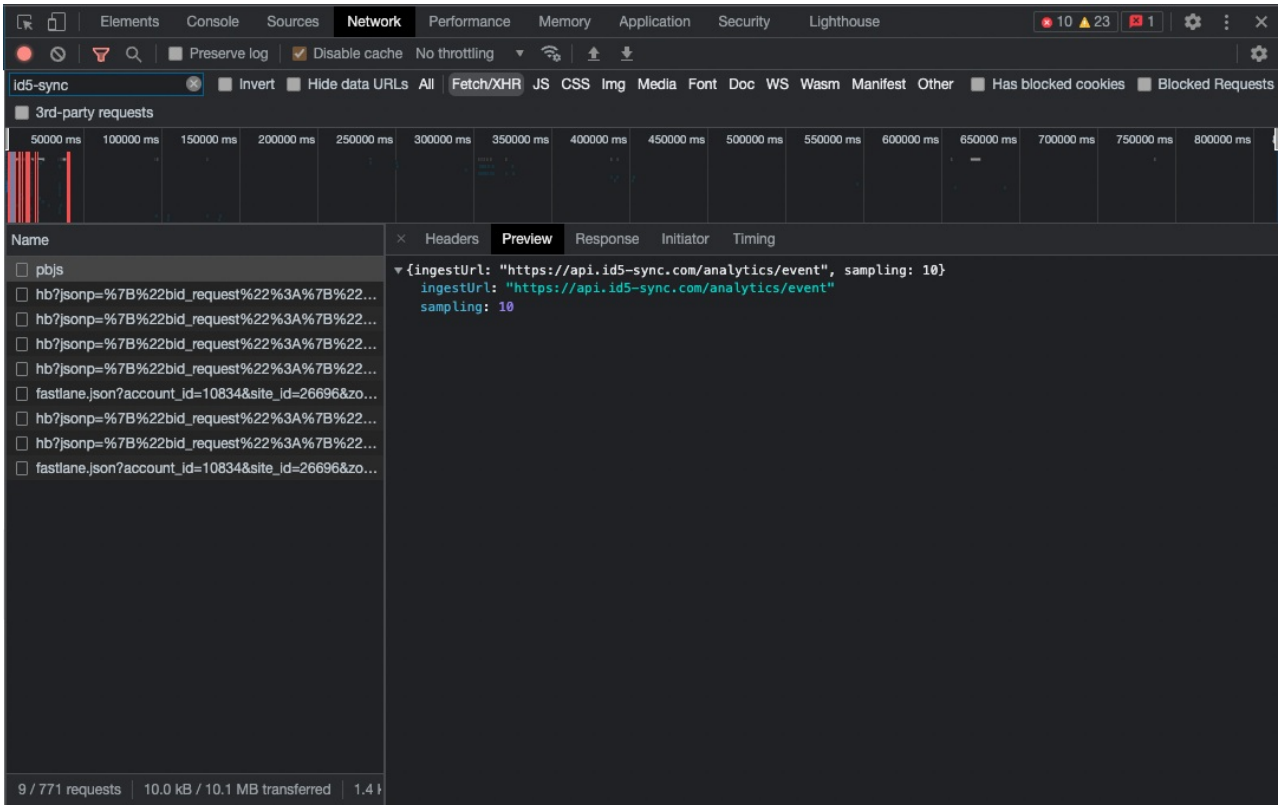
3. Move to the “Console” and type: `pbjs.installedModules`. This will return all the modules from your prebid configuration;

4. Among all modules returned, you should be able to see `id5AnalyticsAdapter`. Seeing the id5AnalyticsAdapter will mean that the ID5 Identity Insights were correctly installed:



How to Check if the ID5 Analytics Adapter Is Configured Correctly

1. Open a fresh incognito Chrome page, block 3rd party cookies and open the Developer Tools (Right Click -> Inspect);
2. Load a web page where Prebid is configured with the ID5 module setup;
3. On the “Network” tab filter the calls on type XHR and/or `id5-sync` word.
4. You should now see a call named `{your_partner_id}.json` and another one named `pbjs`. The call named `{your_partner_id}.json` should retrieve the ID5 ID. The call named `pbjs` is made by the analytics module;
5. Clicking on the call and after clicking on “Preview” will allow you to see the response from ID5: `{ingestUrl: "https://api.id5-sync.com/analytics/event", sampling: 10}`
6. Seeing the `pbjs` call and the response will mean that the implementation is correct.


Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
10 23
1
Preserve log
Disable cache
No throttling
id5-sync
Invert
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
3rd-party requests
50000 ms
100000 ms
150000 ms
200000 ms
250000 ms
300000 ms
350000 ms
400000 ms
450000 ms
500000 ms
550000 ms
600000 ms
650000 ms
700000 ms
750000 ms
800000 ms
Name
pbjs
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
fastlane.json?account_id=10834&site_id=26696&zo...
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
hb?jsonp=%7B%22bid_request%22%3A%7B%22...
fastlane.json?account_id=10834&site_id=26696&zo...
Headers
Preview
Response
Initiator
Timing
{ingestUrl: "https://api.id5-sync.com/analytics/event", sampling: 10}
ingestUrl: "https://api.id5-sync.com/analytics/event"
sampling: 10
9 / 771 requests
10.0 kB / 10.1 MB transferred

Minimum Prebid.js Integration Requirements for ID5's Real Time Bid Enrichment Service

07/08/2026 7:10 am EDT

Prebid.js Integration Requirements for ID5's Real Time Bid Enrichment Service

ID5's Real-Time Bid Enrichment Service is an optional, paid-for service that enables publishers and publisher monetization platforms to improve inventory addressability. When enabled, the ID5 Graph is used to enrich the EID object provided in ID5's response with SSP and DSP cookie IDs alongside ID5 IDs.

Partners that want ID5 to enable this service via prebid.js must meet the following integration requirements.

Build Prebid.js with the ID5 User ID Module and ID5 Analytics module

Bid Enrichment requires **Prebid.js 9.31 or later**. To receive GAM targeting keys, you must either:
- use **Prebid 9.31+** with `externalModuleUrl` configured and `storage.refreshInSeconds` set to 0
or
- upgrade to **Prebid 10.13.10 or later** if you are not using `externalModuleUrl`, as this version includes native support for GAM targeting keys without relying on the external module.

Follow the step-by-step instructions for installing and configuring the Prebid.js User ID Module with the ID5 ID and the ID5 Prebid Analytics Module for Identity Insights.

Once the *userId, id5IdSystem,* and *id5Id5AnalyticsAdapter* are included in your modules, you will also need to ensure you enable analytics (in addition to configuring the ID5 Prebid User ID module):

```
pbjs.enableAnalytics({
  provider: 'id5Analytics',
  options: {
    partnerId: 173    // change to the Partner Number you received from ID5
  }
});
```

Publishers with real time bid enrichment enabled, should **NOT** configure the abtesting parameter in the prebid user ID module. ID5 will automatically set up an A/B test with a 90:10 split between enriched and control groups. The % split can be customised upon client request. This means:

- 90% of eligible users: Enriched group - will receive enriched IDs
- 10% of eligible users: Control group - will not receive enriched IDs

Note this split is done independently of any other A/B testing the publisher might have configured. If the ID5 abtesting parameter is configured whilst bid enrichment is enabled, the enrichment/control group split will apply to the **normal group** that has the ID5 ID enabled.

Example

If a publisher enables ID5's AB Testing with an 85:15 split, then 85% of users are assigned to the **normal group** and 15% to the **control group**. If bid enrichment is also enabled with a 90:10 split, this applies only within the normal group: **90% of those users** will receive enrichment, while **10% will serve as the enrichment control group**. By running the two tests at the same time, the bid enrichment potential will be limited and the results of the AB Test designed to assess uplift associated with presence of the ID5 ID will be biased.

Optimise your ID5 Prebid configuration

Ensure you have installed the ID5 User ID module and ID5 Analytics Module as part of your prebid configuration. The ID5 bid enrichment service requires the ID5 User ID module to be configured as follows:

Name	Value	Description
externalModuleUrl	`https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js`	Ensures you are benefiting from the latest version of our code and your bid enrichment configuration
canCookieSync	`true`	Maximises match rates with SSPs and DSPs through cookie syncing with user-consented vendors.
gamTargetingPrefix	`id5`	When a non-empty `params.gamTargetingPrefix` is configured **and** the ID5 module has successfully initialized, the module sets GAM targeting keys that should be used to generate uplift reporting in Google Ad Manager. Enable GAM reporting to track performance metrics - GAM targeting support is from 10.13.0 or when externalModuleUrl is used. *
exposeTargeting	true	In cases when it's not feasible to use `gamTargetingPrefix` described above to measure uplift, `exposeTargeting` can be used. It will expose the control and uplift groups via tags to the callbacks in a way described in Exposing targeting to external Reporting system.

*When a non-empty `params.gamTargetingPrefix` is configured, the module sets the following GAM targeting keys that will be available:

Key	Description

Key	Description
`{prefix}_id`	Set to `"y"` if a valid `id5id` is available. The key is **not set** if no ID is present.
`{prefix}_ab`	Set when A/B testing is enabled: `"n"` – Normal group (ID5 ID returned) `"c"` – Control group (no ID5 ID returned)
`{prefix}_enrich`	Set when **bid enrichment** is enabled: `"y"` – Enriched IDs returned `"s"` – Enrichment attempted, no enriched IDs found `"c"` – Control group (no enrichment was performed, used for uplift measurement)



To maximise value and match rates achieved by the **ID5 Real Time Bid Enrichment Service**, we recommend you pass all available signals e.g hashed emails (instructions) and upgrade your ID5 integration by deploying ID5's TrueLink solution.

Please preview the sample Prebid configuration below for reference:

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: 'id5Id',
      params: {
        partner: 173,           // change to the Partner Number you received from ID5
        externalModuleUrl: 'https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js',
        pd: 'MTLiNTBjY...'      // visit our "Passing Signals to ID5" page to learn more
        abTesting: {
          enabled: false,
          controlGroupPct:
        }, canCookieSync: true,  gamTargetingPrefix: "id5"
      },
      storage: {
        type: 'html5',          // 'html5' is the required storage type
        name: 'id5id',          // 'id5id' is the required storage name
        expires: 90,            // storage lasts for 90 days
        refreshInSeconds: 7200   // refresh ID every 2 hours to ensure it's fresh
      }
    }],
    auctionDelay: 250           // 250ms maximum auction delay, applies to all userId modules
  }
});
```

ID5 User ID Module for Prebid Server Java

10/15/2026 6:44 am EDT

Overview

The ID5 User ID module enriches the OpenRTB auction requests your Prebid Server sends to bidders with the ID5 universal identifier. During each auction, it fetches an ID5 ID and injects it into the bid request as an Extended Identifier (EID), improving addressability whilst honoring user privacy choices.

- Fetches the ID5 universal identifier and injects it into bid requests as an EID (`user.eids`).
- Skips fetching and injection when an ID5 ID is already present in the request
- Honors privacy signals (GDPR/TCF, CCPA/US Privacy, COPPA, GPP).
- Lets you control rollout: filter by account, country, or bidder, and sample a percentage of traffic.
- Runs asynchronously and non-blocking - a failed or slow ID5 fetch never blocks the auction.
- Processes both web and mobile in-app traffic, including requests sent to your Prebid Server by the Prebid Mobile SDK.

Prerequisites

Register with ID5

The ID5 ID is free to use, but requires a registration with us.

Deploying for yourself: If you're integrating ID5 as part of your own Prebid Server setup, visit our website to sign up and request your **ID5 Partner ID**.

Deploying on behalf of clients: If you're integrating ID5 as part of a Prebid Server setup for your clients, each client must sign our ID5 agreement, and you must include their Partner ID in any requests you send us from their media property. If it's useful, ID5 can set up a co-branded sign-up page for your clients.

To deploy on behalf of your clients, you need to declare yourself as the provider name — please check with ID5 for the exact value to use in the `provider-name` field.

Add the Dependency

The module is published to **Maven Central**. Add it to the Maven build that assembles your Prebid Server distribution (the module/POM that produces your runnable PBS JAR), so the module is on the server's runtime classpath:

```
<dependency>
  <groupId>io.id5.prebid.server.hooks.modules</groupId>
  <artifactId>id5-user-id</artifactId>
  <version>3.43-rc.1</version>
</dependency>
```

Contact the ID5 team for advice on which version to use. In general, the module version tracks the Prebid Server version line it is built against, in the form `<PBS-major>.<PBS-minor>-<release>` . In Example:

Your Prebid Server Version	Module Version
3.43.x	`3.43-rc.1`

Browse all available versions on Maven Central

Enable and Configure the Module

Add the module configuration to your Prebid Server configuration file and configure hooks

Minimal Setup

```
hooks:
  id5-user-id:
    enabled: true
    provider-name: "my-pbs-host"                    # identifies who operates this PBS instance
    fetch-endpoint: "https://api.id5-sync.com/gs/v2"  # ID5 API endpoint
    partner: 173                                    # your ID5 Partner ID - see "Partner ID" below
```

Partner ID

> The Partner ID argument given to the **partner** parameter must be decided together with the ID5 team

The right choice depends on your specific case - it may be the host's own Partner ID, or, in some cases, the publishers' Partner IDs. The ID5 team will work with you to determine the appropriate Partner ID strategy and implement it according to your case and needs.

There are two ways to supply the Partner ID, matching the two common strategies:

1. Static Partner ID (default). Set the `partner` property (see the Configuration Reference below). The same Partner ID is used for every request. This is the simplest option when a single host-level Partner ID fits.

2. Dynamic Partner ID. When the Partner ID must vary per request - for example, derived from the account, channel, or publisher the auction comes from - implement the `Id5PartnerIdProvider` interface and register it as a Spring bean. The module picks it up automatically.

```
package com.mycompany.pbs;

import org.prebid.server.auction.model.AuctionContext;
import org.prebid.server.hooks.modules.id5.userid.v1.model.Id5PartnerIdProvider;
import org.springframework.stereotype.Component;

import java.util.Optional;

@Component
public class MyId5PartnerIdProvider implements Id5PartnerIdProvider {

    @Override
    public Optional<Long> getPartnerId(AuctionContext auctionContext) {
        // Derive the Partner ID from the auction context
        // (account, bid request, privacy data, ...).
        // Return Optional.empty() to skip the ID5 fetch for this request.
        return Optional.of(1234L);
    }
}
```

When you use a dynamic provider:

- `getPartnerId` receives the full `AuctionContext` (account, bid request, privacy information), so you can select the Partner ID per request.
- Returning `Optional.empty()` skips the ID5 fetch for that request.
  - **Omit the `partner` property.** The built-in constant provider is only created when `partner` is set, so leaving it out avoids having two competing `Id5PartnerIdProvider` beans. (Alternatively, mark your bean `@Primary`.)

Configuration Reference

Property	Required	Type	Default	Description
`enabled`	Yes	boolean	-	Must be `true` to activate the module.
`provider-name`	Yes	string	-	Identifier sent to the ID5 API describing who operates this Prebid Server instance (e.g. `"my-company-pbs"`).
`fetch-endpoint`	Yes	string	-	ID5 API endpoint URL: `https://api.id5-sync.com/gs/v2`.
`partner`	Conditional	long	-	Your ID5 Partner ID, for a single static Partner ID. Required unless you provide a custom `Id5PartnerIdProvider` - see Partner ID above.
`inserter-name`	No	string	*none*	Canonical domain of the entity adding the EID (e.g. `"my-company.com"`). Written to the EID's `inserter` field.
`fetch-sampling-rate`	No	double	`1.0`	Fraction of requests to process, `0.0` - `1.0` (e.g. `0.1` = 10%). Useful for gradual rollout.
`account-filter`	No	filter	*none*	Limit which accounts trigger an ID5 fetch.
`country-filter`	No	filter	*none*	Limit which countries trigger an ID5 fetch.
`bidder-filter`	No	filter	*none*	Limit which bidders receive the ID5 EID.

Filters

Each filter takes an `exclude` flag and a list of `values`:

```
<filter-name>:
  exclude: false     # false = allowlist (only these), true = blocklist (all except these)
  values:
    - value1
    - value2
```

- `exclude: false` → **allowlist**: only matching requests proceed. Empty list = allow all.
- `exclude: true` → **blocklist**: matching requests are skipped. Empty list = allow all.

Full Configuration Example

```
hooks:
  id5-user-id:
    enabled: true
    provider-name: "my-pbs-host"
    fetch-endpoint: "https://api.id5-sync.com/gs/v2"
    partner: 1234                 # your ID5 Partner ID - see "Partner ID" above
    inserter-name: "my-company.com"
    fetch-sampling-rate: 0.8
    account-filter:               # all accounts except "test-account"
      exclude: true
      values: [test-account]
    country-filter:               # only these countries
      exclude: false
      values: [US, GB, DE, FR]
    bidder-filter:                # only these bidders receive the ID5 EID
      exclude: false
      values: [rubicon, appnexus, pubmatic]
```

Register the Module's Hooks

The module runs through two hooks that you must register in a Prebid Server **execution plan**:

- `id5-user-id-fetch-hook` - stage `processed-auction-request`
- `id5-user-id-inject-hook` - stage `bidder-request`

You can register them globally (host execution plan) or per account. Per-account example:

```
accounts:
  - id: "1001"
    status: active
    hooks:
      execution-plan:
        {
          "endpoints": {
            "/openrtb2/auction": {
              "stages": {
                "processed-auction-request": {
                  "groups": [
                    {
                      "hook-sequence": [
                        { "module-code": "id5-user-id", "hook-impl-code": "id5-user-id-fetch-hook" }
                      ]
                    }
                  ]
                },
                "bidder-request": {
                  "groups": [
                    {
                      "hook-sequence": [
                        { "module-code": "id5-user-id", "hook-impl-code": "id5-user-id-inject-hook" }
                      ]
                    }
                  ]
                }
              }
            }
          }
        }
```

What Gets Added to the Bid Request

When an ID5 ID is fetched, the module adds an EID to the `user.eids` array of requests sent to the selected bidders.

Example EID added:

```
{
 "user": {
  "eids": [
   {
    "source": "id5-sync.com",
    "uids": [
     {
      "id": "ID5*YsvxY...",
      "atype": 1,
      "ext": { "linkType": 2, "pba": "jWwv+..." }
     }
    ],
    "inserter": "my-company.com"
   }
  ]
 }
}
```

Privacy & Data Sharing

The module forwards available privacy signals to the ID5 API, which honors them when generating an identifier:

- **GDPR / TCF** - consent string and GDPR-applies flag
- **CCPA** - US Privacy string
- **COPPA** - COPPA flag
- **GPP** - GPP string and applicable section IDs

To resolve an ID5 ID, the module may send request and device signals (such as site domain/referrer domain, app bundle, device IP, user agent, and mobile advertising ID) to the ID5 API.

> Ensure that your or your client's privacy policy and consent setup cover data sharing with ID5.

Verify the Integration

1. Confirm the dependency is on your server's classpath and the module is `enabled: true`.
2. Confirm the execution plan registers both hooks.
3. Send an auction request and confirm that an `id5-sync.com` EID appears in `user.eids` of the bidder requests (for the bidders allowed by your filters).

If no EID appears, check that the required properties are set, the execution plan is applied to the account/endpoint in use, and no filter (account, country, bidder, or sampling) is excluding the request.

Support

For help with the module or the ID5 service, contact **support@id5.io**.

Mobile In-App Integration - Client-to-Server

09/23/2026 10:49 am EDT

> **Beta:** The client-to-server integration and the ID5 Mobile SDK are currently in beta. APIs and behaviour may still change. We actively welcome your feedback - if you run into any issues, reach out at `support@id5.io,` and we'll address them quickly. Adapters for Prebid and Google Ad Manager will be released in the near future.

The **client-to-server** integration is an alternative to the server-to-server method for publishers who prefer not to operate backend infrastructure to call ID5. Instead of building and sending requests from your own servers, you embed the **ID5 Mobile SDK** directly in your app. The SDK handles the entire identity workflow on your behalf - collecting signals, enforcing consent, building and signing the request, caching the response and signature on the device, and delivering the encrypted ID5 ID to your app.

This approach lets you integrate ID5 with no server-side development or maintenance. The following guide walks through the integration step by step.

1. Register with ID5

You need an ID5 **partner ID** to integrate. If you don't have one yet, contact ID5 or email `contact@id5.io`.

> **NOTICE: TERMS OF SERVICE & BINDING AGREEMENT** - Use of the ID5 SDK and ID5 identifiers is subject to your ID5 partner agreement and applicable data-protection obligations. Ensure you have a valid legal basis and the necessary consent before collecting and sharing user signals.

2. Add the ID5 Mobile SDK

Android

Add the SDK dependency to your Gradle build:

```
implementation("io.id5:id5-sdk-core:0.1.0")
```

Full installation, configuration, and API reference are maintained in the repository:
github.com/id5io/id5-sdk-android

iOS

Release soon. The iOS Mobile SDK is in progress. This section will be updated with installation and usage details when it ships.

3. Configure consent

If you operate under GDPR, initialize your **CMP** (Consent Management Platform) *before* the SDK. The SDK reads the IAB TCF snapshot and waits for a complete consent signal before fetching.

- Under GDPR, both **Purpose 1** and **vendor 131 (ID5)** consent are required for the SDK to read the device cache and the advertising ID.
- On **Android 13+**, declare `com.google.android.gms.permission.AD_ID` in your manifest to allow

advertising-ID lookup.
- On **iOS**, App Tracking Transparency (ATT) authorization is also required to read the IDFA. *(Applies when the iOS SDK ships.)*

4. Optimize SDK API Configuration

To maximise addressability and produce the highest-quality ID5 ID, provide as many signals as possible. The more signals you supply, the higher the ID5 match rate and the better the cross-device linking. In the client-to-server flow, you supply these signals through the SDK's `PartnerConfig` when you initialize it - the SDK hashes and packages them for you.

Signals you can provide:

- **Hashed email (HEM)** - the single strongest signal for match rate. Pass the user's email; the SDK produces the normalized hash.
- **Hashed phone number** - a strong deterministic signal where available. Pass the user's phone; the SDK produces the normalized hash.
- **Partner user ID** - your own stable user identifier, used for cross-device reconciliation.
- **Advertising ID (IDFA / GAID)** - collected automatically by the SDK, subject to consent (ATT on iOS, device-access consent on Android).

Example (Android):

```
PartnerConfig partner = PartnerConfig.builder()
    .partnerId(<your-partner-id>)
    .hashedEmail(HashedSignal.fromEmail("user@example.com"))
    // additional signals (hashed phone, partner user ID) are configured
    .build();
```

Provide these signals whenever you have them and a valid legal basis to do so. Supplying at least one deterministic signal (typically HEM) is strongly recommended. For the full list of supported signals and exact builder methods, see the SDK README.

5. Retrieve and use the ID5 ID

Once initialized, the SDK delivers the encrypted ID5 ID to your app. To share it with demand partners, extract the EID objects from the ID5 response and include them in your bid requests as an `eids` array - the same pattern as the server-to-server integration.

See the SDK README for how to read the ID (listener or one-shot) and the exact response fields.

Related Articles

- Mobile In-App Integration (Server-to-Server)
- ID5 SDK for Android (GitHub)

Mobile In-App Integration - Server To Server

10/07/2026 4:53 pm EDT

Overview

We recommend that you deploy ID5 in mobile in-app environments via a server to server integration. We do support alternative integration pathways depending on your requirements. If required, please reach out to your ID5 customer services representative.

To ensure accurate cross-domain and cross-device reconciliation, clients should provide signals, such as hashed emails and MAIDs in the request. We also expect you to store the 'signature' provided in our response on the user's device and provision it to us in future requests. This is integral to an optimal set up since it helps us re-identify consenting users even if their signals change as well as respect opt out requests.

Below are general instructions on how to retrieve an ID5 ID server-side for mobile app inventory and make it available in bid requests to your demand partners. The auction should be delayed in order to retrieve the user's privacy preferences and the ID5 ID prior to sending out bid requests; the amount you delay may vary and so we recommend making it configurable for optimization purposes.

Process Flow

1. Check the stored cache for an available ID5 ID
    - ***IF*** there is no cached ID
      OR if the ID needs to be refreshed (we recommend a cache of max 8 hours)
      OR if the user's privacy preferences have changed:
        - Initialize a new HTTP POST request to the ID5 endpoint. If you previously had a stored response for a given user on their device, then subsequent requests must include the "signature" as part of the request to ID5;
        - Store the response from ID5 locally (either in local storage or in a database on the server);
    - ***ELSE*** if there is a valid, up-to-date value in cache:
        - Pull the latest ID5 ID response from the cache
2. Using the response from step 1 above (from the request or cache), prepare the bid request data.
    - Extract the **eid object(s)** from the ids structure (e.g., from `ids.id5id.eid` for the ID5 Universal ID, and any additional enriched IDs if services like Real-Time Bid Enrichment are enabled).
3. Include this data in each bid request to your demand partners as an eids array in the user object. The EID structure is already formatted for direct use in OpenRTB bid requests. For more information on how this is typically done, including handling provenance fields for enriched IDs, review the OpenRTB EID Enrichment Guide.

Caching the response

The response we return for the request will need to be cached as some of the response data needs to be used in future requests for the users. Use the `ts` parameter sent in the request in order to set a TTL for the cache refresh and a TTL for the cache deletion.

- The TTL for the cache refresh should be set to 8 hours. After the 8 hours, on the back of a user event in the app, a new request to ID5 should be made and the cached data refreshed using the data from the API response.
- The TTL for the cache deletion should be set to 30 days. The TTL for the cache deletion should be refreshed with every response from the API. If there will not be a user event in the app, in the 30 days time window, the data should be deleted.

Building Server-Side Request

Server-Side Fetch endpoints for in-app

> **Global**
>
> https://api.id5-sync.com/ga/v1

Request Type

HTTP POST with JSON body.

Request Headers

`Content-Type: application/json`

Partner Number

The `PARTNER` in the request body will be an ID5-provided Partner Number. This value will be static for you once we set you up in our system. You may use the endpoint during testing with the Partner Number 173. If you haven't already been assigned a Partner Number, please contact us to request one.

Server-Side Request Parameters

Request Body

Name	Required	Type	Description	Format	Example
ts	x	string	Timestamp in form of string which extends the ISO-8601 extended offset date-time format to add the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
partner	x	integer	Partner Number provided by ID5	int32	123
			A platform-specific application identifier intended to be unique to		com.foo.mygam

Name	Required	Type	Description	Format	Example
bundle		string	the app. On Android, this should be a bundle or package name. On iOS, it is typically a numeric ID.		
ver	x		Application version		
provider		string	Provider of the service (value to be provided by ID5)		
ip	x	string	IPv4 address closest to device	ipv4 dot-decimal	
ua	x	string	The User Agent of the device's default browser		Mozilla/5.0 (Linux; Android 11; moto e20 Build/RONS31.267-88-3; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/109.0.5414.118 Mobile Safari/537.36
signature	after initial call	string	The ID5 signature from a previous call, cached on the device or your server-side		
gdpr		integer	1 if gdpr applies to this request, 0 otherwise		
gdpr_consent		string	(where applicable) the TCF compliant consent string or see 'allowed_vendors'		
allowed_vendors		string array	ID5 Partner identifiers (starting with 'ID5-') or IAB Vendor IDs of vendors allowed to use the ID5ID		`['ID5-78','134']`
us_privacy		string	US Privacy Consent		`1YNN`
gpp_sid		string	The GPP section ID(s) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation		`6,7`

Name	Required	Type	Description	Format	Example
gpp_string		string	A valid IAB Global Privacy Platform consent string.		`DBABzw~1---~BAAAAAAAAQA.QA`
name		string	App name (may be aliased at the publisher's request)		
domain		string	Domain of the app		`mygame.example.com`
maid		string	The device identifier (IDFA in Apple systems, GAID in Android systems)	uuid	
maid_type		string	`idfa` or `gaid`		`idfa`
hem		string	sha256 hash of the cleansed e-mail address. Learn how to cleanse the data here.	sha256	`f97ea86ed181d60b0ba62a30579f1e10ad71eaf21b548e173de75718065c533f`
phone		string	sha256 hash of the cleansed phone number	sha256	
idfv		string	Apple ID for Vendors	uuid	`3fa85f64-5717-4562-b3fc-2c963f66afa6`
puid		string	Partner specific user ID		
ipv6		string	The IPV6 of the device	ipv6	`2001:0db8:5b96:0000:0000:426f:8e17:642a`
country		string	Country the user is located in	ISO-3166-1-alpha-2	`AD`
region		string	Region code using ISO-3166-2; 2-letter state code if USA	ISO-3166-2; 2-letter state code if USA	
city		string	City using United Nations Code for Trade & Transport Locations format	United Nations Code for Trade & Transport Locations format	
att	x (for iOS requests)	boolean	If the user selected "Ask App not to Track", set the value to `1`, otherwise omit the field or set the value to `0`. Also omit the field for all non-IOS requests.	boolean	`1` / `0`
accept_language		string	A string representing languages accepted by end-user device, should be compatible with browser Accept-Language header.	Accept-Language header format	`de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7`
			The segment ids a user may belong to and the destination platform that		

Name	Required	Type	Description	Format	Example
segments		objects array	the segments should be pushed to. Only certain destination platforms are supported and there are backend configurations that need to be made in both ID5's and the destination platform's systems before this feature can be used. Please reach out to your ID5 representative or contact@id5.io for more information and to get started.	Array of segment objects (see below Segment object)	`[{ "destination": "999", "ids": [ "12345", "67890" ] }]`

Segment Object

Name	Required	Type	Description	Example
destination	x	string	The destination platform. Should be the IAB Vendor ID	`'999'`
ids	x	string array	List of segment ids to add the user to	`[ '12345', '67890' ]`

Response

Response Body

Name	Required	Type	Description	Format	Example
created_at	x	string	Timestamp in the form of a string that extends the ISO-8601 extended offset date-time format to include the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
original_uid	x	string	A first-party user ID that remains stable for this user on the domain. It is intended for publisher reference only and should not be shared with other partners. The value will be encrypted and may change periodically, even for the same user on the same domain, while the underlying value remains stable. If ID5 lacks consent, then the value will be "0".		

Name	Required	Type	Description	Format	Example
universal_uid	x	string	This is the UID used for sharing with other parties. The value will be encrypted and will change periodically, even for the same user on the same domain. If ID5 does not have consent, then the value will be "0".		
ids	x	object	Container for all identity providers returned by ID5 in OpenRTB-compatible format See the ids object below.		
privacy	x	object	An object containing privacy information See the privacy object below.		
signature	x	string	The ID5 Signature is a string that must be stored by the caller (either cached on the device or your server-side) and sent back to ID5 with all future requests. See below for details about the Signature.		
ext	x	object	See the extension object below.		
gpId		string	Guarded Publisher ID (see more here)		
enrichment		object	Present only when Real Time Bid Enrichment is enabled for given request. See Enrichment object		

IDS Object

The `ids object` provides identifier data for the ID5 ID and any other Extended Identifiers (EIDs) received from ID5.

Each sub-object is named (e.g., `id5id` for the ID5 Universal ID) and contains an OpenRTB-compatible EID structure. Additional third-party identifiers may be included based on enabled services, such as Real-Time Bid Enrichment.

For example, the **ids object** might include:

- `id5id` : The ID5 Universal ID.

- `euid` : European Unified ID.
- `id5-[provider]` Other providers like PubMatic, Magnite, Xandr, TTD, etc.
  - For example: `id5-ttd` : The Trade Desk ID (enriched via ID5)

Each identifier contains an `eid object` that mirrors the structure of OpenRTB's eids field, enabling direct use in bid requests. For enriched IDs, provenance fields such as **mm** (match method), **matcher**, and **inserter** may be added to improve transparency and compliance with IAB best practices.

Name	Required	Type	Description	Format	Example
id5id	x	object	The container for the ID5 Universal ID. For details, see the EID object below.		

EID Object

Name	Required	Type	Description	Format	Example
source	x	string	Source or technology provider responsible for the set of included IDs. Expressed as a top-level domain.		id5-sync.com
uids	x	object array	An array of extended ID UID objects from the given source. Refer to Object: UIDS.		
mm	For enriched IDs	integer	The match method used by the matcher to establish the ID connection (e.g., exact match, probabilistic, or IP-based). This promotes transparency in identity resolution processes. Refer to AdCOM 1.0 Matching Methods	enum	2
matcher	For enriched IDs	string	The domain of the technology provider responsible for matching or bridging the ID, as defined by the mm value.		id5-sync.com
inserter	For enriched IDs	string	The canonical domain name of the entity (e.g., publisher, SSP, or wrapper) that added the EID to the bid request.		publisher-domain.com

UIDS Object

Name	Required	Type	Description	Format	Example
id	x	string	The identifier for the user. For ID5, the value equals the response field: `universal_uid`		
atype	x	integer	Type of user agent the match is from. Refer to AdCOM Agent Types.	enum	1
ext		object	An extension object for passing additional metadata about this specific ID. Further information on ID5 metadata is available here.		

Privacy Object

Name	Required	Type	Description	Format	Example
jurisdiction	x	string	The legal jurisdiction applicable to the request (e.g. "gdpr", "ccpa", etc), based on the location the request was made from	enum	gdpr

Name	Required	Type	Description	Format	Example
id5_consent	x	boolean	Indicates if ID5 had proper consent on the request to process the user's personal data.		

ext Object

Name	Type	Description	Example
linktype	integer	See details here	1

Enrichment Object

Name	Type	Description	Example
enrichment_selected	boolean	False if request was selected to be in enrichment control group, true if it was selected to be enriched (normal group)	false
enriched	boolean	true if request was selected to be enriched AND at least one enriched ID was found and returned for given request. False otherwise	false

Invalid Request Response

Name	Type	Description	Example
code	string	A code which univocally identifies the error	- partner_id_invalid - sha256_length_invalid - request_format_invalid - user_object_invalid - internal_id5_error
message	string	Human readable message about the error	- unknown partner - failed to parse json POST - missing ua - missing ts - ts older than allowed - invalid ts format - Fetch of ID5 ID disabled. Please contact ID5 to enable it. - Request from a country which is disallowed. Please contact ID5 to enable it.
type	string	A code which identifies the class of error	- validation_error - invalid_request_error - authentication_error

Example Request

Example Valid Response

Example Invalid Response

