RTB Project

OpenRTB API Specification Version 2.3.1

November 2014
Revised June 2015

Introduction

The RTB Project, formerly known as the OpenRTB Consortium, assembled in November 2010 to develop a new API specification for companies interested in an open protocol for the automated trading of digital media across a broader range of platforms, devices, and advertising solutions. This document is the culmination of those efforts and can be found at: www.iab.net

About the IAB's Networks & Exchanges Committee

The IAB Networks & Exchanges Committee is comprised of senior leaders of ad networks and ad exchanges member companies. The committee is dedicated to furthering the interests of digital ecosystem in today's complex ad marketplace. Committee objectives are to foster the highest standards of professionalism and accountability in relationships with publishers, advertisers, intermediaries, and the agency community, to develop programs that enable revenue growth, and to create best practices that protect consumers and the industry.

The RTB Project is a working group within the IAB Advertising Technology Council.

IAB Contact Information

Melissa Gallo
Director of Product – Programmatic Automation and Data, IAB Tech Lab
melissa@iab.net

License

Table of Contents

Getting Started

This specification contains a detailed explanation of a real-time bidding interface. Not all objects are required, and each object may contain a number of optional parameters. To assist a first time reader of the specification, we have indicated which fields are essential to support a minimum viable real time bidding interface for various scenarios (banner, video, etc.).

A minimal viable interface should include the **required** and **recommended** parameters, but the scope for these parameters may be limited to specific scenarios. In these cases, the Description column may further qualify their **required** or **recommended** status. Optional parameters may be included to ensure maximum value is derived by the parties.

	Attribute	Type	Description
Examples of required attributes. Grouped at the tops of tables for convenience.	`id`	string; required	. . .
	`imp`	object array; required	. . .
Examples of recommended attributes. Grouped after required attributes.	`site`	object; recommended	. . .
	`app`	object; recommended	. . .
	`device`	object; recommended	. . .
	`user`	object; recommended	. . .
Examples of optional attributes, with and without defaults. Attributes are assumed optional unless explicitly qualified as required or recommended.	`test`	integer; default 0	. . .
	`at`	integer; default 2	. . .
	`tmax`	integer	. . .
	`wseat`	string array	. . .

Figure 1: Example of how Required, Recommended, and Optional attributes are presented.

IMPORTANT: Since **recommended** attributes are not required, they may not be available from all supply sources. It is suggested that all parties to OpenRTB transaction complete the integration checklist on the next page to identify which attributes the supply side supports in the bid request, and which attributes the demand side requires for ad decisioning.

Integration Checklist

☐ **[Company Name]** is a **supply source**, and these are the objects/parameters **supported** in the bid request.

☐ **[Company Name]** is a **demand source**, and these are the objects/parameters **required** for ad decisioning.

Supported Scenarios

Web Browser	App *(typically mobile)*	Other
☐ Banners ☐ Video ☐ Native	☐ Banners ☐ Video ☐ Native	☐ Please Specify:

Supported Objects/Parameters

Object	Supported?	Recommended Parameters NOT Supported	Optional Parameters Supported
Bid-Request	☒		
Impression	☒		
Banner	☐		
Video	☐		
Native	☐		
Site	☐		
App	☐		
Content	☐		
Publisher	☐		
Producer	☐		
Device	☐		
Geo	☐		
User	☐		
Data	☐		
Segment	☐		
Regulations	☐		
PMP	☐		
Direct Deals	☐		

1. Introduction

1.1 Mission / Overview

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. There are several aspects to these standards including but not limited to the actual real-time bidding protocol, information taxonomies, offline configuration synchronization, and many more.

This document specifies a standard for the Real-Time Bidding Interface that has grown out of previous OpenRTB collaboration on the "block list project" and the "OpenRTB Mobile" project. These protocol standards aim to simplify the connection between suppliers of publisher inventory (i.e., exchanges, networks working with publishers, and sell-side platforms) and competitive buyers of that inventory (i.e., bidders, demand side platforms, or networks working with advertisers).

