OpenRTB Protocol Specification

Table of Contents

Revision History

Version	Date	Updater	Comment
1.0.0	12/09/2010	B. Simmons	Publishing v1.0 of the document.
1.0.1	1/28/2011	C. Pate	Updating the document based upon feedback from early adopters of the spec. Changes include clarification of "Standard Ads" and the removal of the sinceThisTimestamp behavior from the advertiser blocklist requests and responses.
1.1	2/15/2011	C. Pate	No spec changes. Moving spec version to major.minor version number to enable reference implementation to have major.minor.patch version number. The major.minor version numbers will be kept in lockstep with between the spec and reference implementation.

1. Overview & Goals

1.1. Mission

The mission of the OpenRTB project is to spur greater growth in the real-time bidding (RTB) marketplace by providing open industry standards for communication between buyers of advertising and sellers of publisher inventory.

In particular, as a first goal, we aim to reduce the cost and increase the consistency of managing the synchronization of advertiser requirements and publisher requirements with all players in the ecosystem.

Advertiser and Publisher requirements, which are often called "blocklists", are supplied by advertisers to DSPs (demand side platforms) and publishers to SSPs (sell side platforms). These blocklists are often supplied in different ways to different providers, which then need to be translated by hand by DSPs and SSPs parties before a campaign begins. By eliminating this labor-intensive bottleneck in RTB industry, we will spur greater growth for both DSPs and SSPs by lowering operational costs.

This project was launched by a six-company pilot team made up of 3 DSPs and 3 SSPs. We enthusiastically welcome more participants to the project. The initial pilot team members are:

Demand Side Platforms (DSPs)	Sell Side Platforms (SSPs)
DataXu	AdMeld
MediaMath	Pubmatic
Turn	The Rubicon Project

2. Specific Benefits for SSPs and DSPs

The first goal of the OpenRTB working group is to standardize a protocol to remove the labor intensive and error prone process in the RTB ecosystem: Managing publisher and advertiser blocklists. The objective of OpenRTB is to minimize these impediments, so that the RTB ecosystem, made up of DSPs (Demand-Side Platforms) and SSPs (Sell-Side Platforms), can grow more quickly and ensure a higher quality of service to advertisers and publishers.

Specifically, we aim to address the needs of publishers to block certain advertisers or creative types, and the needs of advertisers to block certain publishers or content

types in the RTB environment. This proposal includes an offline (batch) protocol as well as the definition for a standard set of features in a real-time bid request.

2.1. Benefits for SSPs

(1) **Higher CPMs**: If SSPs provide blocklists to their DSP partners in an offline, batch manner, the DSP can incorporate filters in their bidding functionality to not bid on inventory they cannot win. This will result in a higher percentage of valid bids per impression. Statistically, this will result in fewer bids to process for bid request and higher CPMs per impression for the publisher.
(2) **Reduce Risk**: Better and more Uniform Enforcement of Publisher Restrictions: If DSPs and SSPs have a unified taxonomy for ad blocking, it will make it easier for DSPs to strictly follow publisher rules when bidding. This will reduce the risk of potential issues with publishers due to human error.
(3) **Increased Revenue** through easier Integration with DSPs: Reducing the manual work to translate blocklists from a SSP's format to a DSP's format will reduce the time to launch new campaigns. This will result in easier integration for DSPs, more competitive bidding, and higher revenue for the SSP.

2.2. Benefits for DSPs

(1) **Lower costs** through more efficient Ad Delivery: Using pre-bid-filters, DSPs can stop bidding where they can't possibly win due to publisher or creative restrictions. This will reduce DSP costs since they will be able increase their ratio of valid bids per impression.
(2) **Reduce risk** through better enforcement of advertiser and publisher restrictions: If DSPs and SSPs have a unified taxonomy for ad blocking, it is easier for DSPs to strictly follow advertiser and publisher rules when bidding. This will reduce potential issues with advertisers and SSPs due to human error.
(3) **Greater reach** through easier integration with SSPs: Easier on boarding of new campaigns and creatives reduces the tedious manual work required to translate blocklists from an SSP's format to a DSP's format. Using OpenRTB will result in lower costs to onboard a new SSP, and will result in broader reach for the DSP.