```
{
  "code": "partner_id_invalid",
  "message": "unknown partner",
  "type": "invalid_request_error"
}
```

ATTENTION
We recommend configuring your API response parsers to ignore any unrecognized fields. This is a common best practice that ensures your parsing logic remains forward-compatible with future changes and can easily handle dynamic fields supported by OpenRTB.

ID5 Signature

The ID5 Signature will be returned on every response from ID5 and contains all "user state" information necessary to support cross-domain reconciliation. As an example, this could include the following pieces of data:

- Original UID value (an encrypted first-party ID for this user on this domain/publisher)
- Cookie Birthdate
- Last Seen Timestamp (from this domain/publisher)
- Current ID5 ID value (encrypted)
- Link Type

> The Signature is used only by ID5 (it will be encrypted with a private key) and must be passed in every subsequent request to ID5.

Privacy API Opt-out

The Privacy API Opt-out endpoint allows authorized partners to opt out users from ID5 tracking using various privacy identifiers, including ID5 ID, MAIDs, email addresses, and hashed emails.

CTV Integration

09/23/2026 8:30 am EDT

Overview

We recommend that you deploy ID5 in CTV environments via a server to server integration. We do support alternative integration pathways depending on your requirements. If required, please reach out to your ID5 customer services representative.

To ensure accurate identity resolution, clients should provide signals, such as hashed emails in the request. We also expect you to store the 'signature' provided in our response on the user's device and provision it to us in future requests for the same user. This is integral to an optimal set up since it helps us re-identify consenting users even if their signals change as well as respect opt out requests.

Below are general instructions on how to retrieve an ID5 ID server-side for CTV inventory and make it available in bid requests to your demand partners. The auction should be delayed in order to retrieve the user's privacy preferences and the ID5 ID prior to sending out bid requests; the amount you delay may vary and so we recommend making it configurable for optimization purposes.

Process Flow

1. Check the stored cache for an available ID5 ID
    - ***IF*** there is no cached ID
      OR if the ID needs to be refreshed (we recommend a cache of max 8 hours)
      OR if the user's privacy preferences have changed:
        - Initialize a new HTTP POST request to the ID5 endpoint. If you previously had a stored response for a given user on their device, then subsequent requests must include the "signature" as part of the request to ID5;
        - Store the response from ID5 locally (either in local storage or in a database on the server);
    - ***ELSE*** if there is a valid, up-to-date value in cache:
        - Pull the latest ID5 ID response from the cache
2. Using the response from step 1 above (from the request or cache), prepare the bid request data.
    - Extract the **eid object(s)** from the ids structure (e.g., from `ids.id5id.eid` for the ID5 Universal ID, and any additional enriched IDs if services like Real-Time Bid Enrichment are enabled).
3. Include this data in each bid request to your demand partners as an eids array in the user object. The EID structure is already formatted for direct use in OpenRTB bid requests. For more information on how this is typically done, including handling provenance fields for enriched IDs, review the OpenRTB EID Enrichment Guide.

Caching the response

The response we return for the request will need to be cached as some of the response data needs to be used in future requests for your users. Use the `ts` parameter sent in the request in order to set a TTL for the cache refresh and a TTL for the cache deletion.

- The TTL for the cache refresh should be set to 8 hours. After the 8 hours, on the back of a user event, a new request to ID5 should be made and the cached data refreshed using the data from the API response.
- The TTL for the cache deletion should be set to 30 days. The TTL for the cache deletion should be refreshed with every response from the API. If there will not be a user event in the app, in the 30 days time window, the data should be deleted.

Building Server-Side Request

Server-Side Fetch endpoints for CTV

> **Global**
>
> https://api.id5-sync.com/gc/v1

Request Type

HTTP POST with JSON body.

Request Headers

`Content-Type: application/json`

Partner Number

The `PARTNER` in the request body will be an ID5-provided Partner Number. This value will be static for you once we set you up in our system. You may use the endpoint during testing with the Partner Number 173. If you haven't already been assigned a Partner Number, please contact us to request one.

Server-Side Request Parameters

Request Body

Name	Required	Type	Description	Format	Example
ts	x	string	Timestamp in form of string which extends the ISO-8601 extended offset date-time format to add the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
partner	x	integer	Partner Number provided by ID5	int32	173
appid	x	string	The store ID of the app in an app store (e.g., Apple App Store, Google Play).		
			Array of Identifiers For		

Name	Required	Type	Description	[{"id":"<ID>", "type":"<TYPE>"}, {<other identifiers if available>}]	[{"id":"bfd378ed-483e-4b11-be1e-12a599d042e0", "type":"DPID"}]
identifiers	when available	Identifier array	Identifier consists of `id` and `type`. Check Identifier object for supported types		
ver	x		Application version		
provider		string	Provider of the service (value to be provided by ID5)		
ip	x	string	IPv4 address closest to device	ipv4 dot-decimal	
ua	x	string	The User Agent of the device's default browser		Mozilla/5.0 (Linux; Android 11; moto e20 Build/RONS31.267-88-3; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/109.0.5414.118 Mobile Safari/537.36
hardware_code		string	The code pertaining to the device (i.e. TV model)		
signature	when available	string	The ID5 signature from a previous call, cached on the device or your server-side		
gdpr		integer	1 if gdpr applies to this request, 0 otherwise		
gdpr_consent		string	(where applicable) the TCF compliant consent string or see 'allowed_vendors'		
allowed_vendors		string array	ID5 Partner identifiers (starting with 'ID5-') or IAB Vendor IDs of vendors allowed to use the ID5ID		`['ID5-78','134']`
us_privacy		string	US Privacy Consent		`1YNN`
gpp_sid		string	The GPP section ID(s) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up		`6,7`

Name	Required	Type	Description	Format	Example
			a comma. More information in GPP documentation		
gpp_string		string	A valid IAB Global Privacy Platform consent string.		`DBABzw~1---~BAAAAAAAAQA.QA`
hem		string	sha256 hash of the cleansed e-mail address. Learn how to cleanse the data here	sha256	`f97ea86ed181d60b0ba62a30579f1e10ad71eaf21b548e173de75718065c533f`
phone		string	sha256 hash of the cleansed phone number	sha256	
idfv		string	Apple ID for Vendors	uuid	`3fa85f64-5717-4562-b3fc-2c963f66afa6`
ipv6		string	The IPV6 of the device	ipv6	`2001:0db8:5b96:0000:0000:426f:8e17:642a`
country		string	Country the user is located in	ISO-3166-1-alpha-2	`AD`
region		string	Region code using ISO-3166-2; 2-letter state code if USA	ISO-3166-2; 2-letter state code if USA	
city		string	City using United Nations Code for Trade & Transport Locations format	United Nations Code for Trade & Transport Locations format	
accept_language		string	A string representing languages accepted by end-user device, should be compatible with browser Accept-Language header	Accept-Language header format	`de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7`
segments		objects array	The segment ids a user may belong to and the destination platform that the segments should be pushed to. Only certain destination platforms are supported and there are backend configurations that need to be made in both ID5's and the destination platform's systems before this feature can be used. Please reach out to your ID5 representative or	Array of segment objects (see below Segment object)	`[{ "destination": "999", "ids": [ "12345", "67890" ] }]`

Name	Required	Type	Description	Format	Example
			information and to get started.		
ottid	deprecated	string	The user ID provided by the OTT/OEM. This will be the equivalent of an hard signal 3rd party cookie.		
ctvid	deprecated	string	The specific user id for the CTV ecosystem the user is observed in, as received by the CTV/OTT/Publisher provider		`00000000-34sd-gfss-sd54-12345678abcd`
ctvid_type	deprecated	string	CTV ID type		`tifa`
puid	deprecated	string	Partner specific user ID		

Identifier Object

Name	Required	Type	Description	Example
id	x	string	The IFA value, usually in the UUID format. Should be lowercased	`bfd378ed-483e-4b11-be1e-12a599d042e0`
type	x	string	The type of identifier provided, case insensitive. Supported values: AAID - Android TV RIDA - Roku AFAI - Amazon Fire IDFA - Apple tvOS MSAI - Xbox/Microsoft TIFA - Samsung (Tizen ID For Advertizing) LG - LG ID VIDA - Vizio Advertising ID PSID - PlayStation ID DPID - Generic device platform ID PPID - Publisher provided ID SSPID - SSP provided ID SESSIONID - Short-lived session ID UNKNOWN - any other value will be treated as UNKNOWN type	`DPID`

Segment Object

Name	Required	Type	Description	Example
destination	x	string	The destination platform. Should be the IAB Vendor ID	`'999'`
ids	x	string array	List of segment ids to add the user to	`[ '12345', '67890' ]`

Response Body

Name	Required	Type	Description	Format	Example
created_at	x	string	Timestamp in the form of a string that extends the ISO-8601 extended offset date-time format to include the time-zone	`yyyymmddThhmmss<ffffff>+|-hhmm`	`2013-02-01T12:52:34+09:00`
original_uid	x	string	A first-party user ID that remains stable for this user on the domain. It is intended for publisher reference only and should not be shared with other partners. The value will be encrypted and may change periodically, even for the same user on the same domain, while the underlying value remains stable. If ID5 lacks consent, then the value will be "0".		
universal_uid	x	string	This is the UID used for sharing with other parties. The value will be encrypted and will change periodically, even for the same user on the same domain. If ID5 does not have consent, then the value will be "0".		
ids	x	object	Container for all identity providers returned by ID5 in OpenRTB-compatible format See the ids object below.		
privacy	x	object	An object containing privacy information See the privacy object below.		
signature	x	string	The ID5 Signature is a string that must be stored by the caller (either cached on the device or your server-side) and sent back to ID5 with all future requests. See below for details about the Signature.		

Name	Required	Type	Description	Format	Example
ext	x	object	See the extension object below.		
gpId		string	Guarded Publisher ID (see more here)		
enrichment		object	Present only when Real Time Bid Enrichment is enabled for given request. See Enrichment object		

IDS Object

The `ids object` provides identifier data for the ID5 ID and any other Extended Identifiers (EIDs) received from ID5.

Each sub-object is named (e.g., `id5id` for the ID5 Universal ID) and contains an OpenRTB-compatible EID structure. Additional third-party identifiers may be included based on enabled services, such as Real-Time Bid Enrichment.

For example, the **ids object** might include:

- `id5id` : The ID5 Universal ID.
- `euid` : European Unified ID.
- `id5-[provider]` Other providers like PubMatic, Magnite, Xandr, TTD, etc.
  - For example: `id5-ttd` : The Trade Desk ID (enriched via ID5)

Each identifier contains an `eid object` that mirrors the structure of OpenRTB's eids field, enabling direct use in bid requests. For enriched IDs, provenance fields such as **mm** (match method), **matcher**, and **inserter** may be added to improve transparency and compliance with IAB best practices.

Name	Required	Type	Description	Format	Example
id5id	x	object	The container for the ID5 Universal ID. For details, see the EID object below.		

EID Object

Name	Required	Type	Description	Format	Example
source	x	string	Source or technology provider responsible for the set of included IDs. Expressed as a top-level domain.		id5-sync.com
uids	x	object array	An array of extended ID UID objects from the given source. Refer to Object: UIDS.		
mm	For enriched IDs	integer	The match method used by the matcher to establish the ID connection (e.g., exact match, probabilistic, or IP-based). This promotes transparency in identity resolution processes. Refer to AdCOM 1.0 Matching Methods	enum	2
matcher	For enriched IDs	string	The domain of the technology provider responsible for matching or bridging the ID, as defined by the mm value.		id5-sync.com

Name	Required	Type	Description	Format	Example
inserter	For enriched IDs	string	The canonical domain name of the entity (e.g., publisher, SSP, or wrapper) that added the EID to the bid request.		publisher-domain.com

UIDS Object

Name	Required	Type	Description	Format	Example
id	x	string	The identifier for the user. For ID5, the value equals the response field: `universal_uid`		
atype	x	integer	Type of user agent the match is from. Refer to AdCOM Agent Types.	enum	1
ext		object	An extension object for passing additional metadata about this specific ID. Further information on ID5 metadata is available here.		

Privacy Object

Name	Required	Type	Description	Format	Example
jurisdiction	x	string	The legal jurisdiction applicable to the request (e.g. "gdpr", "ccpa", etc), based on the location the request was made from	enum	gdpr
id5_consent	x	boolean	Indicates if ID5 had proper consent on the request to process the user's personal data. See below for more information about how ID5 handles requests with and without consent.		

ext Object

Name	Type	Description	Example
linktype	integer	See details here	1

Enrichment Object

Name	Type	Description	Example
enrichment_selected	boolean	False if request was selected to be in enrichment control group, true if it was selected to be enriched (normal group)	false
enriched	boolean	true if request was selected to be enriched AND at least one enriched ID was found and returned for given request. False otherwise	false

Invalid Request Response

Name	Type	Description	Example
code	string	A code that univocally identifies the error	- partner_id_invalid - sha256_length_invalid - request_format_invalid - user_object_invalid - internal_id5_error

Name	Type	Description	Example
message	string	Human-readable message about the error	- unknown partner - failed to parse json POST - missing ua - missing ts - ts older than allowed - invalid ts format - Fetch of ID5 ID disabled. Please contact ID5 to enable it. - Request from a country that is disallowed. Please contact ID5 to enable it.
type	string	A code that identifies the class of error	- validation_error - invalid_request_error - authentication_error

Example Request

```
curl "https://api.id5-sync.com/gc/v1" -X POST -H 'Content-Type: application/json' -d '{
   "ts": "2023-09-19T12:32:34+02:00",
   "partner": 173,
   "ver": "1.0",
   "ip": "4.5.6.7",
   "ua": "Mozilla/5.0 (Linux; Android 11; moto e20 Build/RONS31.267-88-3; wv) AppleWebKit/537.36 (KHTML, like Ged
   "appid": "ABCDE12345.com.example.foo",
   "gpp_sid": "2",
   "gpp_string": "DBAA",
   "accept_language": "de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7",
   "identifiers": [{"id": "bfd378ed-483e-4b11-be1e-12a599c042e0", "type":"DPID"},{"id": "ID-12345", "type":"PPID"}]
}'
```

Example Valid Response

```
{
   "created_at": "2025-11-14T16:12:20.968Z",
   "original_uid": "ID5*KDdqfh7p46U_Eb_xD8-9.WYkX2xsHEQqpJP2YNzKqC7__",
   "privacy": {
      "id5_consent": true,
      "jurisdiction": "gdpr"
   },
   "signature": "ID5_AoPiwhPFUccYOQLCwfbQTEbMTk85rQD10hAZYixkNIZeozXos46w4uqpJqq",
   "universal_uid": "ID5*1Ls3HtzGAlqquMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
   "ext": {
      "linkType": 2
   },
   "ids": {
      "id5id": {
         "eid": {
            "source": "id5-sync.com",
            "uids": [
               {
                  "atype": 1,
                  "ext": {
                     "linkType": 2,
                     "pba": "Fnqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWJP200AngDPxw=="
```

```
            },
            "id": "ID5*1Ls3HtzGAIdqduMjJmL2CcyuGOw9JArVgXc_LS34MC7__"
          }
        ]
      }
    },
    "euid": {
      "eid": {
        "source": "euid.eu",
        "uids": [
          {
            "atype": 3,
            "ext": {
              "provider": "id5-sync.com"
            },
            "id": "E4.AAAAAezwMkyH6T4wjwiMRLR7C7l77M2Vyl8q_pQSzPLdlsiffl86arrL3FKC4T"
          }
        ]
      }
    },
    "id5-ttd": {
      "eid": {
        "inserter": "publisher-domain.com",
        "matcher": "id5-sync.com",
        "mm": 2,
        "source": "adserver.org",
        "uids": [
          {
            "atype": 1,
            "ext": {
              "provider": "id5-sync.com",
              "rtipartner": "TDID"
            },
            "id": "09d89c84-2e59-4c80-9c80-09db5f901234"
          }
        ]
      }
    }
  }
}
```


Example Invalid Response


```
{
 "code": "partner_id_invalid",
 "message": "unknown partner",
 "type": "invalid_request_error"
}
```


ATTENTION

We recommend configuring your API response parsers to ignore any unrecognized fields. This is a common best practice that ensures your parsing logic remains forward-compatible with future changes and can easily handle dynamic fields supported by OpenRTB.

ID5 Signature

The ID5 Signature will be returned on every response from ID5 and contains all "user state" information necessary to support cross-domain reconciliation. As an example, this could include the following pieces of data:

- Original UID value (an encrypted first party ID for this user on this domain/publisher)
- Cookie Birthdate
- Last Seen Timestamp (from this domain/publisher)
- Current ID5 ID value (encrypted)
- Link Type

> *The Signature is used only by ID5 (it will be encrypted with a private key) and must be passed in every subsequent request to ID5.*

Passing Signals to ID5

10/15/2026 6:29 am EDT

What is Partner Data?

Partner Data includes all signals provided by partners that assist ID5 in accurately identifying users. By sharing additional information, you improve the precision, stability, and effectiveness of user identification. To ensure the highest-quality ID5 ID and unlock the full potential of your addressable audience, we recommend providing as many user signals as possible via the Partner Data (PD) string.

Why Share More Signals?

With browsers increasingly obfuscating signals to protect user privacy, it is vital to proactively share robust data to maintain addressability and maximize user recognition.

Recommended Next Steps:

- **Share Core Signals**: Ensure IP address and user agent are always included in the PD string.
- **Include Additional Signals**: Contribute hashed emails, mobile ad IDs, or other identifiers for users who have provided consent. These signals help strengthen user identity resolution and deliver better outcomes across your properties.
- **Explore Custom Signals**: If you have specific signals that could enhance ID5's ability to reconcile users across your properties, please reach out to your ID5 representative. We will provide tailored advice on how to integrate them effectively.

How is Partner Data Used?

Signals passed in the ID5 ID request are used to inform ID5 ID connections across domains. "Hard signals", such as hashed email addresses, take priority for cross-domain linking purposes, and help to train ID5's probabilistic algorithm. Publisher provided signals, such as IP address and user agent, may be used in ID5's probabilistic algorithm when ID5 determines that they may be more accurate than those that ID5 can source directly from the HTTP request. ID5 requires sha256 hashing & URL-safe base64 encoding to ensure that personally-identifiable information isn't transmitted in the ID5 ID call.

Supported Partner Data Keys

Key	Description	Required	Example
0	Other	Optional	
1	SHA256 Hashed Email[1]	Recommended	`f97ea86ed181d60b0ba62a30579f1e10ad71eaf21b548e173de75718065c5`
2	SHA256 Hashed Phone Number[2]	Recommended	`f687e4a1be889a45c13e417f77cb9bff9c67f46e35fd68d936e6b01a933ecbc`
3	Cross-Partner User ID Value	Optional	`e3206fbc-b2f1-11ed-afa1-0242ac120002` *(a user id that could be used across ID5 Partner Numbers / Accounts)*
4	Cross-Partner User ID Source *(hard-coded value will be provided by ID5)*	Optional	`super-pub-identifier`
5	Partner-Specific User ID Value	Optional	`e3206fbc-b2f1-11ed-afa1-0242ac120002` *(i.e. a user id that is specific to a single ID5 Partner Number / Account)*
6	Apple ID for Advertising (IDFA) *(lowercased)*	MAIDs should only be sent via the pd string for mobile web traffic where available. Please use our mobile app specific endpoint for in app traffic. Documentation here	`ea7583cd-a667-48bc-b806-42ecb2b48606`

Key	Description	Required	Example
7	Google Advertising ID (GAID) *(lowercased)*	MAIDs should only be sent via the pd string for mobile web traffic where available. Please use our mobile app specific endpoint for in app traffic. Documentation here	`cdda802e-fb9c-47ad-9866-0794d394c912`
9	Domain	Recommended	`id5.io`
10	IPv4 Address of the end-user's device	Recommended	`77.99.190.227`
11	IPv6 Address of the end-user's device	Optional	`2001:0db8:85a3:000:000:8a2e:0370:7334`
12	User Agent String of the end-user's device	Recommended	`Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, lik Gecko) Chrome/94.0.4606.81 Safari/537.36`
13	Is Burner Email[3]	Optional	`false`
14	Apple ID for Vendor (IDFV) *(lowercased)*	Recommended (when available)	`f325g3gb-12fc-352f-c6c3-dz52f0f690d8`
15	**DEPRECATED** ~~CTV ID~~	CTV IDs can be sent when integrating our ctv specific endpoint for ctv traffic. Documentation here	
16	**DEPRECATED** ~~CTV ID Type~~	CTV IDs can be sent when integrating our ctv specific endpoint for ctv traffic. Documentation here	
17	An IAB TechLab Tokenization Framework token (e.g. uid2)	Optional	`AdvertisingTokenmZ4dZgeuXXl6DhoXqbRXQbHlHhA96l....`

> **ATTENTION**
> If any of the signals is not available at the moment of the PD string creation, please leave key out of the PD string. Don't include keys for which you don't have the values or include default values you set.

- [1] See below for instructions on "Normalizing Emails Prior to Hashing"
- [2] See below for instructions on "Normalizing Phone Numbers Prior to Hashing"
- [3] A flag for whether the hashed email provided is a burner email (boolean). As emails are provided to ID5 in hashed form, ID5 is unable to determine whether provided hashed emails are "burner" emails. Publishers can use this flag to signal to ID5 whether the provided email should be treated as a "burner" email. If you're unsure how to determine if an email is a burner, we recommend you simply send `true` for all icloud email addresses

Deriving the Partner Data (pd) Value

The general procedure to build a PD string is:

1. Normalize any values that need to be hashed, like emails, (see details below for how to) and then `sha256` hash them
2. URL-encode each value using UTF-8 charset (according to RFC 3986, or at the very least, in JS using a function like `encodeURIComponent`)

3. Create the raw pd string containing the keys and the URL-encoded value, using querystring formatting (order does not matter)
    - e.g. `<key1>` = `<value1>` & `<key2>` = `<value2>` ...
4. URL-safe base64 (RFC 4648) the entire raw PD string (using a function in JS like `btoa()`)
5. Once you have the encoded PD string, it can be passed into the `pd` field in any of our integrations (i.e.: PD parameter for our Prebid integration or JS API integration)

Normalizing Hashed Inputs

Because ID5 doesn't see the original raw values for some of the signals we accept (e.g. hashed emails, hashed phone numbers), you will need to normalize them first. Normalizing the raw values before hashing them ensures that the signals sent by you and other partners will always be the same, ensuring the ID5 IDs can be properly linked.

Normalizing Emails Prior to Hashing

Prior to hashing an email address, you must normalize the string by removing unnecessary characters:

1. Remove leading and trailing spaces
2. Convert all ASCII characters to lowercase
3. Please find below an example with email accounts ending in `@gmail.com`. You can apply this method to all the email accounts:
    a. Remove `.` (ASCII code 46) from the username of the email address
        - e.g. `jane.smith@gmail.com` normalizes to `janesmith@gmail.com`
    b. Remove `+` (ASCII code 43) and all subsequent characters from the username of the email address
        - e.g. `janesmith+test@gmail.com` normalizes to `janesmith@gmail.com`

Normalizing Phone Numbers Prior to Hashing

Phone numbers should be normalized to the E.164 format, which is an international phone number format to ensure global consistency and uniqueness. When normalizing with the E.164 format, the result should be no more than 15 digits in length, prior to hashing.

1. Remove all spaces, hyphens, parentheses, or other special characters
2. Format the phone number as follows: `[+][country code][subscriber number including area code]`
    - e.g. `+111 22 333-44-555` normalizes to `+1112233344555`
    - e.g. `+1 (222) 333-4444` normalizes to `+12223334444`

Example

Here is an example to show you how to generate a PD string, given you have the following raw signals to share:

- Email = `Jane.Smith+test@gmail.com`
- IPv4 = `77.99.190.227`
- IPv6 = `2001:0db8:85a3:000:000:8a2e:0370:7334`
- UA = `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36`

PD Creation Steps

Step 1: Normalize and Hash Inputs

In this example, we only need to normalize the email address. Following the instructions above ("Normalize Emails Prior to Hashing"), the result is `janesmith@gmail.com`, which when sha256 hashed, becomes `9a0f2978ccf8af196d24f627062a2d4054c9da92e9d998a514bda4a01a3cfec7`

Step 2: URL-encode the Values

- **Email (key 1)**

```
9a0f2978ccf8af196d24f627062a2d4054c9da92e9d998a514bda4a01a3cfec7
```

- **IPv4 (key 10)**

```
77.99.190.227
```

- **IPv6 (key 11)**

```
2001%3A0db8%3A85a3%3A000%3A000%3A8a2e%3A0370%3A7334
```

- **UA (key 12)**

```
Mozilla%2F5.0%20(Windows%20NT%2010.0%3B%20Win64%3B%20x64)%20AppleWebKit%2F537.36%20(KHTML%2C%20like%20
```

Step 3: Create Raw PD String

```
1=9a0f2978ccf8af196d24f627062a2d4054c9da92e9d998a514bda4a01a3cfec7&10=77.99.190.227&11=2001%3A0db8%3A85a3%3A00
```

Step 4: URL-safe base64 Raw PD String

```
MT05YTBmMjk3OGNjZjhhZjE5NmQyNGY2MjcwNjJmMmQ0MDU0YzlkYTkyZTlkOTk4YTUxNGJkYTRhMDFhM2NmZWM3JjEwPTc3Ljk5LjE5MC4y
```

Sample Javascript Implementation

Here is just one approach to generate the PD string, but there are numerous other ways to accomplish the same result. Here we're using the same inputs as the example above.