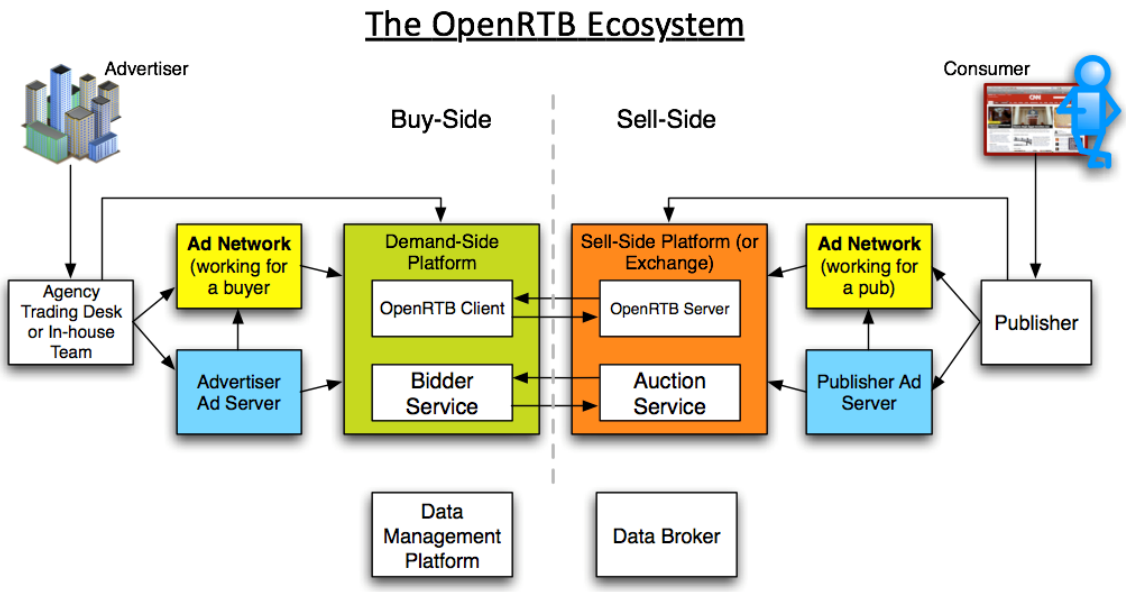

Figure 2: High-level communications between parties in the Open RTB Ecosystem.

The overall goal of OpenRTB is to create a *lingua franca* for communicating between buyers and sellers. The intent is not to regulate exactly how each business operates. As a project, we aim to make integration between parties easier, so that innovation can happen at a deeper-level at each of the businesses in the ecosystem.

1.2 History of OpenRTB

OpenRTB was launched as a pilot project between three demand-side platforms (DataXu, MediaMath, and Turn) and three sell-side platforms (Admeld, PubMatic, and The Rubicon Project) in November 2010. The first goal was to standardize communication between parties for exchanging block lists. Version 1.0 of the OpenRTB block list specification was released in December 2010.

After a positive response from the industry, Nexage approached the OpenRTB project with a proposal to create an API specification for OpenRTB focusing on the actual real-time bid request/response protocol and specifically to support mobile advertising. The mobile subcommittee was formed between companies representing the buy-side (DataXu, Fiksu, and [X+1]) and companies representing the sell-side (Nexage, Pubmatic, Smaato, and Jumptap). This project resulted in the OpenRTB Mobile 1.0 specification, which was released in February 2011.

Following the release of the mobile specification, a video subcommittee was formed with video ad exchanges (BrightRoll and Adap.tv) collaborating with DataXu and ContextWeb to incorporate support for video. The goal was to incorporate support for display, video, and mobile in one document. This effort resulted in OpenRTB 2.0, which was released as a unified standard in June 2011.

Due to very widespread adoption by the industry, OpenRTB was adopted as an IAB standard in January 2012 with the release of version 2.1. Governance over the technical content of the specification remains with the OpenRTB community and its governance rules.

1.3 Version History

OpenRTB Real-Time Bidding API

2.3 Support for Native ad units and multiple minor enhancements *(this document)*.

2.2 New enhancements for private marketplace direct deals, video, mobile, and regulatory signals.

2.1 Revisions for QAG Compliance, minor enhancements, and corrections.

2.0 Combines display, mobile, and video standards into a unified specification.

1.0 Original Release of OpenRTB Mobile.

OpenRTB Display Block List Branch

1.2 Publisher Preferences API *(proposed)*.

1.1 Minor edits to include real-time exchange of creative attributes.

1.0 Original Release of OpenRTB block list specifications.

1.4 Resources

OpenRTB on Google Groups	http://openrtb.info
OpenRTB Github Repository	http://github.com/openrtb/OpenRTB/
Development Community Mailing List	http://groups.google.com/group/openrtb-dev
User Community Mailing List	http://groups.google.com/group/openrtb-user

1.5 Terminology

The following terms are used throughout this document specifically in the context of the OpenRTB Interface and this specification.