3. Needs of Publishers and Advertisers

3.1. Publisher-Side (SSP) Needs

Publishers want to control the quality of content on their site. This includes controlling the types of brands or industries that show up in advertisements on publisher pages. Reasons for blocking creatives may include channel conflict or content conflicting with the publisher's mission.

Publisher Quality Management Needs:

P1. **Block Advertiser by Landing Page URL** – for example: "block www.carbrand.com"
P2. **Block Creatives by Attributes of the Creative** – for example: "block ads that have audio, have video, are 'shaky/flashy'"
P3. **Block Advertisers by Industry** – for example, "block automotive, block credit cards."

Note that a publisher can handling blocking of industries using internal means of matching up advertiser landing pages to industries. A methodology for handling case P3 is not included in the OpenRTB spec, since it can be achieved by filtering on sell-side supplied contextualization, or buy-size supplied contextualization of publisher URLs.

3.2. Advertiser-Side (DSP) Needs

Advertisers are concerned about safety of their brand, specifically, placement of ads near content that is not appropriate for their message.

Specific Quality Management Needs of Advertisers:

A1. **Block or Specifically Allow Publishers by publisher URL** – for example: "block www.badcontent.org, allow www.safecontent.com"
A2. **Block or Specifically Allow Publishers by the Exchange's PublisherID and Site ID**– for example: "block SSP-Pub-5, SSP-Site-6"
A3. **Block or Specifically Allow Content Channels/Verticals/Categories** – for example: "block Sports, allow Financial"
A4. **Block Publishers by Non-Standard Content Flags, or Audience Ratings** – for example: "block content not suitable for children."
A5. **Block or Specifically Allow Ads to be placed on pages with user generated content (UGC)** – for example: moderated UGC is acceptable, and un-moderated is not allowed.

4. High-Level Implementation Concept: Offline Batch & Real-Time

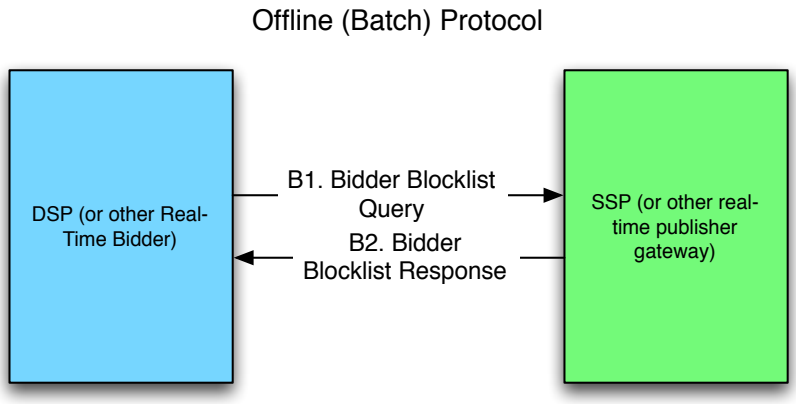
4.1. Incremental Approach

To make a transition to OpenRTB easier for a DSP or SSP, the design of this protocol is intentionally specified in a way that it can be implemented partially or completely by either party. By design, DSPs and SSPs can roll out pieces of this protocol as needed or desired. OpenRTB provides a framework for these kinds of features to be rolled out over time in a way that will be consistent with existing solutions in the industry.

There are two parts of this protocol: (1) Offline Batch Sync, as well as (2) Real-Time Filtering. In general, offline synchronization is used for managing data that has 100s or 1000s of rows that need to be synced and coordinated. Real-Time Filtering is used for the opposite case, where there are 1000s of things, and relatively few filtering criteria. The pilot implementation group has found that due to how each of our internal systems are designed, offline synchronization may work better in some cases, and online real-time filtering may work better in others. OpenRTB provides for flexibility in how restrictions are enforced, as long as the DSPs and SSPs use at least one of the methods.

The specific cases are described in the following sections.