```
// get these values from your webserver or the browser's apis
const ipv4  = '77.99.190.227',
      ipv6  = '2001:0db8:85a3:000:000:8a2e:0370:7334',
      ua    = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36',
      email = 'jane.Smith+test@domain.com'; // alternatively, normalize and then sha256 server-side and return the hashed value

// normalize the email string with normalizeEmail() from  https://github.com/validatorjs/validator.js/blob/master/src/lib/normalizeEmail.js
const cleansedEmail = normalizeEmail(email);

// set the keys and URL-encode each value
const pdKeys = {
      1: CryptoJS.SHA256(cleansedEmail), // requires the crypto-js package https://www.npmjs.com/package/crypto-js
      10: encodeURIComponent(ipv4),
      11: encodeURIComponent(ipv6),
      12: encodeURIComponent(ua),
};

// convert the key/values into a querystring format
const pdRaw = Object.keys(pdKeys).map(key => key + '=' + pdKeys[key]).join('&');

// base64 encode the raw string; this is the final value you can pass into the pd field
const pdString = btoa(pdRaw);
```

Sample Prebid.js Implementation

The Partner Data (PD) string you generate following the instructions in this guide can be directly integrated into various ID5 implementations, including the Prebid.js User ID Module. This allows you to pass additional user signals to ID5 during the ID request process, enhancing user identification and addressability within Prebid auctions.

In Prebid.js, the PD string is supplied via the **pd** parameter in the **userIds** configuration for the **id5Id submodule**. This ensures that the signals are securely transmitted and used by ID5 to generate or refresh the ID5 ID before bids are requested.

Quick reference

```
pbjs.setConfig({
  userSync: {
    userIds: [{
      name: "id5Id",
      params: {
        partner: 173, // Replace with your ID5 Partner Number
        pd: "MT05YTBmMjk3OGNjZjhhZjE5NmQyNGY2MjcwNjJhMmQ0MDU0YzlkYTkyZTlkOTk4YTUxNGJkYTRhMDFhM2NmZWM3JjEwPTc3LjK5
        // Other optional params...
      },
      storage: {
        type: "html5",
        name: "id5id",
        expires: 90,
        refreshInSeconds: 7200
      }
    }],
    auctionDelay: 250
  }
});
```

ATTENTION

For full details and examples on setting up the **Prebid.js User ID Module**, refer to the ID5 Prebid User ID Module wiki

Adobe Experience Cloud

01/09/2025 4:14 pm EST

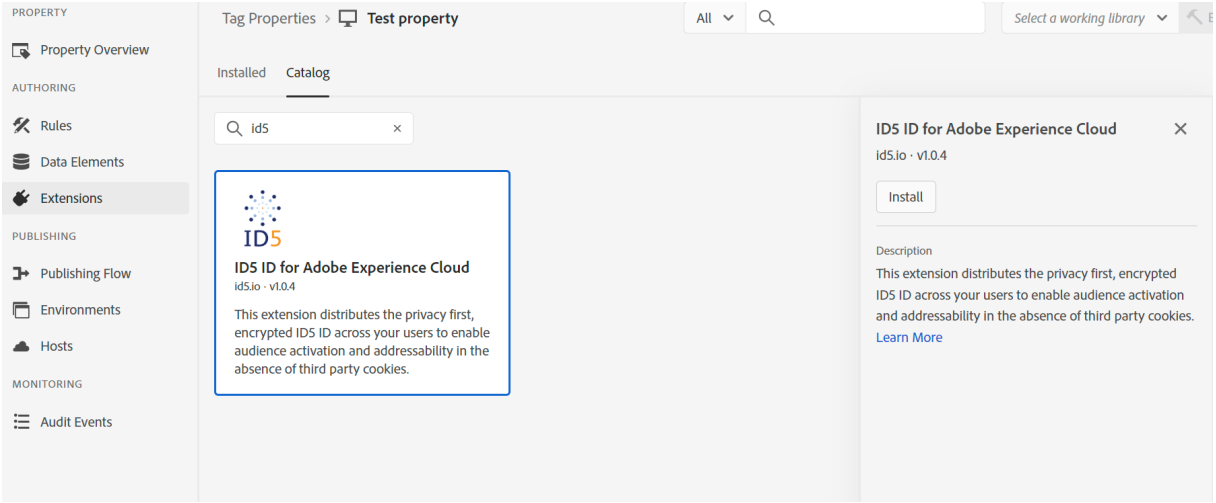
Adobe Experience Cloud

Overview

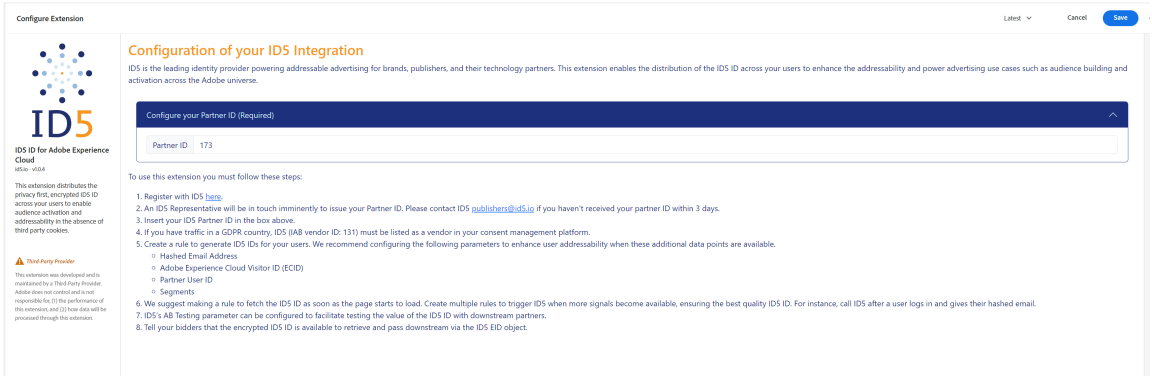
Partners using Adobe Advertising, Adobe Audience Manager and Adobe Real-Time CDP can utilize the 'ID5 extension for Adobe Experience Cloud' in the Adobe Exchange marketplace to seamlessly deploy the ID5 ID across their media properties. By implementing the ID5 extension, media owners will be able to seamlessly transform signals from both authenticated and unauthenticated site visitors into precise, universal, and consented ID5 IDs and securely distribute them to authorized advertising partners.

Integration Steps

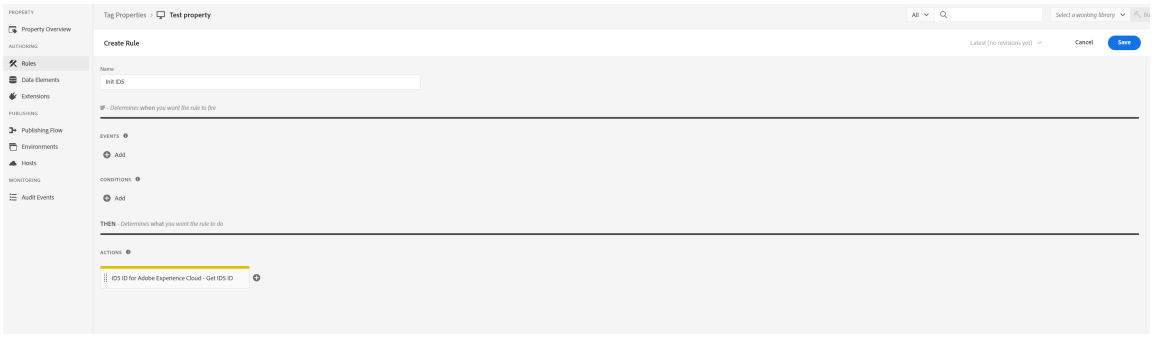
1. Install the 'ID5 ID for Adobe Experience Cloud' within the Adobe Experience Platform Data Collection.



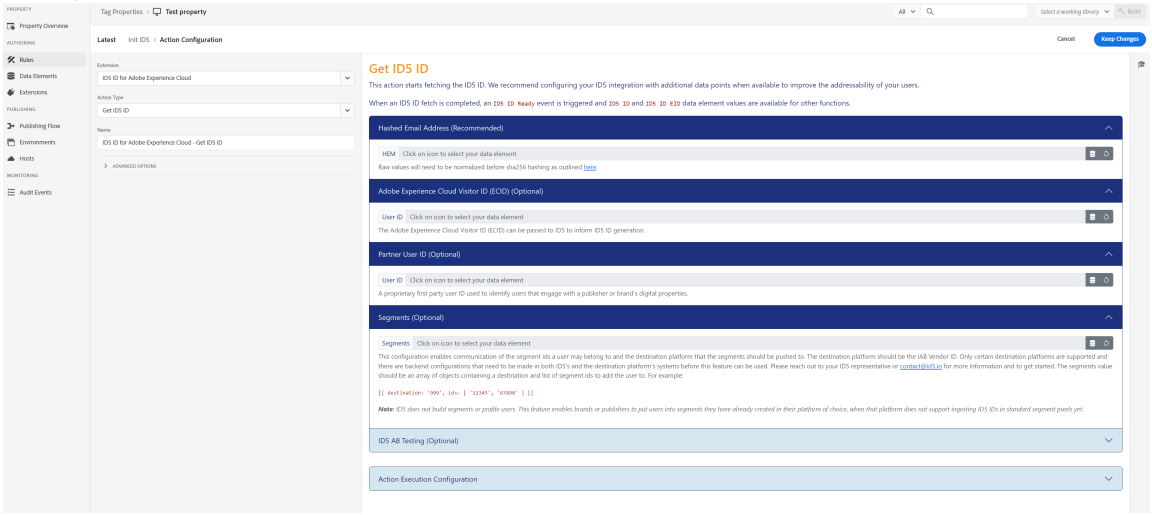
2. Sign Up with ID5 using the following link https://id5.io/universal-id/adobe-experience-platform/
3. An ID5 Representative will be in touch with your ID5 Partner ID.
4. Configure the ID5 integration in the 'ID5 for Adobe Experience Cloud' application including:
    - Input your ID5 Partner ID



    - Create a rule with GET ID5 ID action as soon as the page loads as well as additional rules to call ID5 when additional signals like hashed email become available.



- Configure the relevant parameters to ensure your integration passes hashed emails (HEMS), Adobe Experience Cloud Visitor ID (ECID), Partner User ID and Segments where applicable. Optionally configure ID5's AB Testing parameter to facilitate testing the value of the ID5 ID. A 95:5 split is sufficient.



5. Upon implementation, the encrypted ID5 ID will be accessible through the ID5ID and ID5ID EID DataElements. Adobe provides several storage options for DataElement values, such as storing them in a JavaScript variable or localStorage. Developers should choose one of these options and pass the ID5 ID to bidders using a custom script or another appropriate method. If a bidder has its own extension within the Adobe platform, the encrypted ID5 ID can also be passed by implementing a separate rule. Bidders should be notified that the encrypted ID5 ID is available for retrieval and should be passed downstream via the EID Object.

Amazon (APS) Integration

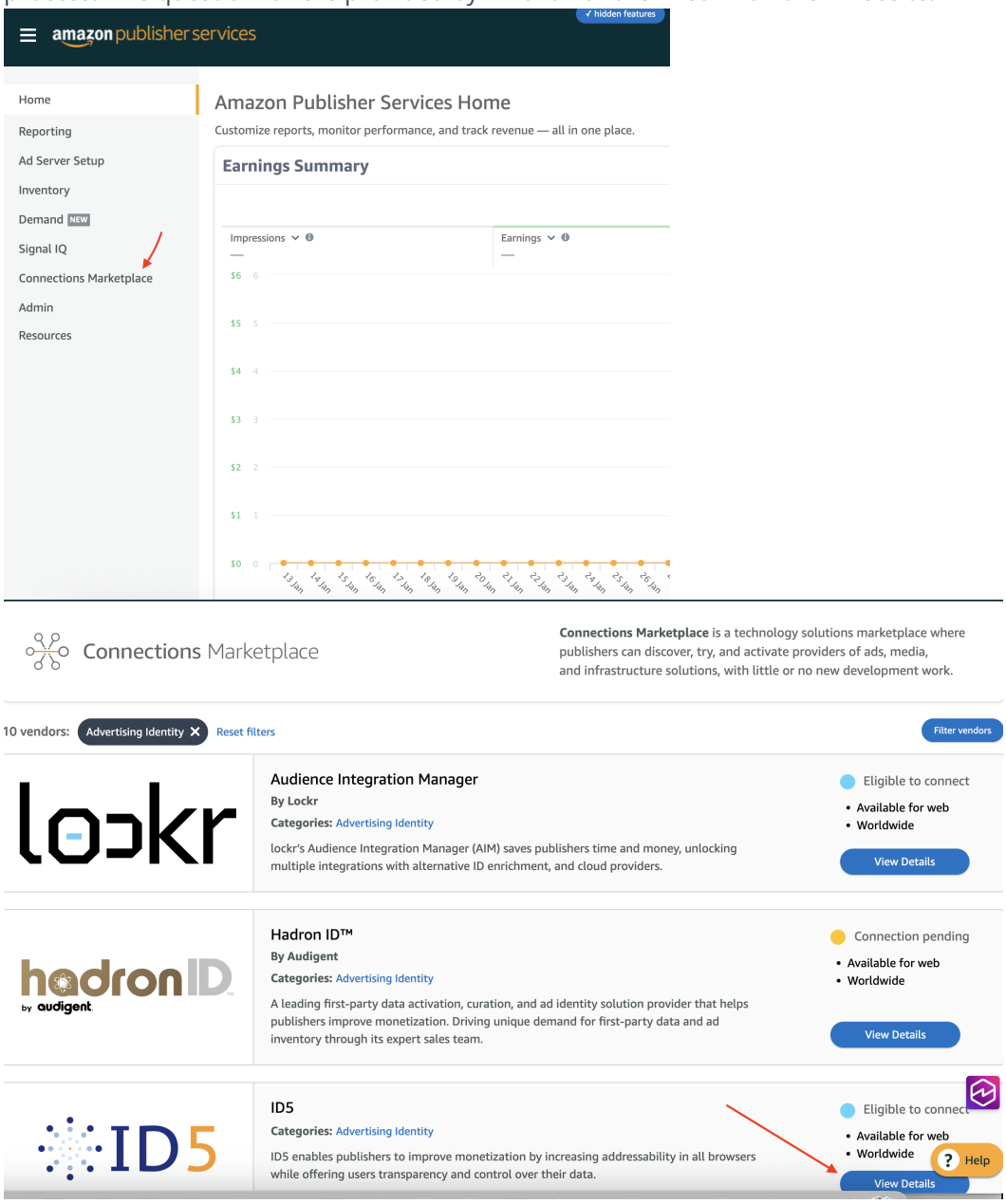
04/08/2026 10:33 am EDT

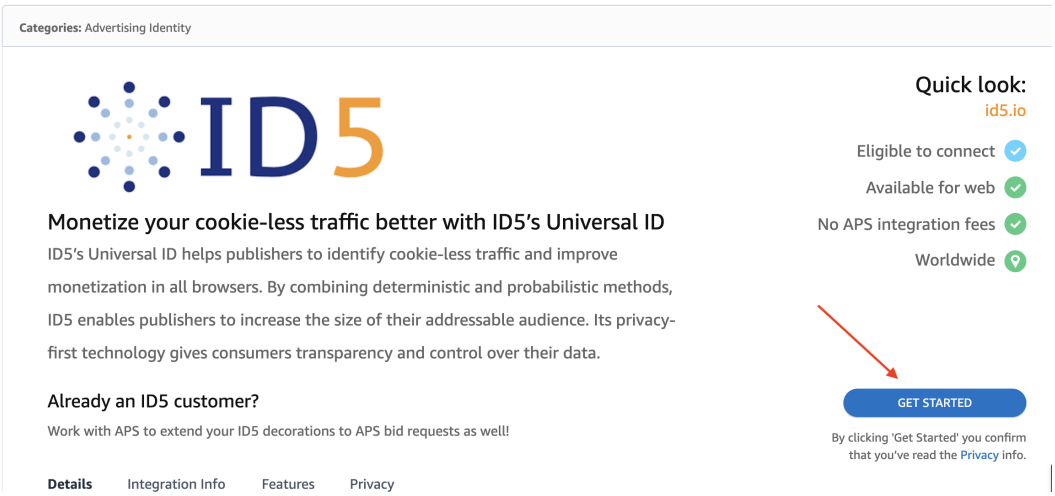
Publisher Integration of the ID5 ID via Amazon Publisher Services

Enabling ID5 in Web Environments

Amazon's APS is the leading suite of cloud services that helps publishers build, monetize, and grow their digital media business. Read below for a step-by-step guide that takes you through the full integration process.

1. **Log into your APS Publisher UI -** Navigate to the Connections Marketplace tab and Explore Services. Select “View Details” for ID5 and then select “Get Started” to follow the questionnaire submission process. This questionnaire is provided by Amazon and is filled in on their website.





2. **Activate ID5 -** Submit the questionnaire to APS to complete your registration with ID5. Part of the activation you submitted through the APS dashboard opted you into ID5's terms and conditions. By accepting ID5's terms, you agree to maintain clear, comprehensive, and accurate privacy notices as stated in our Privacy Policy Requirements.

3. **Add to CMP** - Add ID5 as an approved vendor to your Consent Management Platform (CMP). Our vendor ID is `131`.

4. **Account setup** - Following the creation of your ID5 account, you will now have the opportunity to customize your settings. These settings are updated within Amazon TAM.

5. **Your service begins** - The integration is now complete and ready for action.

Enabling ID5 in CTV Environments

Amazon supports the standard OpenRTB framework for transmitting third-party identifiers, allowing publishers to append the ID5 ID in the user.eids field (or user.ext.eids in OpenRTB 2.5) before sending the bid request. Publishers can use the dedicated S2S endpoint for CTV to generate and retrieve the ID5 ID, and then pass that ID5 ID to APS within their initial OpenRTB ad request using the standard Extended ID (EID) object.

Please see the below for sample bid requests:

OpenRTB 2.6

```
"user": {
  "eids": [
    {
      "source": "id5-sync.com",
      "uids": [
        {
          "id": "ID5_TOKEN_VALUE",
          "ext": {
            "linkType": 2
          }
        }
      ]
    }
  ]
}
```

OpenRTB 2.5

```
"user": {
  "ext": {
    "eids": [
      {
        "source": "id5-sync.com",
        "uids": [
          {
            "id": "ID5_TOKEN_VALUE",
            "ext": {
              "linkType": 2
            }
          }
        ]
      }]}},
```

Name	Required	Type	Description	Example
source	x	string	URL of identity provider	id5-sync.com
id	x	string	ID sent to publisher by identity provider	ID5*abc

Google Secure Signals

03/27/2026 6:36 am EDT

What are Google Secure Signals?

Google Secure Signals enables publishers to deploy the ID5 ID and make it available as a secure signal on RTB requests through Authorized Buyers, Open Bidding and SDK bidding. Signals will only be generated and shared at the Publishers explicit instruction and only with the bidders the publisher has permitted to receive the signals. Publishers benefit by bringing their identity strategy to their Google tech stack with a lightweight integration, resulting in more addressable inventory and improved monetization.

Web Integration

Bidders must choose whether they want to receive a signal (from any publisher) when it's available. We recommend reaching out to your main bidders and letting them know to pick up the encrypted ID5 ID.

You can read more about Google's Secure Signals here.

Steps

1. For publishers to integrate with ID5 through Google Secure Signals, a Partner Number will be needed from ID5 Partner Number. This can be requested by signing up with ID5 on our website).
2. Sign in to Google Ad Manager.
3. Click **Admin**, then Global settings, and then Ad exchange account settings.
4. In the "**Secure signal sharing**" section, click the toggle on to allow secure signal sharing and click Save.
5. Click **Inventory**, then **Secure Signals**, this will show you a list of secure signals you can share.
6. Toggle ID5 **on**
7. **For web integrations, there are a few integration options:**

- **a)** Deploy the signal collection script for an existing Prebid UserID module. Under "Signal collection deployment", select Prebid UserID module. This will enable Google to collect the ID5 ID from your current Prebid UserID deployment of the ID5 ID. If you haven't done so already, to receive the most valuable ID5 IDs, ensure you are passing ID5 additional signals, such as hashed email.
- **b)** Google can deploy the signal collection script for you by selecting **Google deploy** under "**Signal collection deployment**". For this to work, add the following code to the `<head>` part of your

website:

```
window.ID5EspConfig = {
  partnerId: 123 // 123 should be modified with your own partnerId
};
```

Note, this integration does not support the provision of additional signals like hashed emails to ID5 and, therefore, may reduce ID precision.

- **c)** Under "**Signal collection deployment**", select the "**Publisher deploy**" option, deploy the ID5 JS API with GSS provider enabled. The benefit of this option is that you can provide ID5 additional signals, such as hashed email, resulting in more valuable ID5 IDs.

9. Under "**Signal collection deployment**", select Share to share secure signals on **non-personalized (NPA) ad requests**.
10. To share secure signals with bidders, click **Delivery**, then **Demand channel settings**.
11. Under the "**Default settings**" tab, click **Secure signal sharing**.
12. Turn **on** the toggle for the desired demand channels. Bidders can now receive the secure signals you share with them.

In-App / CTV Integration

Server-to-Server (S2S) / CTV

Publishers integrating with ID5 via S2S can make the ID5 ID available as a Google Secure Signal, provided they build and manage their own ad requests. In this case, the ID5 ID should be structured as follows and passed via the ssj parameter as described in Google's instructions:

```
{
    "source": "id5-sync.com",
    "data": "ID5*xxxxxxxx", // ID5ID received
    "isPubCreated": false
}
```



This path is not applicable to publishers using the GMA SDK, as the SDK handles ad request construction directly and requires an adapter to pass third-party signals to Secure Signals.

GMA SDK / CTV

For publishers using the GMA SDK, an adapter is required to pass ID5 signals through Google Secure Signals. ID5 is currently developing mobile and CTV SDKs with accompanying adapters. Please contact your ID5 representative if you wish to be an early adopter.

Permutive

06/04/2026 2:38 pm EDT

Publisher Integration of the ID5 ID via Permutive

As identity infrastructure consolidates inside walled gardens, the ID5 + Permutive integration gives publishers an independent path to full-inventory addressability — built on first-party signals, not borrowed third-party infrastructure. Publishers can activate ID5's Universal ID directly through their existing Permutive instance, with minimal technical lift, and improve programmatic monetization across the bidstream.

1) Initial Contact

Contact your assigned Permutive CSM to initiate the ID5 onboarding process.

2) Pre-Activation Requirements

Before activation, confirm the publisher has the following in place:

A. ID5 Direct Agreement (required)

- For Publishers going Permutive direct: Fill out the sign-up form and someone from the ID5 team will reach out to get the agreement completed.
- Publishers going ID5 direct will work with ID5 sales directly to get agreement completed. Reach out to publishers@id5.io for more information.

B. ID5 Partner ID

- Once agreement is completed, the corresponding ID5 partnerships manager will generate the ID5 partner ID. The partner ID will be required to enable ID5 through Permutive.

C. Consent management / privacy configuration

- ID5 added as an approved vendor in the publisher's CMP / TCF vendor list (vendor ID 131)

⚠ If ID5 is not properly configured in your CMP, the integration will not generate identifiers, even if enabled in Permutive.

D. Prebid integration

- Enable Prebid Identity Manager Module

1. Check your current Prebid configuration or contact your Prebid implementation team

2. If the module is not already included, rebuild Prebid with the permutiveIdentityManagerIdSystem module. For detailed Prebid configuration instructions including the Identity Manager module setup, see the Prebid integration guide.

3. Deploy the updated Prebid build to your website

E. Permutive Web SDK

- Activate the ID5 integration in Permutive:
    - Go to Identity → Catalog in Permutive dashboard
    - Find ID5 in the identity provider catalog and select Connect
    - Enter the publisher's ID5 Partner ID and save
    - The Permutive SDK will begin retrieving ID5 identifiers on the next SDK build / release cycle
- Optional
    - APS support is available on request via the publisher's Permutive CSM.
    - For more information, reference the Permutive + ID5 integration page here.

Signal Obfuscation

06/04/2025 3:27 pm EDT

How Publishers Can Future-Proof Their Addressability Strategy

Browsers and Operating Systems are taking incremental steps to obfuscate identification signals from third parties. From the recent release of Apple's iOS15, which introduced features including Hide my IP and Private Relay, to Google's announcement that it is planning to reduce the granularity of information presented in user-agent strings on its Chrome browser.

While the removal of traditional identifiers and limiting of signals informing identity resolutions should guarantee higher protection of people's privacy and publisher data, it also poses a challenge. Many players currently rely on these signals for cross-domain and cross-device reconciliation. Apple's recent release of iOS15 has resulted in a rising number of requests where a user's true IP, which contributes to ID creation, is not available to ad tech platforms.

How ID5 Is Mitigating These Challenges

Our mission at ID5 is to help media owners grow sustainable revenue. We have been working alongside publishers and their technology partners to facilitate maximum addressability, monetization, and yield in a privacy-first manner since 2017. We prize our role as a trusted partner to publishers and we want to continue to empower them to address users effectively and minimize the impact of current and upcoming signal restrictions. This is why we have developed an action plan to ensure that our publishers can benefit from a strong, reliable and privacy-first ID.

Supporting additional signals

ID5 supports additional key-values in the pd field for all of its client-side integrations, including Prebid.js, Amazon's APS, and Google ESP. These key-values include:

- IP address
- Domain
- URL
- User agent string

These signals, in addition to the other signals we support, play a vital role in facilitating the reliable identification of users. We strongly encourage publishers to pass these signals to ID5 in the pd string, as detailed in the **Passing Partner Data to ID5** documentation.

By leveraging ID5's ability to ingest additional signals, publishers can benefit from an ID offering maximum accuracy and scale, in addition to standing the test of time if further signal restrictions for ad tech platforms come into play.

TrueLink Integration

03/14/2025 8:42 am EDT

Introduction to TrueLink

TrueLink is an advanced client-side integration method designed to enhance your existing ID5 setup, whether you're using the standard ID5 JS API or a Prebid implementation (version 9.2.0 and above). By implementing TrueLink, ID5 generates a persistent cross-domain signal for individual users within a specific browser, fully independent of third-party cookies. This is achieved by redirecting users through an ID5-operated domain and storing the ID as a first-party cookie. ID5 leverages this deterministic signal to enhance user identification and produce a high-quality ID5 ID.

Since the ID5 ID is informed by signals derived from the TrueLink integration, publishers may want to optionally want to access a publisher first-party user identifier called the ID5 Guarded Publisher ID (GPID). The GPID is a publisher specific version of the ID5 ID which remains unique for a user across their owned and operated properties within a given browser environment. It's quality is implicitly linked to the provision of signals such as hashed emails and TrueLink signals. The GPID can be used to facilitate use cases such as cross-domain audience building and activation, all without relying on third-party cookies. It may also be used as a PPID within Google Ad Manager.

During the beta phase and for a limited time, partners with owned and operated inventory integrating TrueLink can optionally access and use a partner-specific version of the TrueLink ID as a PPID. However, please note that since TrueLink is currently supported only in web environments, its coverage is more limited compared to the GPID.

Integrating TrueLink

Library Source

The TrueLink bootstrap JavaScript library source code is accessible via a private GitLab repository. To obtain access, kindly contact your ID5 Representative and provide the email addresses of the users who require access to the repository.

We compile a minified, public version of the library directly from the repository and host it on our CDN at the following location: https://cdn.id5-sync.com/bootstrap/id5-bootstrap.js.

You should not use the library directly from our CDN. The TrueLink bootstrap library needs to be served through the domain of the webpage.

Integration Guidelines

A TrueLink integration cannot be used independently of an ID5 integration via JS API or Prebid.

1. The TrueLink bootstrap library must be served through a first-party domain. So if the webpage is `your-domain.com` or `sub.your-domain.com` the bootstrap library should be made available at eg. `your-domain.com/id5-bootstrap.js` or `your-domain.com/assets/boot.js` (the `path` part does not matter). The ways this can be achieved are described in the first-party hosting section.

2. The bootstrap library should be loaded and initialized in the `<head>` section of the webpage, ideally before any other scripts, so the redirect (if it's needed) will happen as fast as possible and the browser won't start rendering content. It is required to pass your ID5 partner ID to the init method. For example, if your ID5 partner ID is `99`:

```
<head>
 <title>Your domain</title>
 <script type="application/javascript" src="https://your-domain.com/id5-bootstrap.js">
 <script type="application/javascript">
  window.id5Bootstrap.initTrueLinkId(99)
 </script>
</head>
```

3. The script will register a global object `window.id5Bootstrap` with method `initTrueLinkId` which should be called directly after including the script. Calling `initTrueLinkId` does few things:

    - The process checks whether ID5 has previously obtained consent on the page and whether the user has consented to ID5 (in applicable regions). If the user has consented and the TrueLink ID is either absent or requires a refresh, the redirect process will proceed. However, if there is no consent information or if the user hasn't consented to ID5, the redirect process will be terminated.

    - Checks if there is a first-party cookie with the TrueLink ID - if yes (and it isn't expired), it returns, if not - it redirects the user to ID5 TrueLink server (https://id5-sync.com/true-link) which will redirect back to the original page with encrypted TrueLink ID in the URL.

    - Checks if there is a TrueLink ID the URL - if yes, it sets it as a first-party cookie and returns.

4. In parallel with the TrueLink bootstrap library, you must be running an ID5 ID integration using the latest version of ID5 JS API, available through our CDN or using our module in Prebid.
   If you run the ID5 ID integration via our Prebid User ID Module, you must run **Prebid.js v8.33.0** or higher and you should provide the `externalModuleUrl` parameter to the our identity module.

> The id5-bootstrap.js operates using encrypted values of the TrueLink ID, which change with each call to the TrueLink endpoint even if the underlying value remains the same. These encrypted values are not intended for direct use. Instead, you should obtain the decrypted TrueLink ID using the ID5 API library methods like described in the example below.

Serving id5-bootstrap.js as First-Party

The TrueLink integration library id5-bootstrap.js must be served as a "first-party" script in order to be able to work. In this sections we document a few possibilities.

Proxy Method (Preferred)

We suggest proxying the TrueLink id5-bootstrap.js library version provided by ID5 through our CDN. This involves fetching the contents of the id5-bootstrap.js script from the CDN on demand or periodically and serving it as a regular asset on your domain.

Alternatively you can set up a proxy rule on your serving or load-balancing infrastructure.

The method for accomplishing this depends heavily on your specific technology stack. If you encounter any issues or have questions, please don't hesitate to reach out to us with a description of your tech stack so we can offer assistance.

This method is preferred since you will benefit from improvements ID5 makes to the id5-bootstrap.js library automatically.

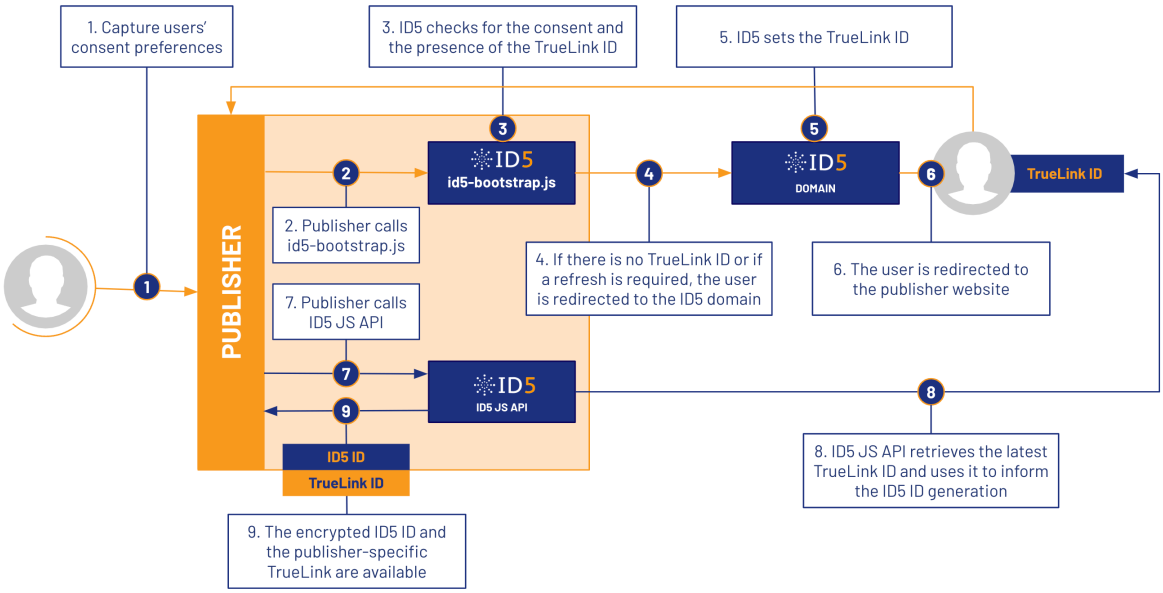
Static Method

An alternative method to achieve first-party hosting is to either copy the TrueLink id5-bootstrap.js library from our CDN or build it from source and then serve the content as you would for any other javascript asset on your webpage.

While this method gives you full control over the contents of the id5-bootstrap.js library, you will be required to update the contents of the script whenever ID5 makes changes to it (to introduce new functionality, bugfixes, etc.) as we don't support outdated integrations.

TrueLink Workflow

The below workflow describes how the TrueLink ID is generated and made available. Please note, users will only be redirected if there is no TrueLink ID set or it requires refreshing. Initially, ID5 only plans to refresh the TrueLink ID once per user per 7 days but may adjust this.



Demo Pages

To illustrate how ID5 creates the encrypted and publisher-specific TrueLink IDs for users across publisher media properties, we created the following demo pages - Domain A and Domain B. To see the demo,

please start with Domain A and follow the step-by-step commentary to learn more about the actions taking place on each page.

Guarded Publisher ID

03/14/2025 8:37 am EDT

Introduction to Guarded Publisher ID (GPID)

ID5 is offering partners a license to the Guarded Publisher ID, a publisher specific first-party cross-domain identifier. The GPID is a partner specific version of the ID5 ID which remains unique for a user across a owned and operated properties within a given browser environment. It's quality and stability is implicitly linked to the provision of signals such as hashed emails and TrueLink signals. It is especially useful as a first party user identifier for media owners or brands with multiple owned and operated properties with a limited amount of authenticated traffic.

Since the GPID is derived from the ID5 ID but it will not have cross-publisher user reconciliation properties. This means that the same user will have a different GPID on other publishers' domains or apps. The GPID will expire after 90 days in order to protect user's privacy.

The GPID can be used to facilitate use cases such as:

- Cross domain audience building and activation
- Campaign optimisation and frequency capping
- Measurement
- As a Publisher Provided ID (PPID) in an ad server such as Google Ad Manager, Xandr.

The GPID is at its most accurate and stable when publishers:

- Share additional signals like hashed email as described here
- Implement an ID5 TrueLink integration alongside their standard ID5 deployment

Publishers with a Prebid.js integration

Using GPID as PPID in Prebid v9.31 or higher.

Guarded publisher ID is provided as PPID and can be natively used in Prebid after version 9.31. To use it, the `gpid.id5-sync.com` needs to be provided as a ppid source in prebid userSync configuration.

> This implementation will only work if the `externalModuleUrl` is configured in our UserIds Prebid module.

```
pbjs.setConfig({
  userSync: {
    ppid: 'gpid.id5-sync.com',
    userIds: [],  //userIds modules should be configured here
  }
});
```

Publishers with an ID5 JS API Integration

> This is sample code only and may need some modification in your production environment.

```
var id5Callback = function (id5Status) {
// ... existing callback code


// pick up the gpid
  var ppid = id5Status.getGpId();
  if (ppid) {
// pass the gpid as PPID to GPT tag
    window.googletag.pubads().setPublisherProvidedId(ppid);
  }
};
var id5Status = ID5.init({ partnerId: 173 }).onUpdate(id5Callback);
```

Publisher Provided ID (PPID) Provisioning

04/03/2025 10:01 am EDT

Publisher Provided ID (PPID) Provisioning

Some monetization and ad serving platforms like Google and Xandr supporting the passing of a Publisher Provided Identifier (PPID). Google's PPID, for example, allows publishers to send Google Ad Manager an identifier for use in frequency capping, audience segmentation and audience targeting, sequential ad rotation, and other audience-based ad delivery controls across devices. Provision of a PPID can increase CPMs from Google's buying stack.

Some publishers have expressed interest in injecting a decrypted ID5 ID into the PPID field, however, there are some challenges with doing this:

- PPID is expected to be a publisher first party ID. The ID5 ID has cross publisher and cross-domain reconciliation properties.
- User's privacy choices are managed through encryption of the ID5 ID and only vendors with consent are able to decrypt the ID.
- Decryption should not take place client side as this could expose secret keys

ID5 Solutions

The Guarded Publisher ID (GPID) as a PPID

ID5 is offering partners with an ID5 ID deployment a license to the Guarded Publisher ID, a publisher specific first-party cross-domain identifier. The GPID is a partner specific version of the ID5 ID which remains unique for a user across a owned and operated properties within a given browser environment. It's quality and stability is implicitly linked to the provision of signals such as hashed emails and TrueLink signals. You can read more about the GPID and how to retrieve it here.

ID5 JS API Lite

04/01/2025 4:14 am EDT

ID5 JS API Lite

ID5 JS API Lite is a reduced-functionality version of the standard ID5 JS API. Unlike the full API, which facilitates a comprehensive set of identity-related functions, the Lite version is purposefully built to perform a single operation - retrieving an encrypted ID5 ID that has been previously generated by another ID5 integration present on the same page. Compatible ID5 integrations include the standard ID5 JS API, ID5 Prebid module, Amazon Publisher Services (APS), and Google ESP.

ID5 JS API Lite does not perform ID generation, user consent handling, or any additional identity-related processes. It is intended solely for scenarios where ID retrieval is required without the need for full-featured API capabilities.

> If no other qualifying ID5 integration has been executed on the page, ID5 JS API Lite will not be able to retrieve an ID5 ID.

Who is it for?

ID5 JS API Lite is designed for partners with specific technical or operational constraints, such as:

- Scenarios where a minimal integration footprint is critical, for example, when deploying from within a creative.
- Situations where partners need to access an existing encrypted ID5 ID but are unable or prefer not to manage user consent directly.

Example Use Cases

The ID5 JS API Lite is approved for use in a defined set of scenarios, including:

1. Campaign Measurement
Advertisers, ad servers, or measurement providers may deploy ID5 JS API Lite within a creative to collect existing encrypted ID5 IDs and associate them with campaign events such as impressions, clicks, and conversions. This enables accurate campaign performance analysis and ROI measurement in environments where cookies are unavailable or restricted. [Learn more]

2. Activation and Measurement Under Constraints
Certain vendors may require access to existing encrypted ID5 IDs without generating new ones due to legal or technical limitations. For instance, vendors operating within an i-frame may be unable to directly fulfill ID5's contractual obligations regarding user consent collection. In such cases, ID5 JS API Lite can be used to retrieve IDs already generated elsewhere on the page.

ID5 Javascript API Workflow

Here's a step-by-step guide to integrating ID5 effectively. After configuring your ID5 JS API to retrieve the ID5 ID, you can share it with partners on your page using a single JavaScript variable. Partners can then transmit the ID5 ID to their platforms via existing tags or pixels to communicate user identity.

The ID5 ID serves as a key to build and activate audiences, optimize campaigns, and measure performance, even in cookie-less environments. This helps publishers future-proof their user addressability and sustain advertising revenue streams.

1. Register with ID5

The ID5 ID is free to distribute but requires a simple registration with us. If you don't already have an account with ID5, please visit our website to sign up and request your ID5 Partner Number.

2. Integrate the ID5 JS API Lite



We recommend that you monitor our releases in order to stay up-to-date with any changes to the library.

The ID5 JS API is an open-source library, available on GitHub: https://github.com/id5io/id5-api.js. All documentation for building and installing the library will be maintained in GitHub, but please reach out to support@id5.io if you have any questions or need help.

Data Onboarding - Match Preview

10/01/2026 10:48 am EDT

Match Preview: Match Test Guide

This guide covers how to run a match test using ID5's Data Onboarding Service for an initial match test against the ID5 identity universe. The match test (**Match Preview**) is free and returns a coverage report showing how many of your hashed emails (HEMs) ID5 can resolve to ID5 IDs and optionally other identifiers.

Prerequisites

1. A file containing SHA-256 hashed emails, formatted as described in Input File Specification

2. One of the following for data transfer options:

Data transfer options

Option	Description
Your own S3 bucket (preferred)	Grant ID5 read access to a prefix where you upload the input file, and write access to a prefix where ID5 delivers the result
ID5-hosted bucket (`eu-central-1`)	See sub-options below

If ID5 hosts the bucket:

Your AWS status	What happens
You have an AWS account	ID5 creates a bucket and authorizes your AWS account on it directly. You read and write using your own IAM roles and users — no credentials are exchanged, and ID5 can revoke access at any time
You have no AWS account	ID5 creates the bucket and issues a dedicated access key and secret, restricted to your upload and delivery prefixes. The secret is shared once through a secure channel

---

1. Service Configuration

Before onboarding, confirm the following configuration preferences with your ID5 account manager:

Configuration	Description
Match methodology	**Current Matching** — Returns matches for the last observed identifier for the input signal **Individual Matching** — Returns matches for all identifiers associated with the individual linked to the input signal in the ID5 graph. **Household Matching** COMING SOON — Returns matches for all identifiers associated with the household linked to the input signal in the ID5 graph
Geography	Countries considered for the match test.
Lookback Days	The number of days to retrieve connections in the past (max 90 days).
Partners to match cookie IDs with	List of partner IDs to perform partner cookie ID matching with.

2. Input File Specification

HEM is the input signal type used throughout this guide; the field name `input_uid` is generic to support additional signal types in future.

Emails must be normalized and hashed using SHA-256 before sending to ID5:

- Remove leading and trailing spaces
- Convert all ASCII characters to lowercase

Your input file must contain one HEM per row, with a header row.

Format: Parquet (zstd compression supported)

Field	Required	Description

Field	Required	Description
`segment`	Optional	The segment associated with the input signal. Include a segment column if you want to to see match rate broken out by segment.
`input_uid`	Required	The signal to be matched.
`input_uid_type`	Required	The type of signal (e.g., `HEM`).

Upload Location & Structure

Location	S3 location agreed upon with ID5
Partitioning	Date-partitioned folders, format `/DATE=yyyy-mm-dd` (processing date)
Completion signal	An empty `_SUCCESS` file must be written to the S3 prefix once upload is complete, to signal the file set is ready for processing

3. Match Test Output

The match test output returns, for each identifier type, the count and percentage of HEMs with at least one match, along with the count of unique matched identifiers.

Delivery Details

Location	Configured prefix, defaulting to `/match-preview/DATE=yyyy-mm-dd`
Partitioning	By date (`DATE=yyyy-mm-dd`)
Format	CSV

Fields

Field	Description
`matched_uid_type`	The identifier type (or `(all)` for the aggregate row)
`num_input_uids_distinct`	Distinct count of input identifiers
`num_matched_input_uids_distinct`	Distinct count of input identifiers with at least one match per identifier type
`match_rate_pct`	`num_matched_input_uids_distinct` ÷ `num_input_uids_distinct`, as a percentage
`num_matched_uids_distinct`	Distinct count of matched identifiers of this type

Example output — no `segment` column in input

matched_uid_type	num_input_uids_distinct	num_matched_input_uids_distinct	match_rate_pct	num_matched_uids_d
(all)	10000	8000	80	30312
ID5_ID	10000	8000	80	24100

ID5 Match Preview - Snowflake Native Application

09/14/2026 12:19 pm EDT

Set Up the ID5 Match Preview Native App in Snowflake

This guide walks you through running a free match test against the ID5 identity universe using the **ID5 Match Preview** native app in Snowflake. The app is publicly listed on the Snowflake Marketplace, no prior access request needed. You provide your first-party identifiers and the app returns a coverage report showing how much of that input we can resolve within the ID5 identity graph. The report breaks down match rates and unique match counts against ID5 IDs, MAIDs, CTV IDs, and partner cookies, so you can see exactly where your data resolves before committing to activation.

Prerequisites

- An active Snowflake account (any edition), Before proceeding, a user with the ORGADMIN role (or higher) must accept the Snowflake Consumer Terms of Service — see Snowflake's instructions.
- A table in Snowflake containing your HEMs formatted as described below available in AWS / eu-central-1
- A warehouse (compute) available to run queries — medium size is recommended for most use cases

Step 1: Prepare Your HEM Table

Create a table in your Snowflake account with the following structure:

Column	Type	Description
`id`	VARCHAR	The (hashed) email (SHA-256 hash of the lowercased email address)
`id_type`	VARCHAR	`HASHED_EMAIL` or `EMAIL`

SQL Example:

```
CREATE TABLE my_database.my_schema.hem_input (
    id      VARCHAR NOT NULL,
    id_type VARCHAR NOT NULL
);

INSERT INTO my_database.my_schema.hem_input VALUES
    ('b642b4217b34b1e8d3bd915fc65c4452...', 'HASHED_EMAIL'),
    ('e99a18c428cb38d5f260853678922e03...', 'HASHED_EMAIL');
```

Important formatting notes:

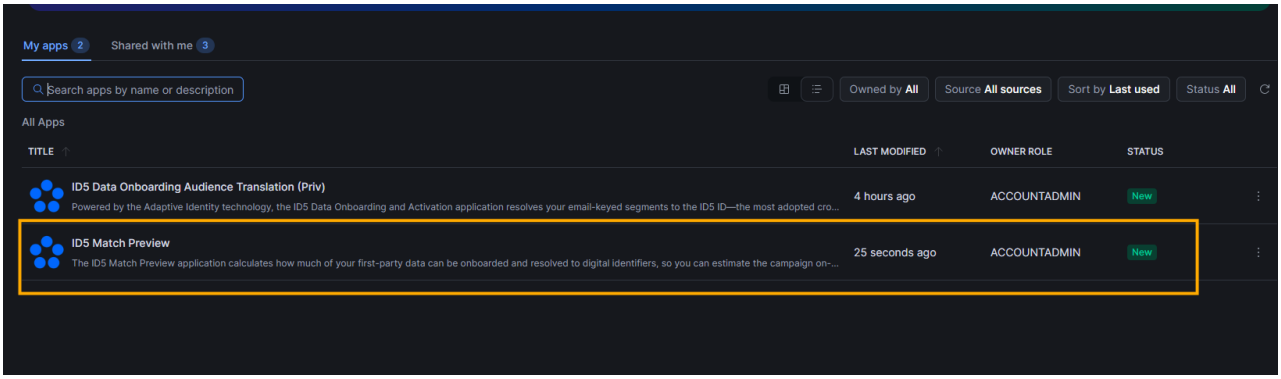
- Emails must be lowercased before hashing.
- The hashing algorithm must be SHA-256.
- If emails are not pre-hashed, we hash them before onboarding — please ensure they are normalized first, as described in Normalizing Emails Prior to Hashing.
- Both the `id` and `id_type` columns must be `NOT NULL`.

- Duplicate rows are handled automatically by the app — no need to deduplicate beforehand.

Step 2: Install the Match Preview App

> The ID5 listing terms are accepted upon installing the application from the Snowflake Marketplace.

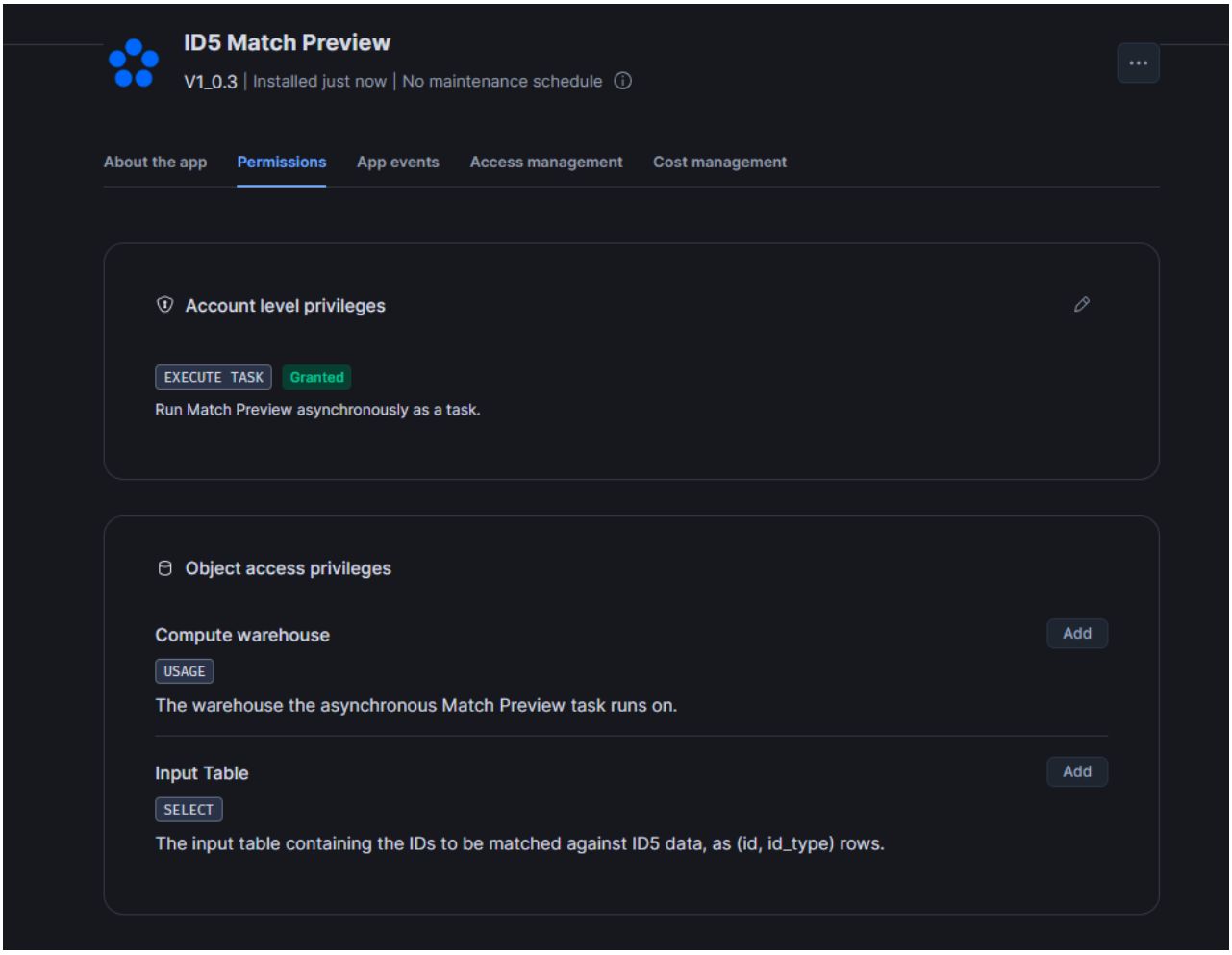
1. In Snowflake, navigate to **Marketplace** in the left-hand menu.
2. Search for **"ID5 Match Preview."**
3. Select **ID5 Match Preview** from the results.
4. Click **Get.**
5. Accept the permissions prompt — this allows the app to access only the resources you explicitly grant in the next step.
6. Optionally rename the app (e.g. `MATCH_PREVIEW`) and confirm installation.
7. Once installed, the app will appear under **My Apps** — installation takes less than 2 minutes.



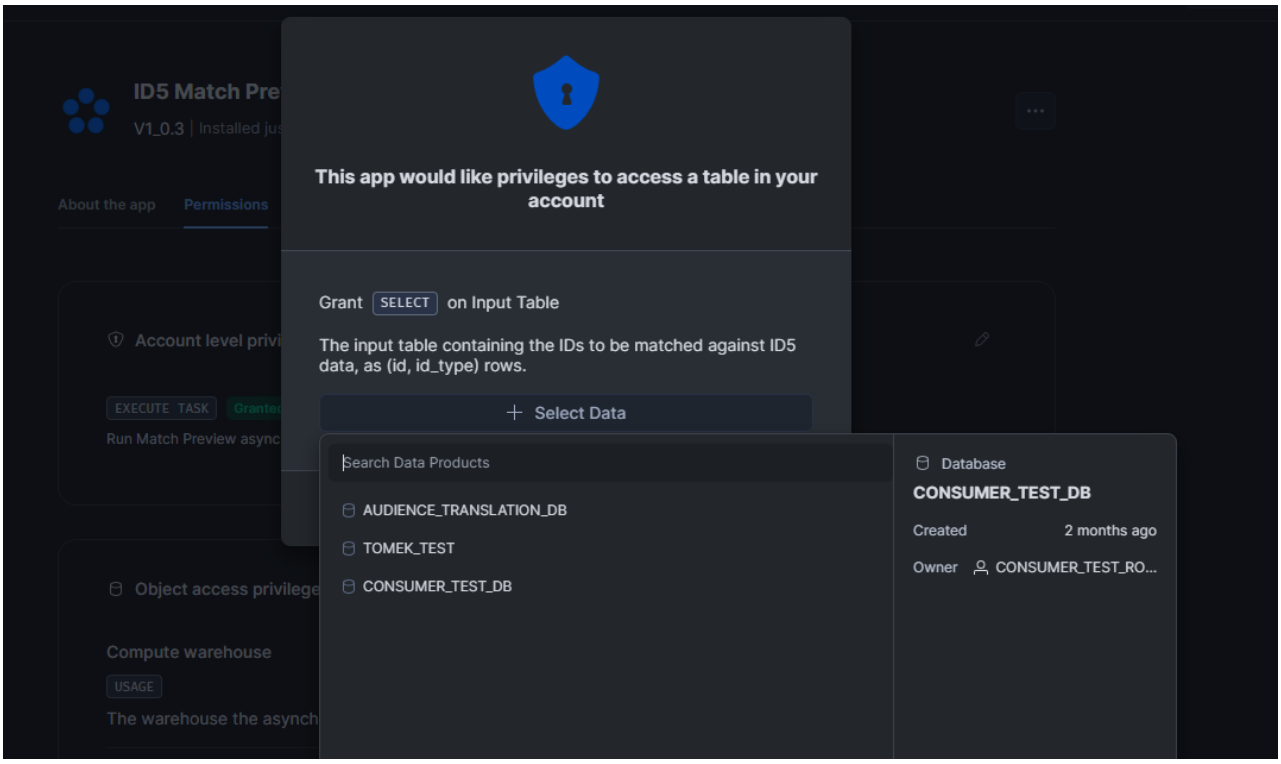
Step 3: Set Up Permissions (References)

> This step is required for privacy reasons — the app can only access resources you explicitly allow. It only requests read (SELECT) access to your input table; your data is never retained inside the app.

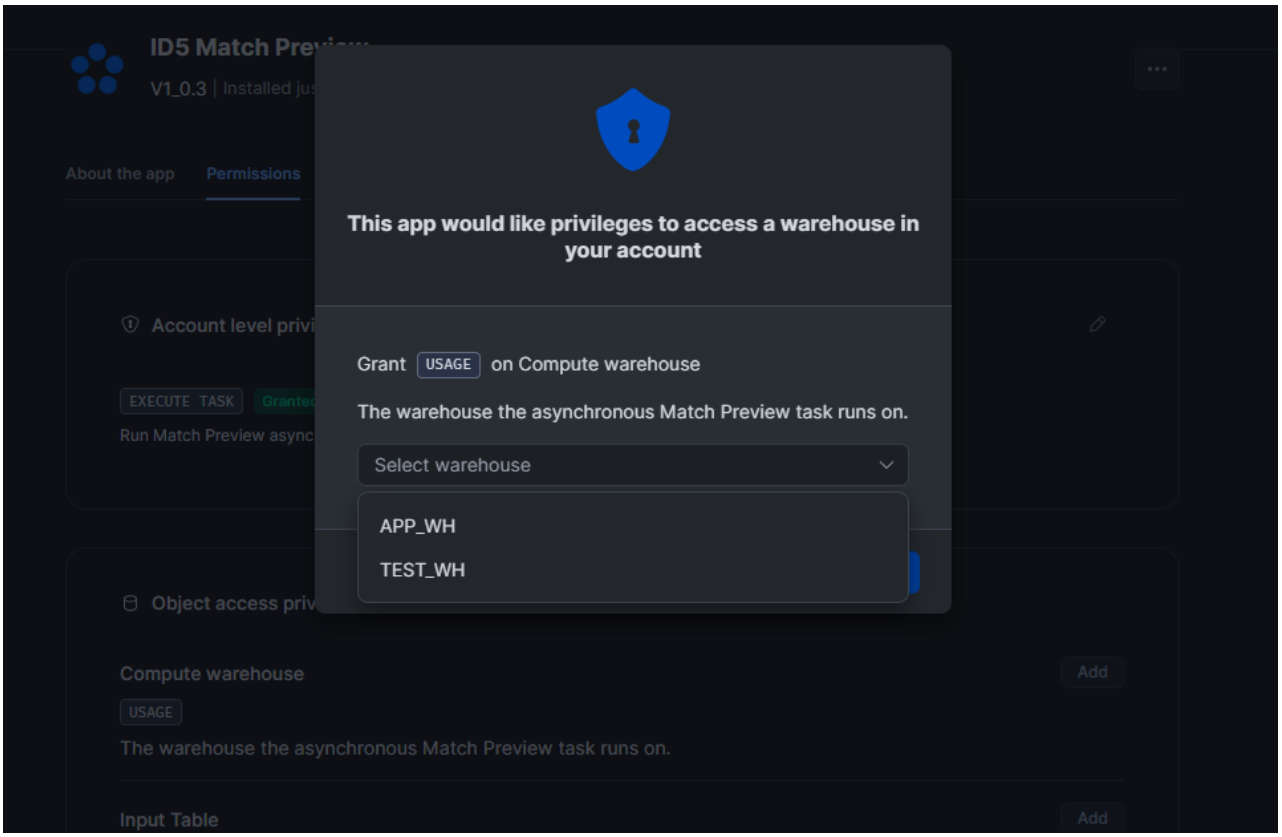
1. After installation, you need to grant the app access to your warehouse and input table. Go to**Apps > Select Match Preview App > Permissions:**



2. Grant the application access to your data by adding your table name to the **Input Table.** This table should contain the input signals you want to join on (e.g. your SHA-256 normalized hashed emails), to get a high-level understanding of match rates via the app.



3. Choose the **compute warehouse** you want to use. Medium sized is recommended for most use cases.



A warehouse is a specification of compute resources used to perform operations, governed by Snowflake's sizing and pricing rules. Warehouses auto-suspend after a period of inactivity. See Snowflake's warehouse documentation for more on creating and using them.

Alternatively run the following in a Snowflake worksheet:

```
-- Grant the app access to a warehouse
CALL MATCH_PREVIEW.core.register_reference(
    'COMPUTE_WAREHOUSE', 'ADD',
    SYSTEM$REFERENCE('WAREHOUSE', 'MY_WAREHOUSE', 'PERSISTENT', 'USAGE'));
-- replace MY_WAREHOUSE with your warehouse name

-- Grant the app read access to your HEM table
CALL MATCH_PREVIEW.core.register_reference(
    'MATCH_PREVIEW_INPUT_REF', 'ADD',
    SYSTEM$REFERENCE('TABLE', 'MY_DATABASE.MY_SCHEMA.HEM_INPUT', 'PERSISTENT', 'SELECT'));
-- replace MY_DATABASE.MY_SCHEMA.HEM_INPUT with your table path
```

Step 4 — Run a match test

Run the match procedure with your chosen parameters:

SQL

```
CALL MATCH_PREVIEW.core.match_preview(
    'ID',          -- name of your id column
    'ID_TYPE',     -- name of your id_type column
    ARRAY_CONSTRUCT('CURRENT', 'INDIVIDUAL')  -- match methods (see options below)
);
```

Match method options (at least one required):

Match Method	Description
CURRENT	Direct one-to-one HEM → ID5 ID mapping (last observed HEM to ID5 ID connection)
INDIVIDUAL	Expands to all ID5 IDs linked to the same individual via the ID5 graph

The results combine all selected methods. The procedure runs asynchronously — it returns immediately with an execution ID and processes in the background. A few hundred thousand HEMs on a medium warehouse typically completes in a few minutes.

Step 7 — Check Status

SQL

```
CALL MATCH_PREVIEW.core.match_preview_status();
```

This returns the status of the most recent run: **SCHEDULED, COMPLETED**, or **FAILED**, along with timestamps and the execution ID.

Step 8 — View Results

Once the status shows **COMPLETED**:

SQL

```
SELECT * FROM MATCH_PREVIEW.core.match_preview_output;
```

The output returns the amount and percentage of unique HEMs that have been matched to at least one ID5 ID as well as the total matched ID5 Ids.

Example output:

```
{
"matched_id5Ids": {
"total_matched_id5_ids": 2000,
"matched_input_hems": 1000,
"matched_input_hems_perc": 50
}
}
```

Snowflake compute costs (Client-Side)

Running the match test incurs a small Snowflake compute cost on your side (not charged by ID5). Based on our testing, we recommend starting with a medium warehouse. If you encounter resource errors, increase the warehouse size.

Data Sharing and Telemetry

To help us support you, the app shares aggregated telemetry with ID5. By accepting the app's permissions during installation, you agree to this sharing. The telemetry includes:

- Number of input HEMs and matched ID5 IDs
- Which match methods were used
- Execution duration and status
- Your Snowflake account and organisation identifier (to attribute the run to your account)

ID5 cannot see your individual HEM values. Your raw input data stays within your Snowflake account at all times and is never retained by the app. Telemetry data is kept for 14 days on the ID5 side.

Troubleshooting

If the match preview does not return a result, you can check the following options to troubleshoot.

Issue	What to check
App not visible in Marketplace	Search for "ID5 Match Preview" — if it still doesn't appear, check with your Snowflake admin that you have Marketplace access enabled
Job fails immediately	Check that the warehouse and input table references are correctly registered (Step 3) Check if your input data is stored and in **AWS / eu-central-1**. If not, ask your Snowflake representative to replicate the data to the target destination.

Low or zero matches	Verify emails are lowercased and normalised according to ID5 requirements before hashing, and that id_type is set to 'HEM'
Job takes a long time	Consider increasing the warehouse size

If you're stuck, share the execution ID with your ID5 account manager and we can check the error log on our side.

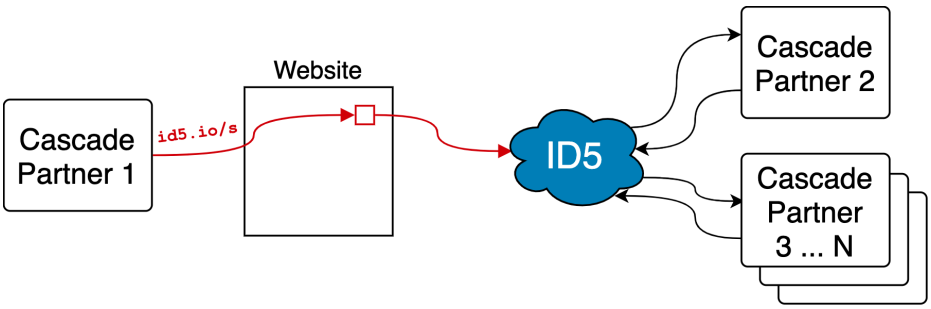
Initiate Cookie Sync to ID5

04/24/2026 5:21 am EDT

Process Overview

While third-party cookies are still in use, ID5 can synchronize with ad tech vendors to enhance user recognition, improving audience addressability and monetization.

ID5 can provide a pixel that can be added to publisher or advertiser pages or dropped alongside other code, like creatives or additional pixels. It can be deployed by a media owner or any ad tech vendor with access to a publisher's page. The pixel sends a request to ID5 with your UID (optional) to initiate a cookie synchronization with approved platforms. ID5 takes user consent into account when determining whether to initiate a cascade and in choosing which platforms to sync cookies with.



1. When you have access to a user's browser (via client-side tags, delivering a creative, etc.), drop our pixel on the page
2. ID5's servers receive the request, store your UID (optional), and determine the best platform to sync with (respecting user privacy choices and your preferences, and optimized by our algorithms)
3. ID5 redirects the request to this platform who in turn redirects back to ID5 with their UID
4. ID5 repeats steps (2) and (3) until we have no more platforms to sync with or the number of cascades you have allowed is reached

ID5 Cookie Sync Pixel URL with a User ID

https://id5-sync.com/s/**[id5AccountNum]**/**[numCascadesAllowed]**.gif?puid=**[encodedPlatformUid]**&gdpr=**[gdprFlag]**&gdpr_consent=**[tcfConsentString]**&gpp=**[gppConsentString]**&gpp_sid=**[gppApplicableSectio**

URL Path Parameters

Parameter	Type	Description
id5AccountNum	Integer	Your ID account number, provided by ID5
numCascadesAllowed	Integer	Number of additional platform calls allowed on the back of the initial synchronisation call to ID5 (step (4) above). We recommend setting this value to 9.

Querystring Parameters

Parameter	Type	Description
puid	String	Your unique ID for the user, encoded as a url parameter
gdpr	Integer (optional)	Whether or not GDPR applies to this request. Accepted values are 0 or 1.
gdpr_consent	String (optional)	A valid IAB TCF consent string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no consent string and process accordingly.
gpp	String (optional)	A valid IAB Global Privacy Platform consent string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no consent string and process accordingly.
gpp_sid	String (optional)	The GPP section ID(s) (integers) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation
us_privacy	String (optional)	A valid IAB US Privacy string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no US Privacy string and process accordingly.

Standard pixel

https://id5-sync.com/s/113/9.gif?puid=AABBC12345&gdpr=1&gdpr_consent=BOEFEAyOEFEAyAHABDENAI4AAAB9vABAASA

Initiate by Cookie Sync Pixel URL without a User ID

If you have access to the page you can simply add a pixel tag with URL:

https://id5-sync.com/i/**[id5AccountNum]**/**[numCascadesAllowed]**.gif?
gdpr=**[gdprFlag]**&gdpr_consent=**[consent_string]**&us_privacy=**[us_privcacy_string]**

or add an iframe with URL:

https://id5-sync.com/iwp/**[id5AccountNum]**/**[numCascadesAllowed]**.html?
gdpr=**[gdprFlag]**&gdpr_consent=**[consent_string]**&us_privacy=**[us_privcacy_string]**

This endpoint will drop an initial pixel and allows the inclusion of additional pixels, configurable in the ID5 console. For more details or assistance with configuring additional pixels, please contact your ID5 Account Manager or reach out to support at support@id5.io.

URL Path Parameters

Parameter	Type	Description
id5AccountNum	Integer	Your ID account number, provided by ID5
numCascadesAllowed	Integer	Number of additional platform calls allowed on the back of the initial synchronisation call to ID5 (step (4) above). We recommend setting this value to 9.

Querystring Parameters

Parameter	Type	Description
gdpr	Integer (optional)	Whether or not GDPR applies to this request. Accepted values are 0 or 1.
gdpr_consent	String (optional)	A valid IAB TCF consent string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no consent string and process accordingly.
gpp	String (optional)	A valid IAB Global Privacy Platform consent string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no consent string and process accordingly.
gpp_sid	String (optional)	The GPP section ID(s) (integers) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation
us_privacy	String (optional)	A valid IAB US Privacy string. If the string is missing, misconstructed, or otherwise invalid, we will treat the request as if it has no US Privacy string and process accordingly.

Pixel URL

```
https://id5-sync.com/i/113/9.gif
```

Standard Pixel

```
<img src="https://id5-sync.com/i/113/9.gif" style="display:none;" height="0" width="0" id="id5-pixel"/>
```

See a working example on the sample page.

Adding a pixel with JavaScript

```
function firePixel(url) {
let img = new Image();
img.src = url;
}

(function () {
let syncUrl = "https://id5-sync.com/i/113/9.gif"
if (document.readyState !== 'loading') {
  firePixel(syncUrl);
} else {
    document.addEventListener('DOMContentLoaded', function () {
    firePixel(syncUrl);
  });
}
})();
```

See a more advanced example with GDPR consent collecting on the sample page

Adding a pixel as an iFrame

```
<iframe src="https://id5-sync.com/iwp/113/9.html"/>
```

Initiate by ID5-API.JS (Publisher)

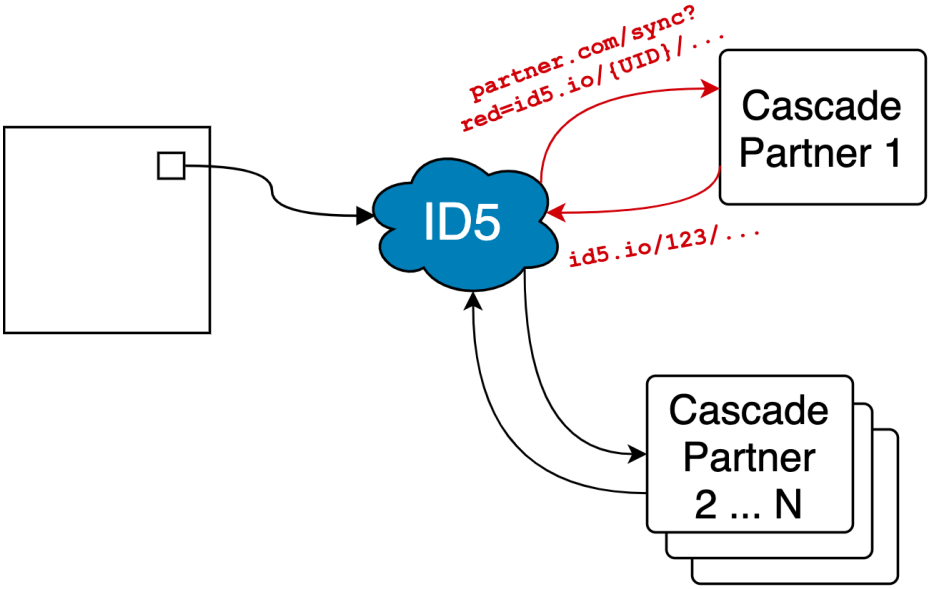
If you have integrated our id5-api.js on your page, user syncing will automatically initiate user syncing once the ID5 ID has been provisioned to the page. No additional user sync integration is required. You can configure the maximum number of syncs using the `maxCascades` property. If you have your own proprietary user ID, you can pass it to the ID5 API via the `partnerUserId` configuration property. More details can be found here.

Receive Cookie Sync from ID5

04/24/2026 4:53 am EDT

Process Overview

As part of our synchronization process, we will need to make requests to your system that will return back to ID5 with your UID. There are several methods supported for this process that are explained in more detail.



1. ID5 will initiate ID synchronizations to you; we refer to these synchronizations as "cascades". In order to receive cascades, you will need to provide ID5 with your cookie sync URL (endpoint). ID5 relies on HTTP redirects (i.e. 302) to handle these cascades.

2. ID5 will call the URL you provide and expect to be called back by redirecting the user to a URL provided by ID5.

> You must always redirect users to the callback URL provided by ID5, even if the user is unknown to your system. If the user is unknown, you should create an identity for this new user and return their new UID to ID5 in the callback, or pass a `0` as the UID.

GDPR Support

ID5 will check that you have consent before performing cookie synchronizations, in addition to checking that ID5 has consent before processing. Within the pixel URL that you provide us during integration setup, you may specify parameters for ID5 to pass you a GDPR flag and an IAB TCF consent string. Typically, the two querystring fields are `gdpr` (with a value of `0` or `1`) and `gdpr_consent`, respectively, but if you have a different setup, please let us know.

If GDPR applies to a particular request, we have a consent string, and you do not have consent, ID5 will not call your pixel. However, if we do call you and you then determine that you do not have proper consent for you to process data, you should still redirect back to ID5 like normal, and pass an empty string as the value for your UID.

Callback URL Options

We support three methods for callback URLs that you must supply us (step 2 in the diagram above). In all cases, we only support making calls to HTTPS endpoints.

1. **Dynamic Callback URL**
    - If you support receiving a dynamic callback URL in the querystring, ID5 will provide a URL containing a macro to be replaced by your UID. This process allows ID5 to update its own synchronization URL without maintenance from you. **This is our recommended implementation.**

2. **Static Callback URL**
    - If you are not able to use a dynamic URL callback in pixel redirection, ID5 can store the state of cascade in cookies or via parameters on the request.
    - *NOTE: this means that you may have to update ID5's callback URL in case of major changes/updates in the process.*

Dynamic Callback URL

In these scenarios, ID5 will dynamically generate a callback URL to our system, providing a mechanism for you to include your UID back to us to complete the sync. We will pass this callback URL in the querystring of the pixel call to your system.

There are three types of callback URLs we can provide:

- Encoded URL with a macro *(recommended)*
- Unencoded URL with a macro
- Appendable encoded URL

Dynamic Callback URL Types

Encoded URL with a Macro

In this scenario, ID5 will call your pixel, passing an encoded URL with a macro into a parameter you define. Your endpoint is expected to decode the URL, replace the macro within the ID5 URL with your UID, and then redirect the browser with a 302 HTTP response to the URL.

URL Syntax

> https://s.platform.com/cookie-sync?callback=**[ENCODEDURL]**
> **{{UID}}**&foo=bar&gdpr_consent=**[TCF_CONSENT_STRING]**&gdpr=**[GDPR_FLAG]**&gpp=**[GPP_CONSENT_STRING]**&gpp_sid=**[GPP_APPLICABLE_SEC**

URL Parameters

Parameter	Type	Description
ENCODEDURL	String	A dynamically generated encoded URL that ID5 will populate in real time during the cascade process
UID	String	UID macro to be replaced with the *encoded* user identifier from your system. This will be placed inside the {ENCODEDURL} value by ID5. The macro text (i.e. `{UID}`) can be any string of text that you will replace with your own ID; this will need to be communicated to ID5 during the pixel set up phase.
TCF_CONSENT_STRING	String	(required in EEA) A valid IAB TCF consent string.
GDPR_FLAG	Integer	(optional) Whether or not GDPR applies to this request. We will send either 0 or 1.
GPP_CONSENT_STRING	String	(optional) A valid IAB Global Privacy Platform consent string.
GPP_APPLICABLE_SECTIONS	String	(optional) The GPP section ID(s) (integers) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation

Unencoded URL with a macro

In this scenario, ID5 will call your pixel, appending an unencoded url with a macro onto your endpoint. Your endpoint is expected to replace the macro within the ID5 URL with your UID and then redirect the browser with a 302 HTTP response to the URL.

URL Syntax

> https://s.platform.com/cookie-sync?callback=**[UNENCODEDURL]**
> **{{UID}}**&foo=bar&gdpr_consent=**[TCF_CONSENT_STRING]**&gdpr=**[GDPR_FLAG]**&gpp=**[GPP_CONSENT_STRING]**&gpp_sid=**[GPP_APPLICABLE_SEC**

URL Parameters

Parameter	Type	Description
UNENCODEDURL	String	A dynamically generated unencoded URL that ID5 will populate in real time during the cascade process
UID	String	UID macro to be replaced with the *encoded* user identifier from your system. This will be placed inside the {ENCODEDURL} value by ID5. The macro text (i.e. `{UID}`) can be any string of text that you will replace with your own ID; this will need to be communicated to ID5 during the pixel set up phase.
TCF_CONSENT_STRING	String	(required in EEA) A valid IAB TCF consent string.
GDPR_FLAG	Integer	(optional) Whether or not GDPR applies to this request. We will send either 0 or 1.
GPP_CONSENT_STRING	String	(optional) A valid IAB Global Privacy Platform consent string.
GPP_APPLICABLE_SECTIONS	String	(optional) The GPP section ID(s) (integers) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation

Appendable encoded URL

In this scenario, ID5 will call your pixel, appending an encoded url without a macro onto your endpoint. Your endpoint is expected to decode the callback URL, append the ID5 URL with your UID, and then redirect the browser with a 302 HTTP response to the URL.

URL Syntax

> https://s.platform.com/cookie-sync?
> foo=bar&gdpr_consent=**[TCF_CONSENT_STRING]**&gdpr=**[GDPR_FLAG]**&gpp=**[GPP_CONSENT_STRING]**&gpp_sid=**[GPP_APPLICABLE_SECTIONS]**&

URL Parameters

Parameter	Type	Description
TCF_CONSENT_STRING	String	(required in EEA) A valid IAB TCF consent string.

Parameter	Type	Description
GDPR_FLAG	Integer	(optional) Whether or not GDPR applies to this request. We will send either 0 or 1.
ENCODEDURL	String	A dynamically generated unencoded URL that ID5 will populate in real time during the cascade process
GPP_CONSENT_STRING	String	(optional) A valid IAB Global Privacy Platform consent string.
GPP_APPLICABLE_SECTIONS	String	(optional) The GPP section ID(s) (integers) in force for the current transaction. In most cases, this field should have a single section ID. In rare occasions where such a single section ID can not be determined, the field may contain up to 2 values, separated by a comma. More information in GPP documentation

Example Implementation

1. ID5 calls your pixel:

> https://s.platform.com/cookie-sync?callback=https%3A%2F%2Fid5-sync.com%2Ftcb%2Ft%3Fpuid%3D{{UID}}&foo=bar&gdpr_consent=BOEFEAyOEFEAyAHABDENAI4AAAB9vABAASA&gdpr=1&gpp=DBABMA~BOEFE

2. You decode the URL and replace `{UID}` with your UID for the user (in this example, ABCDE-12345), then redirect the user to:

> https://id5-sync.com/tcb/t?puid=ABCDE-12345

3. ID5 continues cascading to other platforms

Static Callback URL

In these scenarios, you will hardcode a portion or the entire URL to call back to ID5 after receiving a cookie sync request from us.

There are two types of static callback URLs we support:
- Static Callback URL with Dynamic Parameters *(recommended)*
- Static Callback URL

Static Callback URL Types

- Static Callback URL with Dynamic Parameters
- Static Callback URL

In this scenario, your pixel endpoint will be expected to redirect the browser with a 302 HTTP response to a hardcoded URL that we provide below, while appending any querystring parameters we provided on the call to your servers.

This method is preferred over the normal static callback integration, as it doesn't rely on cookies to store the cascade state.

Callback URL Syntax

> https://id5-sync.com/k/**[id5AccountNum]**.gif?puid=**[encodedPlatformUid]**&**[allQsParamsFromInitialCall]**

Callback URL Parameters

Parameter	Type	Description
id5AccountNum	Integer	Your ID account number, provided by ID5
encodedPlatformUid	String	Your UID for the user, encoded as a url parameter
allQsParamsFromInitialCall	String	One or more querystring parameters that were originally passed to your platform on the request to you. These should not be modified

Example Implementation

1. ID5 calls your pixel:

> https://s.platform.com/cookie-sync?p=123&c=2&ca=8

2. Your pixel would redirect to the following URL

> https://id5-sync.com/k/113.gif?puid=ABCDE-12345&p=123&c=2&ca=8

3. ID5 continues cascading to other platforms

Self-Testing the Cascade

To test the cascade process by yourself, you can emulate the cascade by using the following values as the ID5 redirect URL, placing your UID macro at the end in place of `uidValueHere`:

```
[UNENCODEDURL] = https://id5-sync.com/tcb/t?puid=uidValueHere
[ENCODEDURL] = https%3A%2F%2Fid5-sync.com%2Ftcb%2Ft%3Fpuid%3DuidValueHere
```

The browser should reply with your own user ID. This will allow automated test on both side.

Example Testing

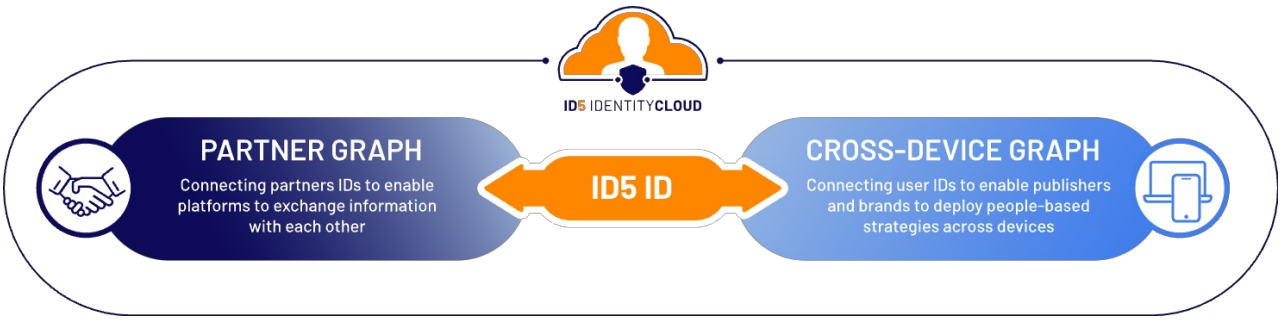
Load your pixel endpoint in a browser, substituting the `[ENCODEDURL]` or `[UNENCODEDURL]` macros from above, making sure to include your UID at the end of our test callback URL.

Overview of ID5 Identity Graphs

01/16/2025 1:48 pm EST

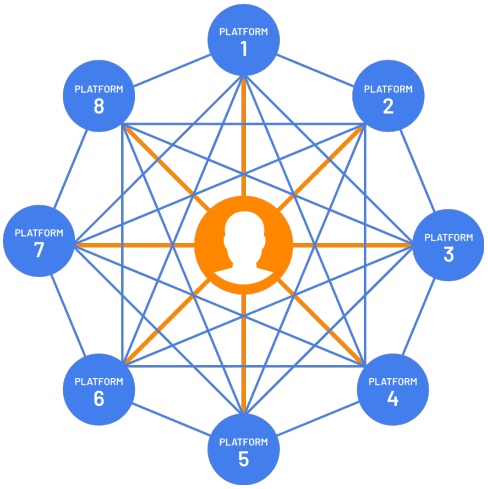
Overview

IdentityCloud includes a Partner Graph and a Cross-Device Graph, both enriched by the ID5 ID.



Partner Graph Overview

ID5's Partner Graph is a centralized graph containing information on how ID5 IDs map to other platforms' cookie-based user IDs and how these different platform user IDs map to each other. By leveraging the ID5 Partner Graph, technology platforms can operate efficiently in cookie-based browsers, while preparing for the post-cookie era, and improve match rates with partners, maximizing advertising results.



Accessing the Partner Graph

Once cookie syncing with ID5 has been implemented, we are able to start delivering user mappings to other platforms. We can deliver these mappings to you in several ways:

> **Streaming File Transfer** | Preconfigured Match Partner UIDs are automatically pushed to you on an incremental basis at regular time intervals

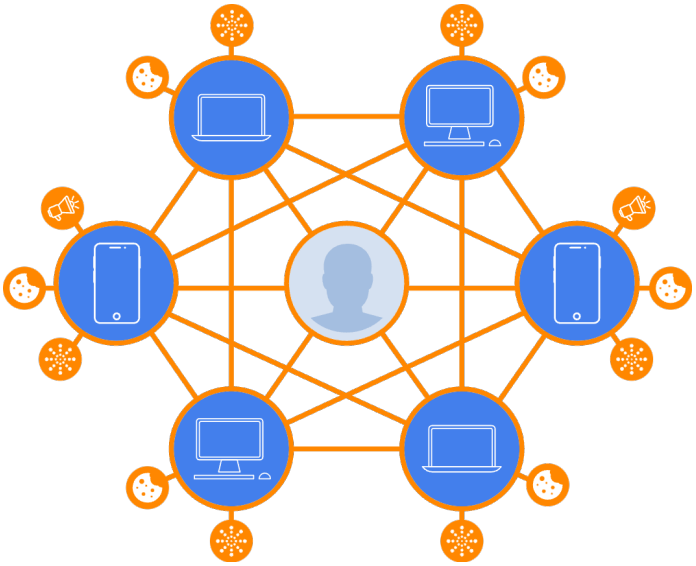
> **File-based UID Matching** | ID5 will match a set of your UIDs provided via an uploaded file to one or more preconfigured Match Partners and return them via file upload

> **Web Service API** | You will make requests to an ID5 endpoint to request user matches to/from a

> preconfigured set of Match Partners for a set of users with near-realtime responses

Cross-Device Graph Overview

ID5's Cross-Device Graph is a privacy-first solution enabling brands and publishers to deploy people-based marketing strategies across devices. It helps brands, ad tech platforms and large publishers to connect different signals (Cookies, MAIDs, CTV ids, etc) to the same individual and/or household. The Cross-Device Graph covers all major European countries. Rather than relying on bidstream data to build its Cross-Device graph, ID5 leverages its direct relationship with publishers to capture consent signals and provide a high quality, privacy-compliant graph.



Accessing the Cross-Device Graph

Typically, cookie syncing is a pre-requisite to the Cross-Device graph (but not always, so speak to your ID5 representative about this).

> **Streaming File Transfer** | ID5 sends the entire graph with pre-configured settings on a weekly basis

Cloud Storage

For both the Partner and Cross-Device graph, we generally rely on an AWS (Amazon Web Service) S3 buckets to exchange data; the S3 Bucket can be provided by ID5 or by you. We also have the option to use a different cloud platform. Please read more on our Client Data-Sharing Specification documentation page.

To comply with the General Data Protection Regulation (GDPR), the AWS S3 bucket must be physically located in a European datacentre (Paris, Dublin or Frankfurt) if it has to handle data related to European users. ID5's platform follows a privacy-by-design principle ensuring that data is protected and that no other piece of information is transferred alongside UIDs.

We recommend that you own the AWS S3 bucket in which pseudonymized UIDs are exchanged. To share the bucket with ID5, you will need to provide ID5 permission via our AWS canonical ID (bucket and permissions can be manually set

from https://s3.console.aws.amazon.com/ or using the `aws-cli`).

Client Data Sharing Specification

08/14/2026 11:43 am EDT

Exchanging data with ID5

ID5 uses RClone to send and receive data, which determines the supported transfer protocols.

Our main delivery mechanism is to exchange data directly via an AWS S3 bucket or an S3-compatible object storage (e.g., GCS) controlled by the client.

There are several different ways to share data:

1. Client-Hosted Bucket (push method)

The preferred method is for clients to host their own AWS S3 or S3-compatible storage. To help with this, ID5 will share our AWS account ID.

The required permissions and configuration examples for select cloud providers are detailed below.

2. ID5-Hosted Buckets (pull method)

For clients unable to host their own S3-compatible storage, or for specific use cases where pull is the better fit, we offer an alternative where ID5 hosts the bucket. In this scenario, we support two methods for client authentication and access, both of which require the client to have an AWS account:

Direct Cross-Account Access via Bucket Policy

We attach a bucket policy to the hosted bucket that explicitly grants read (and/or write) permissions to principals in the client's AWS account. Once this is in place, clients are free to manage access on their side — for example, delegating permissions to specific IAM users, roles, or applications through their own IAM policies. This approach is straightforward and works well when your team already has established IAM governance.

IAM Role-Based Access (AssumeRole)

We create a dedicated IAM role in our AWS account with the appropriate permissions on the bucket and configure its trust policy to allow the client's account to assume it. Client applications then call `sts:AssumeRole` to obtain short-lived, scoped credentials. This is the recommended approach for programmatic or automated workloads, as it avoids the need for long-lived credentials and provides a clear audit trail through CloudTrail.

In either case, we require the client's AWS account ID(s) or canonical user ID(s). The choice between these options will depend on your security preferences and existing AWS setup.



For security reasons, ID5 does not support creating buckets or providing client access via Access and Secret keys.

Data Retention

For all storage hosted by ID5, our data retention policy expires data after 90 days.

Supported Data Formats

We prefer to deliver and receive data in Parquet format with `zstd` compression. We also support CSV and JSON, compressed with `gzip` or `zstd`, as well as `snappy` compression for Parquet.

Permissions Details

Amazon Web Services (AWS)

S3 Bucket Policy Requirements

- Account ARN is: **arn:aws:iam::243105029713:root**
- Dedicated IAM role: **arn:aws:iam::243105029713:role/id5-cross-account-data-access**
- We can handle either an entire bucket or a specific prefix.

Permissions are split into read-only and read-write depending on your use case.

Read-only:
Required for retrieving data and verification purposes:

Permission	Scope	Why
`s3:ListBucket`	Bucket	List objects in a bucket or prefix
`s3:GetObject`	Object	Download objects, get metadata

Read-write:
Required when ID5 also uploads data to your bucket:

Permission	Scope	Why
`s3:ListBucket`	Bucket	List objects in a bucket or prefix
`s3:GetBucketLocation`	Bucket	Required by most S3 clients for bucket checks
`s3:PutObject`	Object	Uploads and overwrites
`s3:GetObject`	Object	Metadata/integrity checks
`s3:DeleteObject`	Object	Needed in case of re-delivery of the same dataset
`s3:PutObjectAcl`	Object	Uploads and overwrites with the correct ACL
`s3:CreateMultipartUpload`	Object (multipart)	Initiates multipart uploads for large files *(optional)*
`s3:UploadPart`	Object (multipart)	Uploads each chunk in a multipart upload *(optional)*
`s3:CompleteMultipartUpload`	Object (multipart)	Finalizes the assembled object *(optional)*
`s3:AbortMultipartUpload`	Object (multipart)	Cleans up failed upload fragments *(optional)*

Policy Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::243105029713:root"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/some/prefix/*"
      ]
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::243105029713:root"
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "some/prefix/*"
          ]
        }
      }
    }
  ]
}
```

Other S3-compatible services

For S3-compatible services, we require that the client provide us with:

- The endpoint URL
- An Access Key ID
- A Secret Access Key

The bucket should have a policy equivalent to the AWS S3 permissions described above.