Term	Definition
RTB	Bidding for individual impressions in real-time (i.e., while a consumer is waiting).
Exchange	A service that conducts an auction among bidders per impression.
Bidder	An entity that competes in real-time auctions to acquire impressions.
Seat	An entity that wishes to obtain impressions and uses bidders to act on their behalf.
Publisher	An entity that operates one or more sites.
Site	Ad supported content including web and applications unless otherwise specified.
Deal ID	An identifier representing a pre-arranged agreement between a Publisher and a Seat to purchase impressions under certain terms.

2. OpenRTB Basics

The following figure illustrates the OpenRTB interactions between an exchange and its bidders. Ad requests originate at publisher sites. For each inbound ad request, bid requests are broadcast to bidders, responses are evaluated under prevailing auction rules, the winner is notified, and ad markup is returned. The win notice URL and ad markup can contain any of several standard macros that enable the exchange to communicate critical data to the bidder (e.g., clearing price).

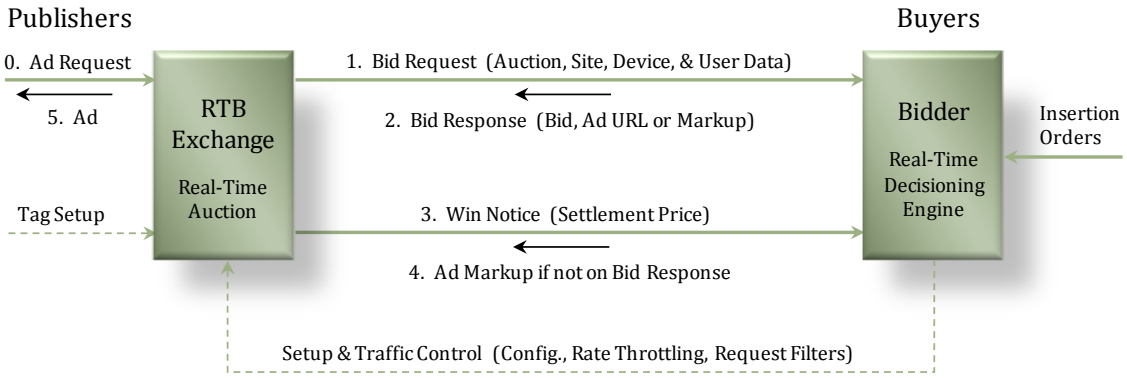

Figure 3: Real-Time request sequence.

Notice that there is no explicit provision for loss notification. This is due primarily to the significant system and bandwidth cost of doing so. However, exchanges are encouraged to supply lost bid data via an offline or separate process outside of the request/response protocol.

This specification focuses on the real-time interactions of bid request and response and the win notice and response. Other interactions (e.g., block list synchronization, traffic control) are candidates for future initiatives or are already defined by OpenRTB.

2.1 Transport

The base protocol between an exchange and its bidders is HTTP. Specifically, HTTP POST is required for bid requests to accommodate greater payloads than HTTP GET and facilitate the use of binary representations. Win notices may be either HTTP POST or HTTP GET at the discretion of the exchange. All calls should return HTTP code 200 except for an empty bid response (i.e., the recommended method of specifying "no bid"), which should return HTTP code 204.

BEST PRACTICE: One of the simplest and most effective ways of improving connection performance is to enable HTTP Persistent Connections, also known as Keep-Alive. This has a profound impact on overall performance by reducing connection management overhead as well as CPU utilization on both sides of the interface.

2.2 Security

SSL (Secure Sockets Layer) is not required for compliance since these are server-to-server calls, which can be protected in other ways. Furthermore, SSL is not recommended due to the additional processing overhead.

2.3 Data Format

JSON (JavaScript Object Notation) is the suggested format for bid request and bid response data payloads. JSON was chosen for its combination of human readability and compactness. The data payloads are described in Section 3 and Section 4.

An exchange may offer additional representations to bidders who may prefer them. These might include a compressed form of JSON, XML, Apache Avro, ProtoBuf, Thrift, and many others.

The bid request specifies the representation as a mime type using the Content-Type HTTP header. The mime type for the standard JSON representation is "application/json" as shown. The format of the bid response must be the same as the bid request.

```
Content-Type: application/json
```

If alternative binary representations are used, the exchange or SSP should specify the Content-Type appropriately. For example: "Content-Type: avro/binary" or "Content-Type: application/x-protobuf". If the content-type is missing, the bidder should assume the type is application/json, unless a different default has been selected by an exchange.