4.2. Offline Batch Synchronization



In the case of OpenRTB, bidders typically have lists of 100s of advertisers, each with at least one, but the possibility of multiple, names and landing pages. Each advertiser may or may not have been blocked by publishers on the SSP side. The offline batch synchronization method provides a method for a DSP to download publisher restrictions during campaign setup. The figure above shows a simple

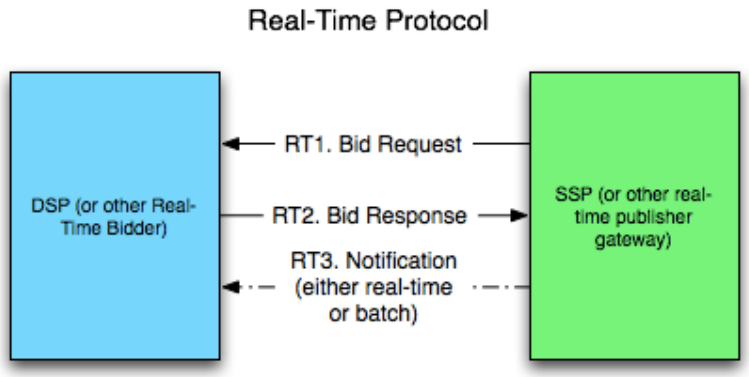
request-response interface between the DSP and SSP. If blocklists change over time due to changes in publisher preferences, a DSP can receive this information by polling the SSP's synchronization service on a periodic basis (for example, hourly).

In the figure above, "B1" is a batch request with a list of current advertisers including landing page URLs and full-text advertiser names (for publisher reference). The SSP will query its blocklist database, and return the message "B2", a response that includes a mapping of each landing page URL to a list of Blocked Publisher IDs and Site IDs.

Under the OpenRTB protocol, participating SSPs will maintain a simple, standard web service to facilitate these synchronization calls.

Note that this method does not preclude the SSP from running their own blocklist database with their own definitions and taxonomies. However, if one is already set up, the SSP will need to translate these blocklist definitions into the universal format. This allows a SSP to provide maximum flexibility and features to a publisher, while still providing a consistent interface to DSPs.

4.3. Real-Time Blocking Protocol



In the case of creative attribute and content category blocking, there is a limited list of attributes, so it may make more computational sense to process these with each real-time bid request. Creative tagging is done ahead of time by DSPs.

Given a standardized list of creative attributes that are (yes/no) questions, DSPs can tag each creative with markers indicating if a creative falls into a category. The SSP will allow publishers to pre-set what's allowed and not allowed on their site. As a part of the real-time bid request (message RT1), the SSP will include the blocked creative attribute types. Message RT1 will also include the Publisher ID and Site ID for real-time advertiser blocking.

The bidder responds in the normal fashion with a bid response (message RT2). If a bid is submitted the DSP will only send advertisers and creatives that are eligible for the specified publisher ID, site ID, and allowed creative attributes. Since data may be out of date or not yet synched using the hourly batch protocol, it's suggested that the SSP double check the contents of the response to make sure the DSP is in compliance. Message RT2 also includes a DSP unique ID identifying the creative, so that any issues can be resolved quickly.

Optionally, some SSPs may choose to provide impression-by-impression or batch reports (message RT3) noting any filtered advertisers or creatives back to the DSP. Message RT3 is not specified in this release of the OpenRTB protocol.

Optionally, SSPs may provide content channel information about each publisher URL. The recommendation is to use the standard taxonomy presented in the IAB network and exchanges guidelines (see: http://www.iab.net/media/file/NE-QA-Guidelines-Final-Release-0610.pdf). A list of these categories as of June 2010 is provided below.

5. Offline/Batch Synchronization Specification

This section contains specifications for three different operations:

- Message Verification (Section 5.1)
- Advertiser-Centric Blocklist synchronization (5.2)
- Publisher-Centric Blocklist synchronization (5.3)

5.1. Batch Synchronization Message Verification

Verification of the message content is handled inline with each request. Every request will contain an identification object. The identifier's `token` is constructed by computing the MD5 checksum of the JSON request message, less the `token` attribute, concatenated with a shared secret. The receiver of the request must remove the `token` from the request and generate an MD5 checksum with the same shared secret to confirm authenticity. The shared secret may take any form (i.e. text string or key encryption) and is communicated between the parties outside of this protocol.