Google Cloud Platform (GCP)

We can interact with GCP buckets in two ways:

S3 Compatibility Layer (HMAC keys)

We require that you generate HMAC keys with the appropriate access to the storage. This leverages the interoperability layer provided by GCP.

Native GCP via Service Account

We require a JSON credentials file for a service account with the permissions listed below.

> **ATTENTION**
> `ObjectAdmin` is **not** required. The built-in Storage Object User or Storage Object Viewer roles cover the necessary permissions. You can also use a Managed Folder for fine-grained access control without granting permissions at the bucket level.

Read-only:

Permission	Scope	Why
`storage.objects.get`	Object	Download objects, get metadata
`storage.objects.list`	Object	List objects in a bucket or prefix
`storage.buckets.get`	Bucket	Bucket existence checks required by most S3 clients

Read-write:

Permission	Scope	Why
`storage.objects.get`	Object	Download / HEAD object
`storage.objects.list`	Object	List objects
`storage.objects.create`	Object	Upload objects
`storage.objects.delete`	Object	Delete objects; also required when overwriting an existing object
`storage.buckets.get`	Bucket	Bucket existence checks
`storage.multipartUploads.create`	Object (multipart)	Initiate multipart uploads
`storage.multipartUploads.abort`	Object (multipart)	Abort incomplete multipart uploads
`storage.multipartUploads.list`	Object (multipart)	List in-progress multipart uploads
`storage.multipartUploads.listParts`	Object (multipart)	List parts of a multipart upload

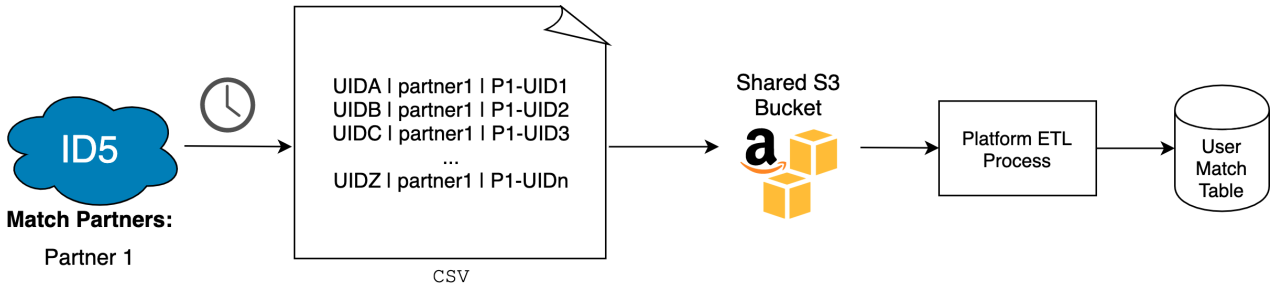
Azure

Azure is not natively supported for data transfers. However, since ID5 uses RClone, Azure integration can be arranged for clients who cannot use S3 or GCS alternatives — reach out to your ID5 contact to discuss.

Partner Graph File Transfer

04/23/2026 11:44 am EDT

Overview



With the Partner Graph File Transfer, your AWS S3 bucket will be filled at regular time intervals with either (1) streaming *new and updated* Match Partner UIDs every 30 minutes coming from the cascading process or (2) daily file delivery with all Match Partner UIDs.

> The time interval that ID5 pushes files could change over time. Your system should be built to look for new files and process them as they become available and not based on an expected interval from ID5. We recommend you look for new files every 5 minutes for streaming files.

Streaming Matched Partner Data

The data will be delivered to a directory called / `incremental` at the root of your bucket with subdirectories broken out by Match Partner (directory names based on the partner's Global Vendor List ID), then folders per UTC day `/YYYYMMDD/`, each containing files named with a timestamp throughout the day. For instance:

```
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221106/000000.csv
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221106/003000.csv
...
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221106/233000.csv
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221107/000000.csv
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221107/003000.csv
...
S3://[bucket]/incremental/[match partner 1 GVL ID]/20221107/180000.csv
S3://[bucket]/incremental/[match partner 2 GVL ID]/20221106/000000.csv
...
S3://[bucket]/incremental/[match partner 2 GVL ID]/20221106/233000.csv
```

Each .csv file contains the **incremental mappings** between your UIDs and the requested Match Partner's UIDs in a pipe-separated format. In other words, only new or changed (since the last file) UID pairs will be included in each file.

> Available Match Partners are defined by contract between you and ID5. To change the list of Match Partners, please reach out to your ID5 representative.

File Format

Column	Type	Description
UID	String	Your User ID for this user
MatchPartnerGVLID	Int	The Global Vendor List ID of the Match Partner
MatchPartnerUID	String	The Match Partner's User ID for this user

Example Output File

```
$ cat /incremental/matchpartner1GVLID/20220305/000300.csv
550e8400-e29b-41d4-a716-446655440000|matchpartner1GVLID|100000187421490458
d2a8378f-fe56-4ec2-96d1-3c05df02bb48|matchpartner1GVLID|1000002629570693845
```

Full Matching Table Refreshes

In addition to the incremental updates we push throughout the day, ID5 can stream the full match table on a regular basis. This ensures a couple of things:

- If any data is lost during the streaming process, the full extract will recover the data, rather than waiting for a change from that user to be pushed
- If any users have opted out or had their mappings expire, this will allow you to remove them from your mapping tables since they will no longer be included in the full extract

If you'd like to receive these refreshes, please check with your ID5 representative

Mapping Table Refresh File Location and Format

Both Single-File and Per-Partner Files are supported. The data files will be stored in a different location than the incremental files, allowing you to process them separately for weekly/monthly refreshes. The files will be pushed to:

Per-Partner File Location

s3://[bucket]/full-extract/[datetime]-[clientname]/[match partner GVLID]/[filename].csv

Example: s3://bucket/full-extract/2025-12-16T14:57:17.765798953-clientname/32/b3a1f108-0e1f-7cee-ad8a-c81a8ba3a68e.csv

Daily Delivery

Data is delivered daily to a designated prefix at the root of your S3 bucket, organized into the following subdirectory structure:
`s3://bucket/prefix/region={region}/date=yyyy-mm-dd/otherPartnerGvlId={gvlId}/`

- **region** *(optional)* — geographic region
- **date** — delivery date in `yyyy-mm-dd` format
- **otherPartnerGvlId** — Match Partner identifier, based on the partner's Global Vendor List (GVL) ID. If Match Partner does not have a GVL ID, will default to the ID5 Partner ID.

Each terminal directory contains .parquet files (compressed with `zstd`) that contain *all* matches seen in the previous day between your UIDs and the requested Match Partner's UIDs.



Available Match Partners are defined by contract between you and ID5. To change the list of Match Partners, please reach out to your ID5 representative.

File Format

Column	Type	Description
partnerUid	String	Your User ID for this user
otherPartnerUid	String	The Match Partner's User ID for this user
lastSeen	Integer	Last seen timestamp for cookie sync pair

Example Output File

partnerUid	otherPartnerUid	lastSeen
550e8400-e29b-41d4-a716-446655440000	100001874214904 58	1774532889
d2a8378f-fe56-4ec2-96d1-3c05df02bb48	1000002629570693845	1774523875

Cleaning Up / Deleting Old Files

By default, ID5 does not delete any files we place in the S3 bucket. When we push files to the bucket, we perform a `sync` operation. This means that if you delete a file from the S3 bucket but it still exists on the ID5 servers, we will re-upload the file to the bucket.

We keep files on our server for **approximately 30 days**.

If your ETL process includes deleting files from the bucket, please let us know so we can work together on a solution that meets your needs.



We recommend deleting only files older than 30 days to avoid any issues. We can also automatically delete old files from the bucket that we have already removed from our servers if you'd like.

Matching File Transfer

10/24/2024 8:01 am EDT

Overview

ID5's Matching File-Transfer protocol converts a file containing a list of your UIDs to a file containing the equivalent list of UIDs for one or more Match Partners. Available Match Partners are defined by contract between you and ID5. To change the list of Match Partners, please reach out to your ID5 representative.

1. You will upload a file containing a list of your UIDs to the S3 bucket in the appropriate `inbox` directory
2. ID5 will process the file and return to the `outbox` in the S3 bucket a new file with a list of your match partner's UIDs for every user we could match

ID5 creates `inbox` and `outbox` folders in the shared S3 bucket. Files to be processed must be uploaded into the `inbox` folder within a sub-directory for the match partner you would like to match to and named with the extension `.csv`. The sub-directory should be named by the match partner's Global Vendor List ID. A scheduled task on ID5's side will process the inbox files and create a file containing Match Partner UIDs with an identical name in the `outbox` folder within a sub-directory for the match partner. The original file will be moved into a `processing` directory when the job is started and will remain untouched and in its original form. You may delete these files at any point. We recommend that files be named with the date and your internal segment id to ensure they are unique.

> Any file pushed into the inbox will be fully processed only once and updates to the same file will not be processed.

Directory Structure

One directory within the inbox and outbox will be created for each Match Partner, containing the corresponding list of UIDs for this platform. The Match Partner directories will be named based on the partner's Global Vendor List ID.

We also support using the ID5 Account Number instead of the GVL ID, prefaced by `ID5-` (e.g. `ID5-43`). Reach out to us if you'd like to use this method so we can give you your Match Partners' ID5 Account Numbers.

```
S3://[bucket]/inbox/[match partner A GVL ID]/20220128_mysegmentidA.csv
S3://[bucket]/inbox/[match partner A GVL ID]/20220129_mysegmentidA.csv
S3://[bucket]/inbox/[match partner B GVL ID]/20220129_mysegmentidB.csv
```

The above files will be processed and saved in

```
S3://[bucket]/outbox/[match partner A GVL ID]/20220128_mysegmentidA.csv
S3://[bucket]/outbox/[match partner A GVL ID]/20220129_mysegmentidA.csv
S3://[bucket]/outbox/[match partner B GVL ID]/20220129_mysegmentidB.csv
```

File Format

Inbox

Column	Type	Description
UID	String	Your UID for this user

> The maximum file size for inbox files is 1GB. If your files are larger, please split them before uploading or they may not be processed correctly.

Outbox

Matched Users Only

Generally, the outbox contains a single column containing the user IDs of the Match Partner that were requested. This is the default for all File-based UID Matching clients.

Column	Type	Description
UID	String	Match Partner's UID for this user

Pairs Output

In some cases, clients will get access to the "pair" and not just the Match Partner IDs that were requested. In this case, the output contains both the original User ID you requested and, if matched, the Match Partner's User ID, separated by a `|`.

Column	Type	Description
UID	String	Your UID for this user
UID	String	Match Partner's UID for this user

> Enabling this setting has a commercial impact, so please speak to your Account Manager if you'd like to discuss this integration method.

Example Files

Inbox Sample

```
$ cat ./inbox/8/20220128_mysegmentidA.csv
4822467914551922867
4700461554370241491
4572858233646453041
5952365002187240239
3896410401262240413
3827513351211796092
```

Outbox Sample

Matched Users Only

```
$ cat ./outbox/8/20220128_mysegmentidA.csv
49240e08-8de6-4deb-b060-6867684a3fd2
9aafa5ab-d867-4e1c-b4e3-1be6747ab097
a6b47ef8-7957-437f-a5fc-d0077399b1ee
06bf16ce-1fd9-4dca-849d-8144c887ad35
eddee745-33a8-49ed-ae63-1707fee2f393
76692099-fc23-4824-a6f8-3b8c60550c6b
```

Pairs Output

```
$ cat ./outbox/8/20220128_mysegmentidA.csv
4822467914551922867|eddee745-33a8-49ed-ae63-1707fee2f393
4700461554370241491|9aafa5ab-d867-4e1c-b4e3-1be6747ab097
4572858233646453041|49240e08-8de6-4deb-b060-6867684a3fd2
5952365002187240239|06bf16ce-1fd9-4dca-849d-8144c887ad35
3896410401262240413|76692099-fc23-4824-a6f8-3b8c60550c6b
3827513351211796092|a6b47ef8-7957-437f-a5fc-d0077399b1ee
```

Reverse UID Lookups

If you need to convert a list of your Match Partners' UIDs to your own (rather than converting your UIDs to your Match Partners', as detailed above), you can simply adjust the names of root directories listed to the following: `inbox-r`, `outbox-r`, and `processing-r`. You can use both sets of directories in parallel if you have different use cases for different partners. All directory structures, file formats, and processes, other than the names of the three root directories previously mentioned, remain the same for both.

Cleaning Up / Deleting Old Files

By default, ID5 does not delete any files we place in the S3 bucket. Once you have processed a file in the `outbox`, you may delete it. You may also wish to delete files in the `processing` directory once all match jobs have been completed. However, if you remove an in-progress file from `processing` before the job is complete, the job may fail or have only partial results.

Read First: Privacy Policy Requirements

02/01/2026 1:41 pm EST

1. Introduction& Disclaimer

Overview

Transparency is the foundation of the ID5 ID. To ensure the ID5 ID Service operates securely and in compliance with global data protection standards, ID5 requires all partners and their downstream parties (collectively, "**Partners**") maintain privacy notices that accurately describe the scope and technical reality of the data flows.

Contractual Status

The standards below constitute mandatory requirements for all Partners utilizing the ID5 ID Service. Depending on the specific agreement you executed with ID5, these standards constitute:

- "ID5 ID Requirements" as defined in your ID5 ID Agreement;
- "ID5 ID Site Offering Requirements," "ID5 IdentityCloud Requirements," or similar (for Partners on the legacy ID5 ID Site Agreement, MSA, or other agreement with ID5); OR
- Any similar term associated with your use of the service in which signals are sent to ID5, for ID5 to return an encrypted ID5 ID in real time (the "**ID5 ID Service**").

(collectively, the "**ID5 ID Requirements**" and the "**Agreement**" as applicable).

Per your Agreement, Partners must comply with the ID5 ID Requirements to activate and maintain the ID5 ID Service. Failure to implement these disclosures constitutes a material breach of your Agreement with ID5, puts both the Partner and ID5 at risk, and may trigger your indemnification obligations to ID5 for any resulting claims or regulatory fines.

> ⚠ IMPORTANT OPERATIONAL DISCLAIMER
>
> ID5 Technology Ltd. provides this document to define the operational and technical parameters for utilizing the ID5 ID Service. This content specifies what must be disclosed to users to ensure the data signals sent to ID5 are valid.
>
> *This document **DOES NOT CONSTITUTE LEGAL ADVICE**. Privacy laws (including the CCPA, CIPA, VPPA, and GDPR) are complex and fact-specific. You are solely responsible for consulting your own legal counsel to ensure your privacy policy, user notices, and consent mechanisms comply with all applicable laws and accurately reflect your specific data practices.*

2. US Publisher Requirements

Applicability: Partners with traffic originating from the United States.

To mitigate risks associated with evolving US privacy litigation - including claims regarding real-time data processing ("Wiretapping"/"Trap and Trace", e.g., CIPA) and video privacy (i.e., VPPA) - and to comply with state privacy laws (CCPA/CPRA and other state laws), Partners must include the following substantive elements in their Privacy Policy:

A. "Sale" and "Sharing" Classification

- **Requirement**: Explicitly state that the transmission of personal data (e.g., IP, Hashed Email, MAIDs) to ID5 constitutes a "Sale" (transfer for value) and/or "Sharing" (transfer for cross-context behavioral advertising) under applicable US State Laws for ID5 to process the data as a "Third Party" or "Business".
    - **Note**: Do NOT classify ID5 as a Processor or Service Provider. As described in your Agreement, ID5 retains the right to use the data for its own purposes (including Graph Building), which creates a "Third Party" or "Business" relationship under California law.
- **Signal Flow:** Ensure ID5 is included in the signal flow of the Partner's "Do Not Sell/Share My Personal Information" mechanism (e.g., GPP string, GPC Signal) so that ID5 is able to ascertain when a particular user has opted-out and automatically suppress processing for opted-out users.

B. Disclosure of "Real-Time" Interception

To ensure transparency regarding automated technologies, you must explicitly state that third-party vendors operate on the digital property concurrently with the user's visit.

- **Requirement:** Disclose that third-party vendors (specifically ID5) are authorized to access the user's device, intercept, and collect signals derived from pseudonymous device info "in real-time" and "during the user's interaction" with the page.
- **Drafting Example:** *We utilize third-party tracking technologies [link to ID5] that operate concurrently with your visit to intercept, monitor, and record your interactions with our page, including the collection of device identifiers and IP addresses, for the purposes of identity resolution and security.*

C. Strict Purpose Bifurcation (Identity vs. Security)

The Partner's privacy policy must legally distinguish between data collected for identity/advertising and data collected for security. This distinction is a condition of ID5 accepting the data.

- **Identity & Advertising:** Disclose that IP Addresses, Hashed Emails, and User Agent strings are shared with ID5 for "Identity Resolution," which enables "Profiling", "Automated Decisioning" "Cross-Device Graphing," and "Targeted Advertising" (or "Cross-Context Behavioral Advertising").
- **Security & Error Detection:** Disclose that Page URLs (Uniform Resource Locators) and Timestamps are collected by ID5 specifically for "Security," "Fraud Prevention," and "Error Detection" purposes. Do **not** list ID5's collection of Page URLs under "Advertising" or "Profiling" purposes. This distinction is critical for compliance with video privacy and pen register statutes.
- **Restriction on Video URLs:** You must NOT transmit full URLs to ID5 if those URLs contain video titles or content descriptions (e.g., site.com/video/how-to-treat-depression) unless you have obtained a standalone, VPPA-compliant consent form.

D. Mandatory Links

- **Requirement:** Include a direct link to the ID5 Platform Privacy Policy: https://id5.io/platform-privacy-policy
- **Requirement:** Include a direct link to the ID5 Opt-Out: https://id5-sync.com/privacy

3. "Consent" Jurisdiction Requirements (e.g., EEA, UK, Quebec, Brazil)

Applicability: Partners with traffic originating from the EEA, UK, Switzerland, Brazil, or Quebec. These jurisdictions are referred to as "Opt-In Jurisdictions" in the ID5 DPAs.

A. ID5 Status: Independent Controller

- **Requirement:** The Partner's Privacy Policy and CMP must identify **ID5 Technology Ltd.** as a data partner and an Independent Controller and/or third-party (not a Processor). ID5 independently determines the means and purposes of building the identity graph.
    - *Note:* Identifying ID5 as a "Processor" or "Service Provider" is inaccurate and is a material breach of your Agreement, as it legally restricts ID5's ability to maintain its Identity Graph.

B. Legal Basis & TCF Compliance

- **For TCF Participants:** Partner must configure its Consent Management Platform (CMP) to signal consent/legitimate interest for **Vendor ID #131 (ID5)** for the appropriate TCF Purposes required by ID5's TCF Global Vendor List.
- **For Non-TCF Participants:** Partner must obtain legally valid opt-in consent for the transmission of IP addresses and Partner signals to ID5 for the specific purpose of creating a persistent advertising ID before the ID5 ID Service initializes on the page.

Note: ID5 must grant a specific exception for any partner processing data in the EEA which does not utilize a TCF-compliant CMP. Please speak to your account manager or primary point of contact at ID5, or email privacy@id5.io for more information.

C. International Data Transfers

- **Requirement:** Disclose that personal data may be transferred to the United Kingdom for processing. ID5 relies on the EU-US Data Privacy Framework, UK Data Bridge, and/or Standard Contractual Clauses to safeguard these transfers.

4. Rest-of-World Requirements

Applicability: Partners with global traffic (outside the US, EU, UK)

If you utilize a single global policy, you must include a "Vendor/Partner" section specifically covering**ID5 Technology Ltd** that addresses the following:

Disclosure Element	Required Standard
Role	Independent Controller (Global/EU); Third Party (US).
Timing	Data collection occurs "in real-time" or "concurrently" during the user interaction, *except*: in jurisdictions where affirmative opt-in is required, data collection occurs only after the user's affirmative consent (including via CMP).
Identity Data	Data such as IP Address, User Agent, Timestamp, and Unique User IDs are used for Identity Resolution, Cross-Device Graphing, and Ad Targeting. In some cases, partners may elect to send Hashed Emails, MAIDs, and/or Cookie IDs.

User Rights	Must link to ID5 Privacy Policy and ID5 Opt-Out.

DSAR - Data Deletion API

10/29/2026 1:25 pm EDT

Overview

The Privacy Requests API allows partners to automate the processing of consumers' Data Subject Access Requests related to data stored by ID5. The Data Deletion service handles deletion of any data associated with the consumer, if any.

1. The first step is to POST the Data Deletion Request with the necessary information for ID5 to process it.

2. The response to this request will contain a job ID that can then be queried using theStatus API to see the status of the request.

Data Deletion Request

Example URL

> https://api.id5-sync.com/partners/v1/{PARTNER}/privacy/requests/deletion?token={TOKEN}

Request Type

HTTP POST with JSON body

Request Headers

`Content-Type: application/json; charset=UTF-8`

Partner Number

The value `{PARTNER}` in the above example url will be replaced by an ID5-provided Partner Number. This value will be static for you once we set you up in our system. You may use the example URL above during testing with the Partner Number 173. If you haven't already been assigned a Partner Number, please contact us to request one.

Available Parameters

Querystring

Name	Required	Description
token	x	A permanent security token provided by ID5. Please contact ID5 at support@id5.io for your token.

Request Body

Name	Required	Description

Name	Required	Description
email		Email of the consumer that is requesting their data be deleted. Should be in plain text format or a `sha256` hash of the email. `sha256` should be calculated as per these guidelines. It is required to provide one of the fields: `email`, `id5id`, `maid`, `partnerUid`.
id5id		ID5 ID of the consumer that is requesting their data be deleted. Should be in encrypted (`ID5*`) or decrypted (`ID5-`) format. It is required to provide one of the fields: `email`, `id5id`, `maid`, `partnerUid`.
maid		Mobile Advertising ID (either `gaid` or `idfa`) of the consumer that is requesting their data be deleted. Should be in text format, eg: `580d2b4c-29a5-7a7b-85dc-44132c023ac8`. It is required to provide one of the fields: `email`, `id5id`, `maid`, `partnerUid`.
partnerUid		Partner specific user ID - the same that is send in `pd` string under key `5`. It is required to provide one of the fields: `email`, `id5id`, `maid`, `partnerUid`.
jurisdiction	x	The jurisdiction under which the request was made. See below for possible *case-insensitive* values
replyToEmail		Plain text email to which a response message will be sent indicating the results of the request. If no value is provided, no confirmation email will be sent

Possible `jurisdiction` values

- GDPR
- CCPA

Example Data Deletion Request

> POST: https://api.id5-sync.com/partners/v1/173/privacy/requests/deletion?token=abc123

```
{
  "email": "ca372fb85148700fa88095e3492d3f9f5beb43e555e5ff26d95f5a6adc36f8e6",
  "id5id": "ID5*j0EDhnOeLA7GJ9KXt05cszkCOLRHRyqVRKNx4Wo9iEtZYPJcnd32YHQ9MjAOLR0UWWEnBhCvzGoO5pggV
  "maid": "580d2b4c-29a5-7a7b-85dc-44132c023ac8",
  "partnerUid": "a-123456789",
  "jurisdiction": "GDPR",
  "replyToEmail": "joe.consumer@gmail.com"
}
```

Data Deletion Response

Successful Response

HTTP Status Code

`200 OK`

Response Body

Name	Description
id	The job ID of the Data Deletion Request

Example Successful Response

```
{
  "id": "a8b6ccc4ee35ddaf5a5bb0f5c696dbd3"
}
```

Unsuccessful Response

Schema

```
{
  "error": {
    "code": "{CODE}",
    "type": "{TYPE}"
    "message": "{MESSAGE}",
  }
}
```

Error Descriptions

HTTP Status Code	Code	Type	Message
401	api_token_invalid	authentication_error	No API token provided
400	partiner_id_invalid	authentication_error	Invalid partner id `{PARTNER_ID}` provided
403	api_token_not_authorized	authentication_error	Api token `{TOKEN}` does not have access to this resource
400	request_format_invalid	invalid_request_error	`application/json; charset=UTF-8` POST required
400	request_format_invalid	invalid_request_error	Missing required JSON body
400	user_objects_invalid	validation_error	Missing required parameter 'jurisdiction'
400	user_objects_invalid	validation_error	Missing one of parameters: ['id5id', 'email', 'maid']
400	user_objects_invalid	validation_error	Provided ID5ID `[ID5ID]` cannot be decrypted
400	user_objects_invalid	validation_error	Provided ID5ID `[ID5ID]` is not a valid one
400	user_objects_invalid	validation_error	Provided maid `[MAID]` is not a valid one
403	api_rate_limit_error	rate_limit_error	Daily limit of 5 **email** requests was reached
403	api_rate_limit_error	rate_limit_error	Daily limit of 5 **ID5ID** requests was reached
403	api_rate_limit_error	rate_limit_error	Daily limit of 5 **MAID** requests was reached

HTTP Status Code	Code	Type	Message
403	api_rate_limit_error	rate_limit_error	Daily limit of 1 **partnerUID** request was reached
403	api_rate_limit_error	rate_limit_error	Daily limit of 15,000 requests per **partner** reached

Example Unsuccessful Response

```
{
  "error": {
    "code": "api_token_invalid",
    "type": "authentication_error"
    "message": "No API token provided",
  }
}
```

Status Request

Example URL

> https://api.id5-sync.com/partners/v1/{PARTNER}/privacy/requests/{PRIVACY_REQUEST_DELETION_JOB_ID}?token={TOKEN}

Request Type

HTTP GET

Available Parameters

URL Path

Name	Required	Description
PARTNER	x	The Partner Number provided by ID5 used in the Data Deletion Request
PRIVACY_REQUEST_DELETION_JOB_ID	x	The `id` from the Data Deletion Request response body

Querystring

Name	Required	Description
token	x	A permanent security token provided by ID5. Please contact ID5 at support@id5.io for your token. This is the same token as used in the Data Deletion Request

Example Request

> GET: https://api.id5-sync.com/partners/v1/173/privacy/requests/a8b6ccc4ee35ddaf5a5bb0f5c696dbd3?token=abc123

Status Response

Successful Response

HTTP Status Code

`200 OK`

Response Body

Name	Description
id	The job ID of the Data Deletion request
jobStatus	The current status of the job. See below for possible values
processingResult	Result of processing the DSAR. See below for possible values. Returned only when jobStatus is DONE
emailSentUnixTimestamp	Unix timestamp when reply-to email was sent. Will be null if no email was sent yet (because it was not processed yet or because there was no `replyToEmail` defined in the request)

Possible `jobStatus` values

- CREATED
- STARTED
- FAILED
- DONE
- SENT
- SEND_FAILED
- CANCELLED

Possible `processingResult` values

- DELETE_DELETED
- DELETE_NO_DATA
- NONE

Example Successful Response

```
{
  "id": "a8b6ccc4ee35ddaf5a5bb0f5c696dbd3",
  "jobStatus": "SENT",
  "processingResult": "DELETE_DELETED",
  "emailSentUnixTimestamp": 1661933465437
}
```

Unsuccessful Response

Schema

```
{
 "error": {
  "code": "{CODE}",
  "type": "{TYPE}"
  "message": "{MESSAGE}",
 }
}
```

Error Descriptions

HTTP Status Code	Code	Type	Message
401	api_token_invalid	authentication_error	No API token provided
400	partiner_id_invalid	authentication_error	Invalid partner id `{PARTNER_ID}` provided
403	api_token_not_authorized	authentication_error	Api token `{TOKEN}` does not have access to this resource
400	user_object_invalid	validation_error	provided job id is not a valid UUID
404	user_objects_invalid	invalid_request_error	provided job UUID not found
500	internal_id5_error	api_error	Internal error id: `{ID}`

Example Unsuccessful Response

```
{
   "error": {
      "code": "user_object_invalid",
      "message": "provided job UUID not found",
      "type": "invalid_request_error"
   }
}
```

Mobile Opt-in/Opt-out API (deprecated)

02/09/2026 6:46 am EST

This endpoint was deprecated on 9th February 2026. Please use the Opt-Out API

Overview

The Mobile Opt-in and Opt-out API allows users to opt in and out of ID5 tracking via MAIDs.