As a convention, the absence of an attribute has a formal meaning. In most cases, this indicates that the value is unknown, unless otherwise specified.

2.4 OpenRTB Version HTTP Header

The OpenRTB Version should be passed in the header of a bid request with a custom header parameter. This will allow bidders to recognize the version of the message contained before attempting to parse the request.

Additionally, it is recommended albeit optional that bidders place an identically formatted message in the HTTP header of the response with the protocol version the bidder has implemented. The message may contain a different version number than the request header.

```
x-openrtb-version: 2.3
```

This version should be specified as <major>.<minor> (e.g., 2.3). First or second level increments on the version are changes to the protocol. In general, second-level changes should be backwards compatible, whereas first level changes need not be backwards compatible. Any third level revisions (such as 2.0.1) should not change the protocol itself; only descriptions and notes that don't affect the protocol content. Third level versions should not be included in this header since they should have no technical impact.

2.5 Privacy by Design

The OpenRTB project fully supports privacy policies as specified by buyers and sellers of advertising. In particular OpenRTB supports do-not-track (Section 3.2.11), COPPA restriction signaling (Section 7.1), and the ability to pass user preferences from sellers to buyers through the User object (Section 3.2.13).

2.6 Relationship to IAB Quality Assurance Guidelines

OpenRTB is fully compatible with the IAB Quality Assurance Guidelines (QAG) available here: http://www.iab.net/ne_guidelines. In particular the taxonomies used in this specification are derived from the QAG.

2.7 Customization and Extensions

The OpenRTB spec allows for exchange specific customization and extensions of the specification. Any object may contain extensions. In order to keep extension fields consistent across platforms, they should consistently be named "`ext`".

3. Bid Request Specification

RTB transactions are initiated when an exchange or other supply source sends a bid request to a bidder. The bid request consists of the top-level bid request object, at least one impression object, and may optionally include additional objects providing impression context.

3.1 Object Model

Following is the object model for the bid request. The top-level object (i.e., in JSON the unnamed outer object) is denoted as `BidRequest` in the model. Of its direct subordinates, only `Imp` is technically required since it is fundamental to describing the impression being sold and its requires at least one of `Banner`, `Video`, and `Native` to define the type of impression (i.e., whichever one or more the publisher is willing to accept; although a bid will be for exactly one of those specified). An impression can optionally be subject to a set of private marketplace.