This specification does not preclude methods and/or procedures that an internal technical operations team will employ to secure this solution; for example ip address filtering and SSL encryption.

Identification Object

Field	Scope	Type	Description
organization	required	string	The identifier used by the receiver to identify the requesting organization.
timestamp	required	long	The number of milliseconds since EPOC this request was made.
token	required	string	The associated MD5 token used to confirm the authenticity of and uniquely identify the request.

Example Request:

```
{
  “identification”: {
    "organization": "The DSP",
    "timestamp": 1289405763341,
    "token": "05876f434f5bf2d7b8c1e3de67135c6a"
  },
...
(rest of message)
,,,
}
```

All request responses include the identification object in addition to the status object. The status object contains the necessary error messages and status for the associated request. The token of the identification object is constructed in the same manner as described above.

Status Object

Field	Scope	Type	Description
requestToken	required	string	The identifier associated with the original request.
statusCode	required	int	An integral status code associated with the request. 0 indicates no error. 1 indicates authentication error. 2 indicates a duplicate transaction. 3 indicates other error (see statusMessage).
statusMessage	required	string	For status codes of 0, ”success” is returned; otherwise and error message is present describing the issue with validation (for example, “incorrect password”, or “user not found”)

Example Response:

```
{
  “identification”: {
    "organization": "The SSP",
    "timestamp": 1289420873341,
    "token": "5ff9c3f8511a79a3edb8b153a18002a3"
  },
  “status”: {
    "requestToken": "05876f434f5bf2d7b8c1e3de67135c6a",
    "statusCode": 0,
    "statusMessage": "success"
  },
```

```
...
(rest of message)
...
}
```

5.2. Advertiser-Centric Offline Synchronization Service

The advertiser-centric offline synchronization service allows a DSP to automatically pull and update all publisher blocklist data from the SSP and set up pre-filters in their bidding system. It is expected that a DSP will periodically poll for updates from the SSP, so that any publisher restrictions can be automatically propagated to buyers.

5.2.1. Advertiser-Centric Request Parameters

Advertiser Blocklist Synchronization Request

Field	Scope	Type	Description
identification	required	object	See "Identification Object" definition.
advertisers	required	list of object	One or more "Advertiser Object" definitions, specified in the table below.

Advertiser Object

Field	Scope	Type	Description
landingPageTLD	required	string	The advertiser's landing page url as a top-level domain (TLD). Advertiser landing pages are expected to be a generic landing page for the advertiser, not (for example) a redirect url through an ad server. Example: "carbrand.com". Should be in lower case letters. This field is used as a unique ID for the advertiser between the DSP and SSP.
name	optional	string	The advertiser's name. This field is used for informational purposes only and is not expected to be unique.

Example Request:

```
{
  "identification": { ... },
  "advertisers": [
    {
        "landingPageTLD": "acmeluxuryfurniture.com",
        "name": "Acme Luxury Furniture"
    },
    {
        "landingPageTLD": "luxurycarbrand.com",
        "name": "Luxury Car Brand"
    }
```

```
  ]
}
```

5.2.2. Advertiser-Centric Response Parameters

Advertiser Blocklist Synchronization Response

Field	Scope	Type	Description
identification	required	object	See "Identification Object" description.
status	required	object	See "Status Object" description. No additional error codes are specified as a part of this response.
advertisers	required	list of objects	One or more "Advertiser Object" definitions, specified in the table below.

Advertiser Object

Field	Scope	Type	Description
landingPageTLD	required	string	The advertiser's landing page url as a top-level domain (TLD). The value specified here should be equivalent to the one specified in the request.
blocklist	optional	list of object	One or more "Blocklist Object" definitions, specified in the table below. If left blank, there is no blocklist for this advertiser.