Opt Out Request

Example URL

> https://api.id5-sync.com/privacy/mobile-optout?ifa={maid}&ifa_type={ifa_type}

Request Type

HTTP POST

Request Headers

`Content-Type: application/json; charset=UTF-8`

Available Parameters

Querystring

Name	Required	Description
ifa	x	The Identifier for Advertising (IFA) ID in a UUID format (8-4-4-4-12) i.e. EA7583CD-A667-48BC-B806-42ECB2B48607
ifa_typex		The type of IFA i.e. GAID, AAID

Example Mobile Opt Out Request

> https://api.id5-sync.com/privacy/mobile-optout?ifa=EA7583CD-A667-48BC-B806-42ECB2B48607&ifa_type=GAID

Mobile Opt Out Response

Successful Response

HTTP Status Code

`200 OK`

Response Body

Name	Description
optout	true if opting out, false if opting in
success	true if operation was successful, false otherwise

Example Successful Response

```
{
  "optout": true,
  "success": true,
  "message": ""
}
```

Unsuccessful Response

Example Unsuccessful Response

```
{
  "optout": true,
  "success": false
  "message": "ifa query parameter must not be null and must have UUID format"
}
```

Opt In Request

Example URL

> https://api.id5-sync.com/privacy/mobile-optin?ifa={maid}&ifa_type={ifa_type}

Request Type

HTTP POST

Request Headers

`Content-Type: application/json; charset=UTF-8`

Available Parameters

Querystring

Name	Required	Description
ifa	x	The Identifier for Advertising (IFA) ID in a UUID format (8-4-4-4-12) i.e. EA7583CD-A667-48BC-B806-42ECB2B48607
ifa_type	x	The type of IFA i.e. GAID, AAID

Example Mobile Opt Out Request

> https://api.id5-sync.com/privacy/mobile-optin?ifa=EA7583CD-A667-48BC-B806-42ECB2B48607&ifa_type=GAID

Mobile Opt Out Response

Successful Response

HTTP Status Code

`200 OK`

Response Body

Name	Description
optout	true if opting out, false if opting in
success	true if operation was successful, false otherwise

Example Successful Response

```
{
  "optout": false,
  "success": true,
  "message": ""
}
```

Unsuccessful Response

Example Unsuccessful Response

```
{
  "optout": false,
  "success": false
  "message": "ifa query parameter must not be null and must have UUID format"
}
```

User Rights Propagation

03/31/2026 8:13 am EDT

User Rights Propagation

Summary

This document details ID5's process for disseminating user opt-out and data deletion requests to its partners. As part of ID5's ongoing commitment to robust privacy practices, this system ensures that when users exercise their right to opt-out or request data deletion from ID5, these choices are efficiently communicated. The mechanism involves securely transferring lists of affected ID5 IDs to partners via dedicated S3 buckets. This enables partners to honor these requests in a timely manner, aligning with regulatory expectations and reinforcing user trust.

Background

In the digital advertising ecosystem, respecting user privacy choices is paramount. ID5's opt-out propagation initiative is a critical function designed to uphold these principles. When an individual interacts with ID5 and chooses to opt-out of data processing or requests the deletion of their data, ID5 initiates a process to ensure this preference is communicated downstream. This "streaming" of opt-out and deletion information is fundamental to ensuring that partners who receive ID5 data can also comply with user requests and meet their own regulatory obligations under frameworks such as GDPR, CCPA, and others. The process is designed to be reliable and consistent, providing partners with the necessary information to take appropriate action.

Propagation Details

- A list of ID5 IDs for users who have opted out or requested data deletion is produced.
- Frequency: This data is updated every 30 minutes on a rolling basis for opt-outs and on a hourly basis for deletion requests, ensuring partners receive timely information. Further data deletions can be occasionally produced due to privacy requests cascaded to ID5 on an unpredictable schedule.

Pull method (preferred for this application)

ID5 makes content available to you through the ID5-owned AWS S3 bucket "id5-privacy" using the methodologies described in https://wiki.id5.io/docs/client-data-sharing-specification

Push method

In case you are not using the AWS cloud, we can push the data to an S3-compatible Object Storage you own a bucket within. The deliveries will happen on a regular schedule (30 minutes) and will contain the feeds described below

Partner Responsibility & S3 Bucket Setup

- Action Required: S3 Bucket Configuration: To receive opt-out and deletion data, each partner must use one of the methods described above. It is crucial that partners proactively contact the ID5 support team (e.g., support@id5.io or your designated ID5 contact) to initiate this setup process or to confirm that your preferred method of integration is correctly configured for this purpose. This step

is mandatory to ensure the secure and reliable delivery of these privacy-related signals.

- Timely Processing: Partners are expected to regularly ingest and process this data. This ensures that end-users' choices are appropriately actioned within the partner's systems, thereby upholding user privacy and meeting regulatory compliance needs.
- Compliance: The accurate and timely processing of these ID lists is essential for partners to fulfil their own legal and contractual obligations regarding data privacy.
- This documentation aims to provide clarity on the opt-out propagation mechanism, facilitating a smooth and compliant process for all parties involved. Please ensure your technical teams review this information and that your integration is correctly configured.

Privacy Feeds — Locations & Delivery

ID5 delivers three privacy-related feeds. Each feed can be delivered to the **id5-privacy** S3 bucket or **pushed to a bucket you own** with a prefix of your choice.

> **Note:** Files may be delivered with some delay after the hour indicated in the path, depending on system load.

> Customisable elements are marked with throughout this page.

Opt-Out Feed

Cadence	Every 30 minutes
ID5-hosted path	s3://id5-privacy/user-opt-outs/type=id5/date=YYYY-MM-DD/hour=hh/HHMMSS.csv
Client-hosted path	s3:// your-bucket/ your-prefix/type=id5/date=YYYY-MM-DD/hour=hh/HHMMSS.csv

File Naming Convention

Within each hourly partition, a new CSV file is generated every 30 minutes. File names follow the pattern `HHMMSS.csv` (hours, minutes, seconds in UTC), reflecting the start of the window they cover.

Example: A file covering the window starting at 11:00 PM UTC on 9 February 2026 would be located at:

```
s3://your-bucket/your-prefix/type=id5/date=2026-02-09/hour=23/230000.csv
```

The next file, 30 minutes later, would be named `233000.csv` in the same hourly partition.

Data Deletions Feed

Property	Detail
Cadence	Hourly

Property	Detail
ID5-hosted path	s3://id5-privacy/user-deletion-requests/type=id5/date=YYYY-MM-DD/hour=hh/<filename>.csv
Client-hosted path	s3:// your-bucket/ your-prefix/type=id5/date=YYYY-MM-DD/hour=hh/<filename>.csv

File Naming Convention

Within each hourly partition, CSV files are assigned a random, system-generated name (not meaningful from a business perspective).

Example: A file produced for the 10 PM UTC hour on 3 March 2026 might be located at:

```
s3://your-bucket/your-prefix/type=id5/date=2026-03-03/hour=22/part-3c127890-162e-4caa-bd4e-2fd2bff2278f-0.csv
```

On-Demand Data Deletions Feed

Property	Detail
Cadence	Best-effort; no fixed schedule
ID5-hosted path	s3://id5-privacy/on-demand-data-deletions/type=id5/date=YYYY-MM-DD/hour=hh/<filename>.csv
Client-hosted path	s3:// your-bucket/ your-prefix/type=id5/date=YYYY-MM-DD/hour=hh/<filename>.csv

File Naming Convention

Same as the Data Deletions feed: files are grouped by hour and assigned a random, system-generated name.

Example:

```
s3://your-bucket/your-prefix/type=id5/date=2026-03-03/hour=17/part-00000-tid-1681382479627015163-88fec790-25f8-48ed-befe-66d3ec5b2409-1695-1-c000.csv
```

> **Note:** Because this feed operates on a best-effort basis, files may be delivered at any point during the day they are produced in.

Summary of Customisable Elements

Element	Description
S3 bucket	You may receive feeds in your own S3 bucket instead of the ID5-hosted bucket.
Path prefix	When delivering to your bucket, the prefix preceding `/type=id5/…` is fully configurable.

Data Format and Structure

The opt-out and deletion data is delivered in Comma Separated Values (CSV) format, chosen for its simplicity and broad compatibility. The data is organized within a specific directory structure described above, making it easy to locate and process.

- File Content: Each CSV file contains a list of ID5 IDs that correspond to users who have opted out or requested data deletion. Each ID5 ID will typically be on a new line within the CSV file.
- Example CSV content:

```
ID5-xxxxxxxxxxxxxxxx1
ID5-yyyyyyyyyyyyyyyy2
ID5-zzzzzzzzzzzzzzzz3
```

Opt-out API

12/04/2025 5:06 am EST

Overview

The Privacy API Opt-out endpoint allows authorized partners to opt out users from ID5 tracking using various privacy identifiers including ID5 ID, MAIDs, email addresses, and hashed emails. The system handles multiple identifier types and automatically opts out associated IDs.

Opt Out Request

Example URL

> https://api.id5-sync.com/privacy/v2/api/optout?partnerId={partnerId}&apiToken={apiToken}

Request Type

HTTP POST

Request Headers

`Content-Type: application/json; charset=UTF-8`

Available Parameters

Querystring

Name	Required	Description
partnerId	x	The partner's unique identifier (integer)
apiToken	x	The API token for authentication

Request Body (JSON)

Name	Type	Required	Description
id5Id	string		The ID5 identifier (ID5-prefixed or ID5* encoded string)
maid	string		Mobile Advertising ID (MAID) in UUID format
maidType	string		Type of MAID (e.g., GAID, IDFA, AAID)

Name	Type	Required	Description
email	string		Plain text email address (we will hash this email)
hem	string		Hashed email (HEM) identifier (SHA-256)

Note: At least one identifier field must be provided in the request body.

ID5 ID Formats Supported

The system accepts ID5 IDs in multiple formats:

- ID5-prefixed format (decrypted ID5ID): ID5-ZHMOMopHd-M3j2tMtnDyZzOPPzVB_4Zj73nGFlJ5UQ
- ID5* encoded format (encrypted ID5ID): ID5*JAdI2WT2d7kLi4l4ICTmUX3137PpzqvzlNb6OUAl0MX96NChTqVI0oGD7idrCmpw

Example Opt Out Request

> https://api.id5-sync.com/privacy/v2/api/optout?partnerId=12345&apiToken=your-api-token

Example 1: Opt out by ID5 ID (decrypted)

```
{
 "id5Id": "ID5-ZHMOMopHd-M3j2tMtnDyZzOPPzVB_4Zj73nGFlJ5UQ"
}
```

Result: Decodes to UUID and stores opt-out for the decoded ID

Example 2: Opt out by MAID

```
{
 "maid": "dc7e803b-fe08-430b-aded-3f4942929000",
 "maidType": "GAID"
}
```

Result:

- Looks up associated ID5 ID for the MAID
- Stores opt-out for the associated ID5 ID
- Stores MAID in mobile opt-out repository

Example 3: Opt out by email

```
{
 "email": "test@test.io"
}
```

Result:

- Converts email to HEM (SHA-256)
- Looks up associated ID5 ID from hard signal storage
- Stores opt-out for the associated ID5 ID
- Records email opt-out

Example 4: Opt out by hashed email (HEM)

```
{
 "hem": "1958e9b00a8319f05cb46cf06b397e724c5c66a2ff2633c7f0884d6a57432af8"
}
```

Result:

- Looks up associated ID5 ID from hard signal storage
- Stores FULL opt-out for the associated ID5 ID
- Records email opt-out

Example 5: Opt out with multiple identifiers (same associated ID)

```
{
 "id5Id": "ID5-ZHMOMopHd-M3j2tMtnDyZzOPPzVB_4Zj73nGFlJ5UQ",
 "maid": "b33fa4b2-5d49-4014-835e-255c462bc9be",
 "maidType": "IDFA"
}
```

Result:

- Stores opt-out for the ID5 ID
- Stores MAID opt-out (if MAID lookup returns same ID5 ID)
- Stores MAID in mobile opt-out repository

Example 6: Opt out with multiple identifiers (different associated IDs)

```
{
 "id5Id": "ID5-ZHMOMopHd-M3j2tMtnDyZzOPPzVB_4Zj73nGFlJ5UQ",
 "maid": "a95a6929-8bfc-445d-9625-7c6593c1e950",
 "maidType": "GAID",
 "email": "test@test.io"
}
```


Result: Stores opt-out for:

- The provided ID5 ID
- The ID5 ID associated with the MAID
- The ID5 ID associated with the email (from hard signal storage)

Plus records MAID and email opt-outs

Opt Out Response

Successful Response

HTTP Status Code

`200 OK`

Response Body

The response body will be empty for successful opt-out operations.

Error Responses

Authentication Error

HTTP Status Code: `401 Unauthorized`


```
{
 "code": "api_token_invalid",
 "message": "Wrong api token for partnerId",
 "type": "authentication_error"
}
```


Missing Required Parameter

HTTP Status Code: `400 Bad Request`

```
{
 "code": "request_parameter_invalid",
 "message": "Required request parameter 'partnerId' is not present",
 "type": "invalid_request_error"
}
```

Important Implementation Notes

Associated ID Lookup

1. **MAID Association:** When a MAID is provided, the system looks up any previously associated ID5 ID and opts that out as well
2. **Email/HEM Association:** The system checks hard signal storage for ID5 IDs associated with the email hash
3. **Multiple Associations:** When multiple identifiers are provided, all associated ID5 IDs are opted out

Email Processing

- Plain text emails are normalized (see normalizing emails) and hashed using SHA-256
- The system looks up associations using the hashed value
- Both `email` and `hem` fields are treated equivalently after hashing

Idempotency

The opt-out operation is idempotent - calling it multiple times with the same identifiers will have the same effect as calling it once.

Rate Limiting

Please refer to your partner agreement for specific rate limiting details. Excessive requests may result in temporary blocking.

Support

For technical support or questions about the Privacy API, please contact your ID5 representative.

ATT Signal Processing (iOS)

05/19/2026 9:39 am EDT

ATT Signal Processing (iOS)

Apple's App Tracking Transparency (ATT) framework gives users control over whether apps may track them across other apps and websites. Because ID5 integrates via server-to-server (S2S), it processes only the data and signals that publishers route to our endpoint; ID5 does not independently access device-level ATT status. Publishers therefore retain full control over how ATT authorisation interacts with identity processing and must configure their integration to reflect their specific platform obligations.

Signal Processing Hierarchy

ID5 evaluates consent signals only where no opt-out has been detected. Where a user is not opted out, signals are evaluated in the following order of precedence:

1. **Consent string** — if a valid consent string is present in the request (IAB TCF, GPP, or ID5 Allowed Vendors), ID5 uses this as the primary basis for ID generation.

2. **ATT signal** — where no valid consent string is available, ID5 falls back to the device-level ATT authorisation passed via the `att` parameter

ID5 requires either a valid consent string or positive iOS device-level authorisation to generate and transmit an ID5 ID.

Passing the ATT Parameter

Publishers should retrieve the user's ATT authorisation status using Apple's AppTrackingTransparency framework and map it to ID5's `att` parameter before sending requests to the ID5 endpoint.

Name	Required	Type	Description	Format	Example
att	x (iOS requests)	boolean	If the user selected "Ask App not to Track", set the value to `1`, otherwise omit the field or set the value to `0`. Also omit the field for all non-IOS requests.	boolean	`1` / `0`

The att field should be used for **iOS requests only**, It's the iOS equivalent of GAID (Google Advertising ID) on Android, or RIDA on Roku.

See the Mobile In-App Integration page for full details on the `att` parameter.

Enforcing a Hard Stop on Denied ATT

To ensure that a denied ATT signal always blocks ID generation regardless of CMP consent, configure your integration to drop the call to the ID5 ID endpoint when a user denies ATT authorisation.

Campaign Measurement with ID5

04/01/2025 3:58 am EDT

ID5 is still expanding its support for campaign measurement and welcomes feedback.

ID5 enables event tracking and matching for campaign measurement and attribution. For instance, the ID5 ID, when paired with the ID5 graph, can be used to record and match events such as campaign impressions, clicks, and conversions. This is especially valuable for identifying incremental matches in environments where users are unauthenticated or in cookieless environments.

To track and connect events, the ID5 JS API should be integrated across all digital properties where event tracking is desired. Campaign metadata and unique event IDs, including impressions, clicks, and conversions, should be recorded alongside the decrypted ID5 ID. By combining the ID5 ID with the ID5 Graph, these events can be connected at both individual and household levels. Since ID5 leverages deterministic signals such as hashed emails, cookies, and MAIDs, along with probabilistic methods for reconciling user identity, ID5 facilitates a more comprehensive view of campaign performance and advertiser outcomes.

Example Use Case

Adentification: An advertiser or advertising ad server to match impression and conversion events for campaign measurement.

Implementation

1. Deploy the ID5 JS API on advertiser's digital properties. This can be limited to pages the users visit after a conversion event or applied across the entire site. To maximise match rates, it is recommended to pass partner data such as hashed email when available.

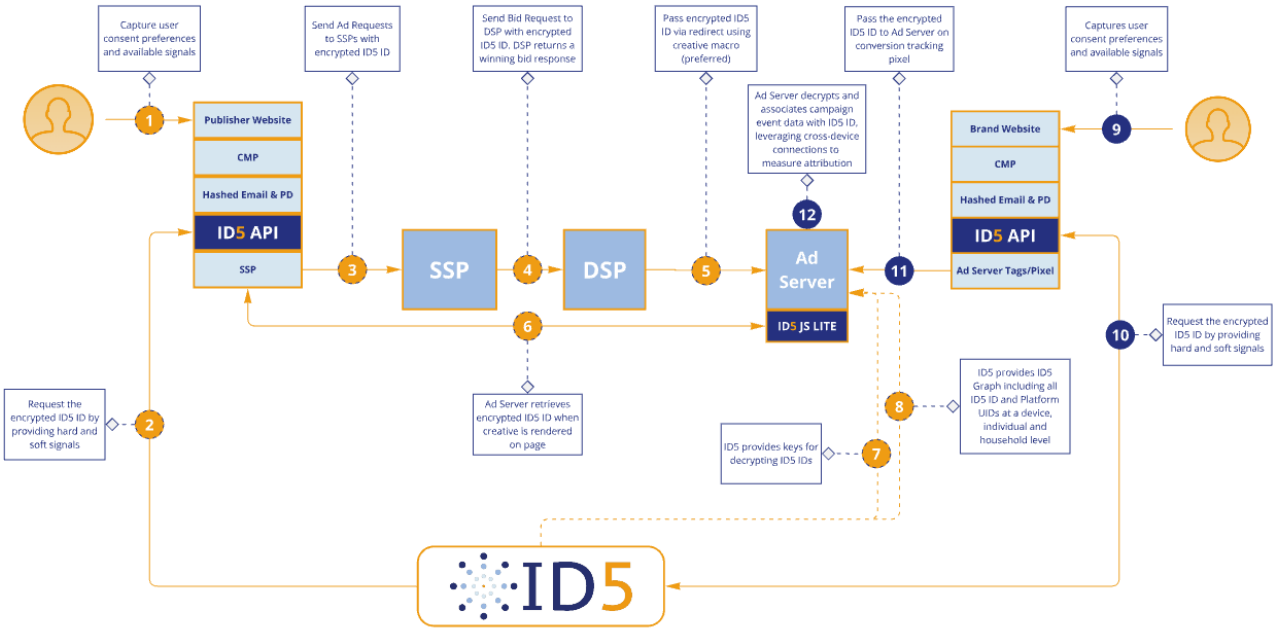
2. Integrate the ID5 JS API Lite in campaign creatives. Ensure a call is made during impression or click events.

3. Decrypt and store the ID5 ID alongside the campaign events.

4. Campaign events can be matched at the ID5 ID level, or, if using the ID5 Graph, at the individual level for more precise insights into campaign ROI.

The ID5 JS API Lite has limited functionality. It functions only when the full ID5 JS API is present on the page the creative is served on. It is permitted for use only when absolutely required in the case of deployment within a creative.

Schema


Capture user consent preferences and available signals
Send Ad Requests to SSPs with encrypted ID5 ID
Send Bid Request to DSP with encrypted ID5 ID. DSP returns a winning bid response
Pass encrypted ID5 ID via redirect using creative macro (preferred)
Pass the encrypted ID5 ID to Ad Server on conversion tracking pixel
Captures user consent preferences and available signals
Ad Server decrypts and associates campaign event data with ID5 ID, leveraging cross-device connections to measure attribution
Publisher Website
CMP
Hashed Email & PD
ID5 API
SSP
SSP
DSP
Ad Server
ID5 JS LITE
Brand Website
CMP
Hashed Email & PD
ID5 API
Ad Server Tags/Pixel
Request the encrypted ID5 ID by providing hard and soft signals
Request the encrypted ID5 ID by providing hard and soft signals
Ad Server retrieves encrypted ID5 ID when creative is rendered on page
ID5 provides keys for decrypting ID5 IDs
ID5 provides ID5 Graph including all ID5 ID and Platform UIDs at a device, individual and household level
ID5

ID5 Metadata

05/18/2026 11:19 am EDT

When ID5 provides an ID5 ID, we may also include certain metadata. Below you will find the different types of metadata we support and the values they could contain:

- Link Type
- A/B Testing Control Group

Link Type

The link type describes the method that we used to link this ID5 ID as a cross-domain identifier.

Possible Values

Value	Description
0	This ID is **not linked** with another ID
1	This ID is linked **probabilistically** with another ID
2	This ID is **deterministically** linked to another ID through a persistent signal such as a MAID, third-party cookie, partner-specific user ID, cross partner user ID, CTV ID, ID5 TrueLink signal, ID5 signature, or first-party cookie
3	This ID is linked by a **hashed email** with another ID

Field Name in eids.uids Object

`linkType`

Example eids.uids Object

```
{
   "id": "ID5*XYZXYZ....",
   "ext": {
      "linkType": 1
   }
}
```

A/B Testing Control Group

The A/B testing control group field indicates whether the A/B testing feature was enabled and the user was placed in the control group. In these cases, the ID would not be returned, so this field informs the receiver that there is no ID because of the A/B test rather than some other reason. We will not send this metadata if A/B testing was not enabled on the request.

Possible Values

Value	Description

Value	Description
true	A/B Testing was enabled and the user was in the control group. No ID will be available
false	The A/B Testing feature was enabled but the user was not in the control group

Field Name in eids.uids Object

`abTestingControlGroup`

Example eids.uids Object

```
{
   "id": "0",
   "ext": {
      "abTestingControlGroup": true,
      "pba": "izK5L4018hFFoR6I+ONgD1FdPQRsy11WIQOOjyzJKJAxtoUxmqw=="
   }
}
```

> Occasionally, we may include the extended field `pba`, which ID5 uses internally as Prebid Analytics metadata.

Real Time Bid Enrichment Service

05/28/2026 2:42 am EDT

What is ID5's Real-Time Bid Enrichment Service?

ID5's Real-Time Bid Enrichment is an optional, paid-for service that enables publishers and publisher monetization platforms to improve inventory addressability. When enabled, the ID5 Graph is used to enrich the EID object provided in ID5's response with SSP and DSP cookie IDs alongside ID5 IDs. By opting in, publishers or their platform partners can make their inventory more addressable to buyers through a responsible and transparent bid enrichment process that fully complies with the IAB Tech Lab's OpenRTB specification.

Key Benefits

- Enhanced audience addressability even in the absence of legacy identifiers
- Improved match rates with supply-side and buy-side partners, increasing the value of publisher inventory and therefore revenue

Requirements

1. Register with ID5

Register with ID5. If you don't already have an account with ID5, pleasevisit our website to sign up and request your ID5 Partner Number.

2. Review and Sign the Addendum for ID5 Bid Enrichment Services

Review the addendum for ID5 Bid Enrichment Services provided by your ID5 Account Manager.

3. Ensure your ID5 Integration meets the minimum requirements for Bid Enrichment Enablement

> To maximise value and match rates achieved by the **ID5 Real Time Bid Enrichment Service**, we recommend you pass all available signals e.g hashed emails (instructions) and upgrade your ID5 integration by deploying ID5's TrueLink solution.

ID5's Real Time Bid Enrichment Service can be enabled for all types of ID5 ID fetch integrations including:

- Prebid. js (minimum integration requirements)
- Prebid Server BETA (if your prebid server host has configured the ID5 user ID module)
- ID5 JS API
- Mobile In App S2S API
- CTV API
- Server Side API

4. Instruct ID5 to enable the Real Time Bid Enrichment service with your preferred configurations

Your ID5 Account Manager will provide you with a list of the currently supported configuration options for you to choose from and enable the service on an agreed date. Configurable options include:

Configurations	Values
Countries	e.g. UK, DE,

Match Partners	e.g. Magnite
Inserter	e.g. e.g. publisher.com This should be the same as the ownerdomain variable in an ads.txt file which designates the business domain that owns or operates the ad inventory.
Match Method	e.g. Inference, Browser Cookie Sync
Enriched vs Control group % split	90:10 default

5. Uplift Reporting



Partners with the real time bid enrichment service enabled are contractually obliged to provide ID5 with uplift reporting.

ID5 will automatically set up an A/B test with a 90:10 split between enriched and control groups. The % split can be customised upon client request. This means:

- 90% of eligible users: Enriched group - will receive enriched IDs
- 10% of eligible users: Control group - will not receive enriched IDs

Note this split is done independently of any other A/B testing the publisher might have configured. If the ID5 abtesting parameter is configured whilst bid enrichment is enabled, the enrichment/control group split will apply to the **normal group** that has the ID5 ID enabled.

Example

If a publisher enables ID5's AB Testing with an 85:15 split, then 85% of users are assigned to the **normal group** and 15% to the **control group**. If bid enrichment is also enabled with a 90:10 split, this applies only within the normal group: **90% of those users** will receive enrichment, while **10% will serve as the enrichment control group**. By running the two tests at the same time, the bid enrichment potential will be limited and the results of the AB Test designed to assess uplift associated with presence of the ID5 ID will be biased.

Partners must provide ID5 with **automated weekly reports** and a **monthly report** no later than 7 days after month end covering all channels competing for impressions across every domain where ID5 Bid Enrichment Services are active. These reports must include *all* competing channels, regardless of whether ID5 Bid Enrichment Services are applied to a given channel. This report must detail the results **broken down by domain**.

Example: *If ID5 Bid Enrichment Services are enabled on publisher.com and the competing channels include Prebid, Amazon TAM, OpenPath, and others, then every channel must be represented in the report.*

How to provide ID5 with Reporting?

Google Ad Manager Reporting

For client side integrations with ID5 (ID5 JS API and Prebid.js), the `params.gamTargetingPrefix` parameter must be configured in requests with a prefix e.g. `id5` . When a non-empty `params.gamTargetingPrefix` is configured, the module sets the following GAM targeting keys:

Key	Description
`{prefix}_id`	Set to `"y"` if a valid `id5id` is available. The key is **not set** if no ID is present.
`{prefix}_ab`	Set when A/B testing is enabled: `"n"` – Normal group (ID5 ID returned) `"c"` – Control group (no ID5 ID returned)
`{prefix}_enrich`	Set when **bid enrichment** is enabled: `"y"` – Enriched IDs returned `"s"` – Enrichment attempted, no enriched IDs found `"c"` – Control group (no enrichment was performed, used for uplift measurement)

It is then possible to create reports in GAM that segment by the `{prefix}_enrich` key-value pairs. Follow the Google Ad Manager Instructions to add these key values (`id5_enrich=y` , `id5 _enrich=s` , `id5_enrich=c)` to your reporting. Schedule a:

- A daily automated report with impressions, paid impressions and revenue by key value for yesterday

- A monthly automated report with impressions, paid impressions and revenue by key value for last month

Custom Uplift Reporting

Client Side ID5 Integrations

In cases when it's not feasible to use `gamTargetingPrefix` described above, ID5 can expose targeting tags through a global window.id5tags object. If `exposeTargeting` is set to true and an ID5 instance has initialized, the ID5 will expose tags through a command queue pattern. The `tags` object is in key-value pair format like this:

```
{
  "tagKey": "tagValue"
}
```

The meaning of specific key-value pairs is following:

Key	Description
`id`	Set to `"y"` if a valid `id5id` is available. The key is **not set** if no ID is present.
`ab`	Set when A/B testing is enabled: `"n"` – Normal group (ID5 ID returned) `"c"` – Control group (no ID5 ID returned)
`enrich`	Set when **bid enrichment** is enabled: `"y"` – Enriched IDs returned `"s"` – Enrichment attempted, no enriched IDs found `"c"` – Control group (no enrichment was performed, used for uplift measurement)

Partners should then include code on their page to queue callback functions to access those tags:

```
window.id5tags = window.id5tags || { cmd: [] };
window.id5tags.cmd.push((tags) => {
   // Use for custom analytics, targeting, etc.
   // eg pass to custom tagging system with something like:
   for (const tag in tags) {
      customPartnerTagging.set(customPartnerPrefix + '_' + tag, tags[tag]);
   }
});
```

Server Side ID5 Integrations

When using Mobile In App S2S API or CTV API the information about enrichment is returned in the "enrichment" object in the response:

Enrichment Object

Name	Type	Description	Example
enrichment_selected	boolean	False if request was selected to be in enrichment control group, true if it was selected to be enriched (normal group)	false
enriched	boolean	true if request was selected to be enriched AND at least one enriched ID was found and returned for given request. False otherwise	false

Partners should use this information in their reporting systems to properly tag the request

Uplift Calculations

1. **Calculate rCPM for each Group**
    a. **Control Group Revenue x1000/ Control Group Impressions (`{prefix}_enrich=c`)**
    b. **Enriched Group Revenue x 1000 / Enriched Group Impressions (`{prefix}_enrich=y` and `{prefix}_enrich=s`)**
2. **Determine the difference in rCPM**
    a. **Enriched Group rCPM - Control Group rCPM = Delta rCPM**
3. **Calculate Revenue Uplift**
    a. **Delta rCPM x (Enriched Group Impressions / 1000)**

Example

- Enriched Group Revenue = $10,000
- Enriched Group Impressions =9,000,000
- Control Group Revenue = $900
- Control Group Impressions=1,000,000

Calculation:

($10,000 x1000)/9,000,000=**$1.11 = Enriched Group rCPM**

($900 x 1000)/1,000,000 = **$0.90 = Control Group rCPM**

$1.11- $0.90 = **$0.21= rCPM Delta**

$0.21 x (9,000,000/1000) = **$1,890 Revenue Uplift**

6. Inform and Collaborate with your Partners

Inform your SSP partners that you will be using the ID5 graph to pass matched SSP and DSP cookie IDs in the `user.ext.eids` or `user.eids` field (according to SSP/DSP specifications), along with provenance metadata. Some SSPs may require you to add your inserter value and/or `matcher=id5-sync.com` to an allow-list. You can find more details on how ID5 handles ID provenance here.

If you are augmenting Prebid.js with Prebid Server, make sure to align with your SSP and demand partners on how they prefer to receive matched cookie IDs when ID provenance is not yet supported.

Enrich for Publishers (based on ID5 Graph)

07/27/2026 12:18 pm EDT

Overview

ID5's Graph empowers publishers (and their tech partners) to unlock greater value by connecting users across both cookie-based and cookieless environments. By combining the scale of the ID5 ID footprint with the intelligence of the ID5 Identity Graph, Enrich significantly boosts partner match rates, strengthens user addressability, and maximizes monetization opportunities.

With Enrich, publishers can:

- Increase match rates with partners by bridging gaps in cookie syncs.
- Translate cookieless IDs into actionable identifiers, enabling seamless targeting and activation.
- Enrich impressions and audience segments with reliable identity signals that drive performance in programmatic campaigns.

In an ecosystem where third-party cookies are fading, ID5 Enrich ensures your platform continues to deliver scale, precision, and revenue growth by making every impression addressable.

How does it work?

Publishers must first deploy the ID5 ID on their digital properties, creating the foundation for identity resolution. Once in place, ID5 delivers a daily Graph that maps ID5 IDs, capturing both past visitors and predicted future ones, to platform cookie IDs across cookie and cookieless environments. Publishers (or their technology partners) can then retrieve and decrypt ID5 IDs from the page or bid stream and then use the ID5 Graph to look up matching platform cookie IDs and enrich bid requests with accurate identity mappings and accompanying provenance information. ID5 can provide the ID5 Graph to an enablement partner such as Optable, who can manage the ID5 ID decryption, graph look up, and real-time bid enrichment on a publisher's behalf.

When should I choose to use ID5's Real Time Bid Enrichment Service vs the ID5 Graph for publishers?

Publishers should choose **ID5's Real-Time Bid Enrichment Service** if they want a simple, plug-and-play way to extend their existing ID5 integration and automatically enrich Prebid requests with cookie IDs for their platform partners. On the other hand, the **ID5 Graph for Publishers** is designed for publishers or their tech vendors that prefer more flexibility, allowing them to combine the graph with other data sources and manage real-time ID5 ID decryption and bid enrichment themselves, or in partnership with providers like Optable.

Implementation Pathway

1. Register with ID5

Sign up with ID5 and and request your ID5 Partner Number.

2. Integrate the ID5 ID on your digital properties

Deploy the ID5 ID across your user base using the implementation methods outlined here.

If you are integrating via Prebid, the following configurations are recommended:

Name	Value	Description
externalModuleUrl	`https://cdn.id5-sync.com/api/1.0/id5PrebidModule.js`	Ensures you are benefiting from the latest version of our code and your bid enrichment configuration
canCookieSync	`true`	Maximises match rates with SSPs and DSPs through cookie syncing with user consented vendors.

3. Receive and Consume the ID5 Graph

Each day, ID5 generates and delivers a customised graph based on users that have visited your digital property in the past and whom we predict might visit your digital properties in the future. This graph maps ID5 IDs to the platform cookies IDs for your configured match partners. The output supports the following configurations:

- **Country**
- **Cookie Match Partners -** Ask ID5 for an updated list
- **ID5 Matchmethod -** `id5_graph_connection` , `cookie_sync` , `ip_match`
- **Lookback window for IP match**
- **Format - csv or parquet**

Delivery Frequency

ID5 will deliver the matching file to the customer's S3 bucket on a daily or weekly basis. The matching happening on a given day will read and write using the date of the previous day / week. For example, on a daily match performed on the 5th October 2024, ID5 will read the data of `2024-10-04`

Matching File Content

Field Name	Description
`id5_id`	ID5 universal identifier
`partner_id`	ID of the matching partner
`partner_uid`	Cookie ID from matching partner
`country`	Country code for where match was observed
`last_match_timestamp`	Timestamp when the match was last confirmed: - For `id5_graph_connection` , it's the CDG generation date - For `cookie_sync` , it's the sync - For `ip_match` , it's the time when ID5 ID was last connected to the cookie ID via IP (the date bridging happens)

Field Name	Description
last_seen_timestamp	Timestamp when the ID5 ID was last observed on the ID5 footprint
id5_matchmethod	Enum: id5_graph_connection , cookie_sync , ip_match
ortb_matchmethod	Enum (0–5 or 500+), per OpenRTB match method (see spec)

Output File Formats and Paths

ID5 supports two file formats when delivering the ID5 Graph file: CSV and Parquet. The output will be stored on a partitioned basis using the format:

`<configurable prefix>/DATE=<ISO date>/<some random filename>.[csv.gz|snappy.parquet]`

Examples:

- `/id5/output/DATE=2024-04-09/part-00000-tid-5623552081912624944-0b8fafa8-51a6-4f35-978c-2cd65b4957f3-85138-1-c000.snappy.parquet`
- `/id5/csv_output/DATE=2024-04-07/part-00000-tid-897652208191266533-ca6f3876-557d-4de3-bf76-32330850a2d8-78652-1-c000.csv.gz`

- Parquet
  - A binary, data efficient file format which will contain all of the supplied data fields in a flat file structure. The files are compressed using the `snappy` algorithm.
- CSV
  - The CSV file contains all of the supplied fields in a flat file structure. The separator is `,` (comma) and the quotes `"` are added only if required. The output files are compressed using the `gzip` algorithm.

    #### CSV Example

    In this example, we're using the CSV format without timestamps. First row in the example is the headers, second one an ip / id5id which additionally has an ID5 partner cookie match and the third one has no partner cookie match.

    ```
    ip,id5id,partner_id,partner_uid
    67.244.50.69,ID5-ZHMOx88UySFiiNN0lpFvLeZ9yAyhi_P0xfGc9fw7yQ,264,436019
    35.137.25.212,ID5-a9ddxNN39oI_silnqpcuG9bYnzrOcmfCt804_cNL0w,,
    ```

Best Practises

ID Relationships

Multiple ID5 IDs may map to a single Cookie ID for a given Match Partner. Conversely, the same ID5 ID may map to different Cookie IDs for a given partner as a result of different match methods. In these cases, ID5 selects the most recently observed match and will include it in the file.

File Consumption and Data Retention

- Upon receipt of each daily file, consumers should:
  - Update existing matches.
  - Insert new matches.

- If a cookie ID or ID5 ID has not been observed within the last *n* days, its associated matches should be purged from your dataset.

  ID5 recommends *n = 30 days*, and no more than *n = 60 days*.

5. Decrypt the ID5 ID and Leverage the ID5 Graph

The encrypted ID5 ID can be obtained in one of two ways:

- Via the **ID5 JS API** on a digital property.
- From the bid stream, where it may appear in the `EID` array.

You should implement ID5 ID decryption capability (see instructions). Upon receipt of an encrypted ID5 ID, decrypt it before proceeding. You may then use the **ID5 Graph** to resolve the decrypted ID5 ID into the corresponding cookie IDs for your chosen match partners . These resolved identifiers can be added to bid requests in accordance to user consent choices and the OpenRTB specification.

Consent & Compliance

- When using the graph, only leverage `ID5 ID → Platform Cookie ID` mappings where the user has provided valid consent to both ID5 and the Platform, in jurisdictions where consent is required.
- Ensure full compliance with the OpenRTB specification and applicable ID provenance guidelines.

Enrich for SSPs (Offline graph)

05/21/2026 4:09 am EDT

Overview

ID5's Graph can empower ad tech platforms to unlock greater value by connecting users across both cookie-based and cookieless environments. By combining the scale of the ID5 ID footprint with the intelligence of the ID5 Identity Graph, Enrich significantly boosts partner match rates, strengthens user addressability, and maximizes monetization opportunities.

With Enrich, SSPs can:

- Increase match rates with partners by bridging gaps in cookie syncs.
- Translate cookieless IDs into actionable identifiers, enabling seamless targeting and activation.
- Enrich impressions and audience segments with reliable identity signals that drive performance in programmatic campaigns.

In an ecosystem where user identification is fragmented and contingent on the environment, ID5 Enrich ensures your platform continues to deliver scale, precision, and revenue growth by making every impression addressable.

Example Use Cases

1. SSPs Enhancing DSP Connectivity

Supply-Side Platforms (SSPs) can use Enrich to:

- Increase cookie sync match rates with their DSP partners.
- Use the ID5 Graph to connect cookieless ID5 IDs with their own cookie-based IDs.
- Enrich cookieless bid requests with the DSPs' cookie IDs, making those impressions addressable for activation.

2. Activating Audience Targeting with DMPs and CDPs

SSPs can leverage Enrich to:

- Strengthen cookie syncs with Data Management Platforms (DMPs) and Customer Data Platforms (CDPs).
- Use the ID5 Graph to connect cookieless ID5 IDs with their own cookie-based IDs.
- Determine whether a user in a cookieless environment belongs to a specific audience segment.
- Enable audience-targeted campaigns in cookieless environments, ensuring advertisers can continue to reach the right users even when cookies are not available.

How does it work?

Platforms using the Enrich service (e.g., SSPs) must first set up a cookie sync with ID5 to maximize match rates across IDs. ID5 then provides a daily Graph that maps ID5 IDs to platform cookie IDs in both cookie and cookieless environments. Platforms can then retrieve and decrypt ID5 IDs from the page or bid stream, then use the Graph with their enhanced match tables to enrich bid requests with accurate ID mappings and provenance information.

Implementation Pathway

Implementation will consist of:

1. Cookie Synchronisation with ID5 - By initiating and receiving a cookie sync to ID5, your cookie sync footprint and match rates will be maximised.

2. Real-time updates – ID5 will provide real-time updates of the most recent matches between your cookie ID and the cookie IDs of your chosen match partners. You can use this to add and update entries in your cookie match tables.

3. Daily file delivery – ID5 will send a single daily file, showing the most recent matches between the ID5 ID and your cookie ID, along with ID provenance metadata. You should store this data and update it daily by adding new matches and refreshing existing connections between the ID5 ID and your cookie ID. Probabilistic connections older than 30 days should be purged from the data set, although you may prefer a shorter retention window.

4. ID5 ID Decryption - Upon receipt of a bid request containing an encrypted ID5 ID, you should decrypt it in order to access the stable version of the ID5 ID.

5. Real-time look-up and bid enrichment - After decrypting the ID5 ID:

1. Find your matching cookie ID.
2. Use this cookie ID together with your enhanced match tables to identify the corresponding DSP/DMP cookie IDs.

This allows you to:

- Check which audience campaigns or deal IDs are eligible to serve.
- Enrich bid requests sent to DSPs with ID provenance metadata.

Step-by-Step Instructions

1. Register with ID5

Sign up with ID5 and request your ID5 Partner Number.

2. Configure Cookie Synchronization with ID5

To maximise your cookie coverage across the ID5 footprint and increase your match rates with ID5 and

cookie sync partners, you should:

- Initiate a cookie sync to ID5
- Receive a cookie sync from ID5

3. Enhance your Cookie Match Tables with ID5's Partner Graph File Transfer

Tell ID5 the platforms for which you require improved match rates. ID5 will provision a Partner Graph File Transfer, delivering files to your designated AWS S3 bucket at defined intervals. Each file contains newly generated and updated Match Partner UIDs derived from the cookie sync process. These UIDs should be ingested to update and enrich your existing cookie match tables. Refer to the full integration documentation here.

4. Receive the ID5 Graph

Each day, ID5 generates and delivers a dedicated graph to your environment. This graph maps ID5 IDs to your proprietary cookie IDs.

Delivery Frequency for the ID5 Graph

ID5 will deliver the Graph to the customer's AWS S3 bucket (or S3-compatible solution) on a daily basis. The matching happening on a given day will read and write using the date of the previous day. For example, on a daily match performed on the 5th October 2024, ID5 will read the data of `2024-10-04`

Matching File Content

Field Name	Description
`id5_id`	ID5 universal identifier
`partner_id`	ID of the matching partner
`partner_uid`	Cookie ID from matching partner
`country`	Country code for where match was observed
`last_match_timestamp`	Timestamp when the match was last confirmed: - For `id5_graph_connection`, it's the CDG generation date - For `cookie_sync`, it's the sync - For `ip_match`, it's the time when ID5 ID was last connected to the cookie ID via IP (the date bridging happens)
`last_seen_timestamp`	Timestamp when the ID5 ID was last observed on the ID5 footprint
`id5_matchmethod`	Enum: `id5_graph_connection`, `cookie_sync`, `ip_match`
`ortb_matchmethod`	Enum (0–5 or 500+), per OpenRTB match method (see spec)

Output File Formats and Paths

ID5 supports parquet file format when delivering the ID5 Graph file. The output will be stored on a partitioned basis using the format:

`s3://bucket_name/parent_folder_name/date=yyyy-mm-dd/region=xyz/partner=SSP_name/file.gz.parquet`

Best Practises

File Consumption and Data Retention

- Upon receipt of each daily file, consumers should:
    - Update existing matches.
    - Insert new matches.
- If a cookie ID or ID5 ID has not been observed within the last *n* days, its associated matches should be purged from your dataset.

  ID5 recommends *n = 30 days*, and no more than *n = 60 days*.

5. Decrypt the ID5 ID and Leverage the ID5 Graph

Retrieval
The encrypted ID5 ID can be obtained in one of two ways:

- Via the **ID5 JS API** on a digital property.
- From the bid stream, where it may appear in the `EID` array.

Decryption
You should implement ID5 ID decryption capability (see instructions). Upon receipt of an encrypted ID5 ID, decrypt it before proceeding.

Graph Lookup

- Use the **ID5 Graph** to resolve the decrypted ID5 ID into the corresponding partner cookie ID.
- Leverage your proprietary cookie match tables (enhanced by the ID5 Partner Graph) to further map this partner cookie ID to the respective DSP and other platform cookie IDs.

Activation
These resolved identifiers can be used to:

- Populate user IDs passed downstream to support addressability.
- Power eligible audience campaigns.
- Enable private marketplace deals where the user can be targeted.

Consent & Compliance

- When using the graph, only leverage `ID5 ID → Platform Cookie ID` mappings where the user has provided valid consent to both ID5 and the Platform, in jurisdictions where consent is required.
- Ensure full compliance with the OpenRTB specification and applicable ID provenance guidelines.

ID5 OpenRTB ID Provenance Support and Best Practices

06/19/2026 12:09 pm EDT

Overview

Publishers and ad tech partners can choose to use the ID5 Graph to enrich their programmatic transactions with additional data. The Graph enables both probabilistic and deterministic connections between ID5 IDs and Platform Cookie IDs, which can be inserted into bid requests in real time to boost addressability. When using the ID5 Graph for bid enrichment, ID5 strongly recommends full compliance with the IAB Tech Lab's ID Provenance Fields and Best Practices. To support this use case, ID5 provides a dedicated version of the Graph for this use case that includes the `matchmethod(mm)` field, indicating how each ID5–to–Platform Cookie ID connection was established.

This page provides practical guidance for publishers enriching bid requests with the ID5 Graph, either directly or through a partner such as Optable, as well as for ad tech partners receiving enriched requests in which ID5 (id5-sync.com) is declared as the matcher.

Publisher Guidance

Publishers collaborating with ID5 or with technology providers that use ID5's Graph to facilitate bid enrichment should follow the OpenRTB ID provenance guidance to ensure compliance and transparency. Following these steps ensures that your bid enrichment setup is recognized, compliant, and properly processed by your SSP and DSP partners.

Recommended Steps

1. Set Inserter and Matcher Values

a. Inserter: Declare your own domain as the inserter. We recommend using the same domain as the ownerdomain in your ads.txt file.

b. Matcher: Declare id5-sync.com as the matcher.

2. Notify Your SSPs

Inform your supply-side platforms (SSPs) that you are enriching bid requests in this way. Some SSPs require whitelisting for both the inserter and matcher values before accepting or using the data.

EID Array Example of Publisher Bid Enrichment with ID5 ID and Platform Cookie IDs

```
{
 "eids": [
  {
   "source": "id5-sync.com",
   "uids": [
    {
     "id": "some-random-id-value",
     "atype": 1,
     "ext": {
      "linkType": 2,
      "abTestingControlGroup": false
     }
    }
   ]
  },
  {
   "source": "ssp-domain.com",
   "matcher": "id5-sync.com",
   "mm": 5,
   "inserter": "publisher-domain.com",
   "uids": [
    {
     "id": "some-random-id-value",
     "atype": 1
    }
   ]
  }
 ]
}
```

SSP Guidance and Expected Behaviour

Build Support for the ID provenance fields. In practice, this means:

1. When you receive a Cookie ID with provenance in the EID array, use your match tables to look up the corresponding DSP cookie IDs and include them downstream in the `user.ext.eids` or `user.eids` field, following the format shown below. We recommend passing only the DSP cookie IDs for vendors to whom the user has given consent, in accordance with the user's jurisdiction and applicable privacy regulations.

```
{
  "source": "<dsp-cookie-domain>",
  "inserter": "ssp-domain.com",
  "matcher": "id5-sync.com",
  "mm": 5,
  "uids": [
    {
      "id": "<dsp-cookie-id>",
      "atype": 1
    }
  ]
}
```

2. When no corresponding DSP cookie ID is found in your match tables, forward any DSP cookie IDs received along with their provenance in the `user.ext.eids` or `user.eids` field (according to DSP specifications).

3. Only include enriched cookie IDs in the `buyeruid` field if explicitly instructed by your DSP and if permitted under your contract.

4. Instruct publishers, or the partners enriching bids on their behalf, to provide the expected `source` value—for example, `ssp-domain.com`.

5. If you enforce allow-listing for bid enrichment at either the matcher or inserter level, add `id5-sync.com` to your matcher allow-list and request the latest list of inserter values from your ID5 representative for allow-listing.

DSP Guidance and Expected Behaviour

DSP Guidance for Handling Bid Requests with ID5 Connections

When processing bid requests that may not contain your DSP cookie ID in the `buyeruid` field, we recommend the following approach:

- **Primary identifier (ID5 ID):**
    - If present, decrypt and use the ID5 ID as the user identifier.
    - This ensures you can transact even when your DSP cookie ID is not included directly.
- **Fallback identifier (cookie ID with provenance):**
    - If using the ID5 ID is not supported, rely on your DSP cookie ID provided with provenance in the `user.ext.eids` or `user.eids` field.
    - Use this cookie ID as the user identifier for campaign targeting.
- **Bidding decision logic:**
    - Once the identifier is established, evaluate eligible advertiser campaigns and decide on bid values accordingly.
    - You may also incorporate the `matcher` and `matchmethod(mm)` values into your bidding algorithms to optimise decision-making.
- **Request your SSPs to conduct `buyeruid` Substitution**
    - If you do not support the ID5 ID or ID provenance fields, instruct your SSPs to pass you bridged cookie IDs in the buyeruid field where 'id5-sync.com' is the source.

By following this sequence, DSPs can ensure accurate user recognition, support transactions in both cookieless and cookie-supported environments, and take full advantage of the high-quality connections enabled by the ID5 graph.

Enriching Bid Requests with the ID5 ID

05/21/2026 4:12 am EDT

This guide explains how to use EID (Extended Identifier) information from an ID5 endpoint response to enrich outgoing OpenRTB bid requests for both OpenRTB 2.6 and 2.5.

Identify and Extract EID Data from ID5 Endpoint Response

The necessary EID data in the ID5 response is extracted from the `ids` object.

Example ID5 Response:

```
{
  "signature": "ID5_AoPiwhPFUccYOQLCwfbQTEbMTk85rQD10hAZYixkNlZeozXos46w4uqpJqq",
  "created_at": "2025-11-14T16:12:20.968Z",
  "original_uid": "ID5*KDdqfh7p46U_Eb_xD8-9lWYkX2xsHEQqpJP2YNzKqC7__",
  "universal_uid": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
  "privacy": {
    "jurisdiction": "gdpr",
    "id5_consent": true
  },
  "ids": {
    "id5id": {
      "eid": {
        "source": "id5-sync.com",
        "uids": [
          {
            "id": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
            "atype": 1,
            "ext": {
              "linkType": 2,
              "pba": "Fhqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWijP20OAngDPxw=="
            }
          }
        ]
      }
    },
    "euid": {
      "eid": {
        "source": "euid.eu",
        "uids": [
          {
            "id": "E4AAAAezwMkyH6T4wJwiMRLR7C7I77M2VyI8q_pQSzPLdIsfffI86arrL3FKC4T",
            "atype": 3,
            "ext": {
              "provider": "id5-sync.com"
            }
          }
        ]
      }
    },
    // EID example when using ID5's Real-Time Bid Enrichment Service
    "id5-ttd": {
      "eid": {
        "source": "adserver.org",
        "uids": [
          {
            "id": "09d89c84-2e59-4c80-9c80-09db5f901234",
            "atype": 1,
            "ext": {
              "provider": "id5-sync.com",
              "rtipartner": "TDID"
            }
          }
        ],
        "mm": 2,
        "matcher": "id5-sync.com",
        "inserter": "publisher-domain.com"
      }
    }
  }
}
```


You should extract the entire **EID objects** for **ID5 ID** and any other IDs provided (e.g., **EUID**).

- **ID5 ID EID:** Is found under `response.ids.id5id.eid`
- **EUID EID:** Is found under `response.ids.euid.eid`
- **TTD EID:** Is found under `response.ids.id5-ttd.eid`
  (Example of a DSP ID added when using the Real-Time Bid Enrichment Service)

Example EID Data Structure:

```
{
 "source": "id5-sync.com",
 "uids": [
  {
   "id": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
   "atype": 1,
   "ext": {
    "linkType": 2,
    "pba": "Fhqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWijP20OAngDPxw=="
   }
  }
 ]
}
```

> ID5 provides extra data about the user ID in the `ext` object. For more information, see our metadata documentation.
>
> For EIDs added via the Real-Time Bid Enrichment Service, we also include IAB ID Provenance fields. For more details, see our ID5 OpenRTB ID Provenance Support and Best Practices.

Enriching an OpenRTB 2.6 Request

In **OpenRTB 2.6,** the `eids` array is a **direct field** within the `user` object.

To enrich the request:

1. Collect all relevant EID objects (e.g., the ID5 ID EID, EUID EID or others).
2. Insert these objects as elements into the `eids` array, placed under the `user` object.

OpenRTB 2.6 Request Snippet (user.eids)

```
{
  "user": {
    "id": "SSP_UID",
    "buyeruid": "DSP_UID",
    "eids": [
      {
        "source": "id5-sync.com",
        "uids": [
          {
            "id": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
            "atype": 1,
            "ext": {
              "linkType": 2,
              "pba": "Fhqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWijP20OAngDPxw=="
            }
          }
        ]
      },
      {
        "source": "euid.eu",
        "uids": [
          {
            "id": "E4AAAAezwMkyH6T4wJwiMRLR7C7I77M2VyI8q_pQSzPLdIsfffl86arrL3FKC4T",
            "atype": 3,
            "ext": {
              "provider": "id5-sync.com"
            }
          }
        ]
      },
      {
        "source": "adserver.org",
        "uids": [
          {
            "id": "09d89c84-2e59-4c80-9c80-09db5f901234",
            "atype": 1,
            "ext": {
              "provider": "id5-sync.com",
              "rtipartner": "TDID"
            }
          }
        ],
        "mm": 2,
        "matcher": "id5-sync.com",
        "inserter": "publisher-domain.com"
      }
// *** INSERT OTHER EID OBJECTS HERE ***
    ]
  }
}
```

Enriching an OpenRTB 2.5 Request

In **OpenRTB 2.5**, the `eids` array is an **extended field** within the `user` object (`user.ext`).

To enrich the request:

1. Collect all relevant EID objects.
2. Insert these objects as elements into the `eids` array, placed within the `ext` object of the `user`

object.

OpenRTB 2.5 Request Snippet (user.ext.eids)

```
{
  "user": {
    "id": "SSP_UID",
    "buyeruid": "DSP_UID",
    "ext": {
      "eids": [
        {
          "source": "id5-sync.com",
          "uids": [
            {
              "id": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
              "atype": 1,
              "ext": {
                "linkType": 2,
                "pba": "Fhqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWijP20OAngDPxw=="
              }
            }
          ]
        },
        {
          "source": "euid.eu",
          "uids": [
            {
              "id": "E4AAAAezwMkyH6T4wJwiMRLR7C7I77M2VyI8q_pQSzPLdIsfffI86arrL3FKC4T",
              "atype": 3,
              "ext": {
                "provider": "id5-sync.com"
              }
            }
          ]
        },
        {
          "source": "adserver.org",
          "uids": [
            {
              "id": "09d89c84-2e59-4c80-9c80-09db5f901234",
              "atype": 1,
              "ext": {
                "provider": "id5-sync.com",
                "rtipartner": "TDID"
              }
            }
          ],
          "mm": 2,
          "matcher": "id5-sync.com",
          "inserter": "publisher-domain.com"
        }
 // *** INSERT OTHER EID OBJECTS HERE ***
      ]
    }
  }
}
```

ATTENTION

It is recommended to verify the SSP/DSP integration specifications to see if they support

OpenRTB 2.6 or still require OpenRTB 2.5 formatting. Placing the EID structure in the correct location is crucial for proper processing by the demand partner.

Accessing EIDs with the ID5 JS API

If you are using the ID5 JS API, the library provides a convenient method to retrieve all available EIDs, including the ID5 ID, already formatted as an array of **OpenRTB EID objects**.

The method: `getUserIdsAsEids()`, retrieves the necessary **EID objects** that can be directly added to an OpenRTB bid request's `eids` array.

Output Array Example

The array returned by `getUserIdsAsEids()` can contain multiple EID objects depending on the integration. For instance, an integration that has enabled the **Guarded Publisher ID (GPID)** will return both the ID5's EID and the GPID EID.

This output array perfectly matches the required structure for the OpenRTB `eids` field.

```
[
  // ID5 EID
  {
    "source": "id5-sync.com",
    "uids": [
      {
        "id": "ID5*1Ls3HtzGAldqduMJjmL2CcyuGOw9JArVgXc_LS34MC7__",
        "atype": 1,
        "ext": {
          "linkType": 2,
          "pba": "Fhqkw5e7uf4ei8WEIXviXDGQnqqkVeEgVWijP20OAngDPxw=="
        }
      }
    ]
  },
  // EUID EID
  {
    "source": "euid.eu",
    "uids": [
      {
        "id": "E4AAAAezwMkyH6T4wJwiMRLR7C7I77M2VyI8q_pQSzPLdIsfffI86arrL3FKC4T",
        "atype": 3,
        "ext": {
          "provider": "id5-sync.com"
        }
      }
    ]
  },
  // EID example when using Guarded Publisher ID (GPID)
  {
    "source": "gpid.id5-sync.com",
    "uids": [
      {
        "id": "m4nzI8tQ3iZ-XuHRu3Bd47Y3AUDNGykf58sOwXehqnIfvK35LIb1DlHXJ-",
        "atype": 1
      }
    ]
  },
  // EID example when using ID5's Real-Time Bid Enrichment Service
  {
    "source": "adserver.org",
    "uids": [
      {
        "id": "09d89c84-2e59-4c80-9c80-09db5f901234",
        "atype": 1,
        "ext": {
          "provider": "id5-sync.com",
          "rtipartner": "TDID"
        }
      }
    ],
    "mm": 2,
    "matcher": "id5-sync.com",
    "inserter": "publisher-domain.com"
  }
]
```

For additional information about ID5's JS API, see our public GitHub documentation.