Blocklist Object

Field	Scope	Type	Description
publisherID	required	string	SSP's unique ID for the publisher. There are no restrictions imposed on the content of this value beyond the specified datatype (i.e. the value may be alphanumeric).
publisherName	optional	string	Human readable name of the publisher. This field is used for informational purposes only and is not expected to be unique.
siteID	required if site level blocking is used	string	If blocking is at the site level, then specify the SSP's site ID. There are no restrictions imposed on the content of this value beyond the specified datatype (i.e. the value may be alphanumeric). If this field is blank, then all sites under a publisher are blocked.
siteName	optional	string	Human readable name of the site ID, if site level blocking is used. This field is used for informational purposes only and is not expected to be unique.

Example Response:

```
{
  "identification": { ... },
  "status": { ... },
  "advertisers": [
    {
```

```
      "landingPageTLD": "acmeluxuryfurniture.com",
      "blocklist": [
        {
          "publisherID": "3422",
          "publisherName": "Joe's News"
        },
        {
          "publisherID": "2342",
          "publisherName": "Big Portal",
          "siteId": "1",
          "siteName": "Finance section"
        },
        {
          "publisherID": "23423",
          "publisherName": "Smith Blog",
          "siteID": "223",
          "siteId": "Technology Section"
        },
        {
          "publisherID": "423",
          "publisherName": "Smith Blog",
          "siteId": "23",
          "siteName": "Cars Section"
        },
        {
          "publisherID": "34223",
          "publisherName": "Jones Blog"
        }
      ]
    },
    {
      "landingPageTLD": "luxurycarbrand.com",
      "blockList": [
        {
          "publisherID": "3422",
          "publisherName": "Joe's News"
        }
      ]
    }
  ]
}
```

6. Real-Time Blocklist Spec

6.1. Real-Time Blocking of Creatives using Universal Creative Attributes

This system will allow enforcement of publisher-mandated creative attribute restrictions in real-time, same way that advertiser blocks are currently enforced.

- SSP sends to DSP list of allowed creative types on each bid request

- DSP sends to SSP list of creative types that apply to the creative on each bid

6.2. (From SSP to DSP) List of Allowed Creative Attributes

DSPs will tag creatives with the following list of universal creative attributes. SSPs will send a list indicating which attributes are blocked and which ones are allowed in real-time with each bid request. The choice of how to implement sending this list of tags is left up to the SSP; however, the attribute lists must use the standard IDs specified below.

Universal Creative Attributes and IDs:

ID	Creative Attribute
1	Audio Ad (Auto Play)
2	Audio Ad (User Initiated)
3	Expandable (Automatic)
4	Expandable (User Initiated - Click)
5	Expandable (User Initiated - Rollover)
6	in Banner Video Ad (Auto Play)
7	in Banner Video Ad (User Initiated)
8	Pop (such as Over, Under, or upon Exit)
9	Provocative or Suggestive Image Ads
10	Shaky/Flashing/Flickering Ad, Extreme Animation, or Smileys
11	Survey Ads (such as Dynamic Logic, Insight Express, Vizu, etc.)
12	Text Only Ads
13	User Interactive (such as embedded games)
14	Windows Dialog Ad or Alert Style Ad

SSPs will send a list of blocked attributes in their real-time bid request to each DSP. The exact format of this message will depend on each SSP's real-time bidding protocol. One possible encoding is an integer or hex string representing a bit field that contains or each blocked or allowed creative attribute.

In real-time, with each bid request, the DSP will eliminate and creatives that do not satisfy the creative attribute restrictions. This allows a publisher to manage very specific tag-level restrictions on their site. For example, a certain ad tag may be the one designated for in banner video ads.

A DSP may implement creative-attribute filtering in real-time, or through pre-filters using the publisher-centric synchronization service.

6.3. (From DSP to SSP) Real-Time Creative Attribute

To allow real-time publisher-side enforcement of creative attribute compliance the DSP must send with each bid response the list of creative attributes that apply. If the creative can be considered a "Standard Ad", the DSP will send no creative attributes in the response.

The SSP will then be able to do real-time enforcement of creative types for each bid, which will work the same way as it does for blocklist enforcement.

The implementation is dependent on each SSP's real-time bidding protocol. One possible encoding is an integer or hex string representing a bit field that contains all creative attributes that apply to the creative.

6.4. (From DSP to SSP) Real-Time Creative ID

The DSP will include a Creative ID in its bid response. This is included so that the SSP can quickly flag any problematic creative for review or blocking. The creative ID will be formatted as a string.

6.5. (From SSP to DSP) Real-Time Publisher ID and Site ID

To complete the pre-filtering of advertisers for each impression, SSPs should provide the Publisher ID and Site ID for impression with each real-time bid-request. These are assumed to be strings, in the same format as was provided by the SSP in the offline / batch sync.

6.6. Real-Time Blocking of Content Categories

The OpenRTB protocol group suggests the use of the IAB Networks and Exchanges Taxonomy for providing information about content categories. Details of the guidelines can be found here:http://www.iab.net/media/file/NE-QA-Guidelines-Final-Release-0610.pdf.

For Reference, tier 1 categories are listed here. IAB25 and IAB26 represent "non-standard content" and "illegal content", as listed in the networks and exchanges guidelines.

ID	IAB Name
IAB1	Arts & Entertainment
IAB2	Automotive
IAB3	Business
IAB4	Careers
IAB5	Education
IAB6	Family & Parenting
IAB7	Health & Fitness
IAB8	Food & Drink
IAB9	Hobbies & Interests
IAB10	Home & Garden
IAB11	Law Gov't & Politics
IAB12	News
IAB13	Personal Finance
IAB14	Society
IAB15	Science

IAB16	Pets
IAB17	Sports
IAB18	Style & Fashion
IAB19	Technology & Computing
IAB20	Travel
IAB21	Real Estate
IAB22	Shopping
IAB23	Religion and Spirituality
IAB24	Uncategorized
IAB25	Non Standard Content
IAB26	Illegal Content

6.6.1. Real-Time Reporting of UGC Code in Bid Request

Many advertisers require that DSPs run on inventory that does not include user-generated content. SSPs should report UGC status of the publisher, site, or placement with each bid request with the following format:

UGC Code	Description
0	No user-generated content
1	Moderated User-Generated Content
2	Unmoderated User-Generated Content

If this field is left blank, it can be assumed that UGC status for that publisher is unknown.

7. Work in Progress

This section contains a list of future topics, which will not be included in the initial release of OpenRTB

7.1. (Optional) Publisher-Centric Offline Synchronization Service

This section discusses a method of offline synchronization for various publisher preferences. Publisher preferences contain inclusion and exclusion lists. Publisher preference lists could be: URL, Creative Attribute, Creative Category RichMedia Vendors, Tracking/Pixel Vendors.

The Publisher Centric Synchronization Service provides a method for ad operation teams to answer queries for pre-filtering that may not be covered by the advertiser-centric method. For example: “tell me all of the publishers that accept creatives with

audio" or "tell me all of the publishers that accept expandables from RichMedia Vendor X".

In addition, depending on how a DSP has set up it's pre-filtering; this synchronization method may be more convenient.

7.2. (from SSP to the DSP) Real-Time Technology Flags

In addition to the creative attributes list, SSPs could send real-time flags indicating allowed technology vendors and allowed creative technologies.

7.3. Unified Handling of Private / Prioritized Seats

 *This section of is an initial version laying out basic principles. It should not be considered a complete standard.*

Private seats and prioritized bidding are two names for a method for a sending a bid request which may be able to receive multiple bid responses serving different parties, which may have different billing relationships with a publisher. Two key principles are laid out here:

1. In the real-time bid request, the SSP should include a flag specifying if seats are required. If no seat is required, then DSP can assume no special restrictions are imposed (bidding follows standard rules)
2. If seats are required for bidding, a list of potential seats should be presented by the SSP in the bid request.
3. If multiple responses are allowed, the maximum number of potential responses per bid response should be given in the bid request.
4. In the bid response, the DSP should provide the seat ID for each bid.