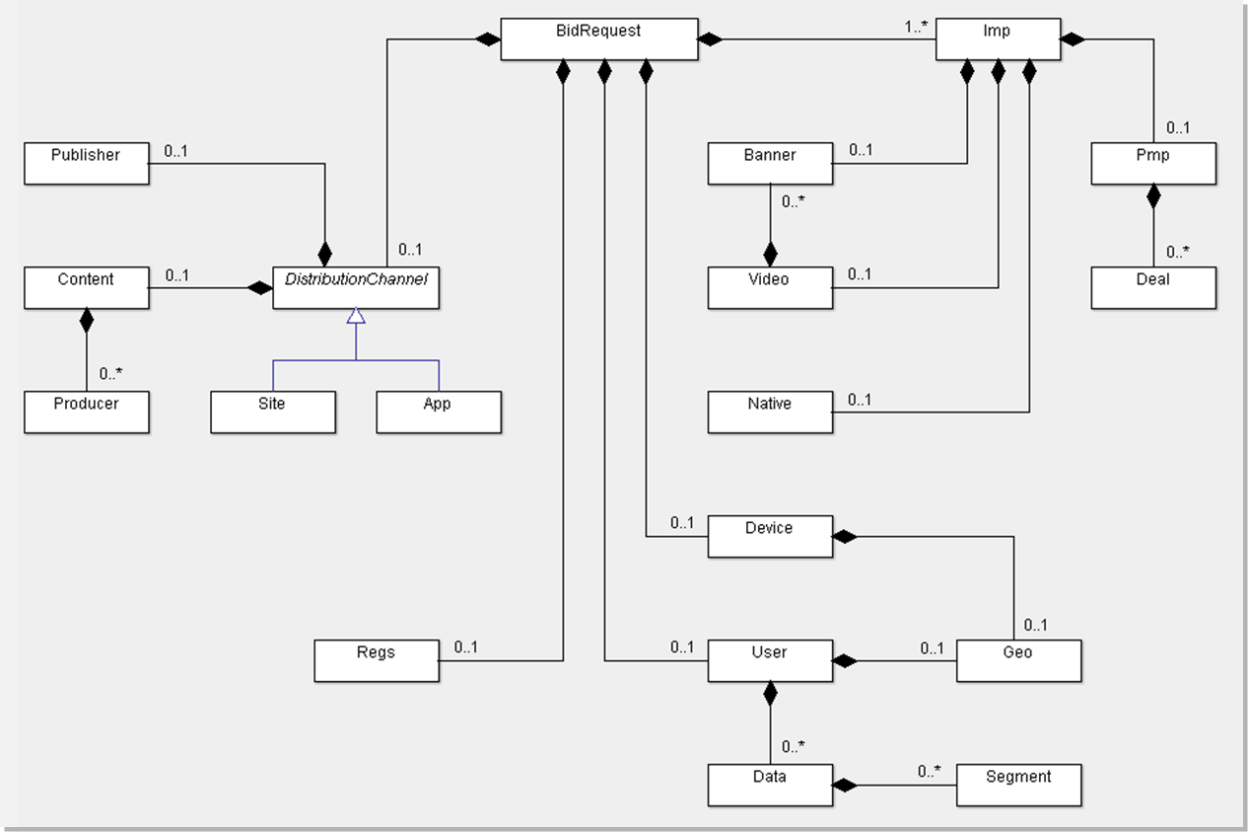

Figure 4: Bid Request object model.

Other subordinates to the BidRequest provide various forms of information to assist bidders in making targeting and pricing decisions. This includes details about the user, the device they're using, the location of either, regulatory constraints, and the content and media in which the impression will occur.

On the latter, there is the distinction between site (i.e., website) and application (i.e., non-browser app typically in mobile). The abstract class called `DistributionChannel` is just a modeling concept to indicate that a BidRequest is related to either a `Site` or an `App`, but not both (i.e., a distribution channel

is an abstraction of site and app). Both sites and apps can be further described by data about their publisher, the content, and the content's producer.

Not shown in the model figure is an extensions object. This is an object of undefined structure that can be added to any other object to convey exchange-specific extensions to the standard. Exchanges using these objects are responsible for publishing their extensions to their bidders.

The following table summarizes the objects in the Bid Request model and serves as an index into the detailed definitions in the subsections that follow.

Object	Section	Description
`BidRequest`	3.2.1	Top-level object.
`Imp`	3.2.2	Container for the description of a specific impression; at least 1 per request.
`Banner`	3.2.3	Details for a banner impression (incl. in-banner video) or video companion ad.
`Video`	3.2.4	Details for a video impression or the video asset of a native impression.
`Native`	3.2.5	Container for a native impression conforming to the Native Ad Spec.
`Site`	3.2.6	Details of the website calling for the impression.
`App`	3.2.7	Details of the application calling for the impression.
`Publisher`	3.2.8	Entity that controls the content of and distributes the site or app.
`Content`	3.2.9	Details about the published content itself, within which the ad will be shown.
`Producer`	3.2.10	Producer of the content; not necessarily the publisher (e.g., syndication).
`Device`	3.2.11	Details of the device on which the content and impressions are displayed.
`Geo`	3.2.12	Location of the device or user's home base depending on the parent object.
`User`	3.2.13	Human user of the device; audience for advertising.
`Data`	3.2.14	Collection of additional user targeting data from a specific data source.
`Segment`	3.2.15	Specific data point about a user from a specific data source.
`Regs`	3.2.16	Regulatory conditions in effect for all impressions in this bid request.
`Pmp`	3.2.17	Collection of private marketplace (PMP) deals applicable to this impression.
`Deal`	3.2.18	Deal terms pertaining to this impression between a seller and buyer.

3.2 Object Specifications

The subsections that follow define each of the objects in the bid request model. Several conventions are used throughout:

- Attributes are "required" if their omission would technically break the protocol.
- Some optional attributes are denoted "recommended" due to their elevated business importance.
- Unless a default value is explicitly specified, an omitted attribute is interpreted as "unknown".

3.2.1 Object: BidRequest

The top-level bid request object contains a globally unique bid request or auction ID. This `id` attribute is required as is at least one impression object (Section 3.2.2). Other attributes in this top-level object establish rules and restrictions that apply to all impressions being offered.

There are also several subordinate objects that provide detailed data to potential buyers. Among these are the `Site` and `App` objects, which describe the type of published media in which the impression(s) appear. These objects are highly recommended, but only one applies to a given bid request depending on whether the media is browser-based web content or a non-browser application, respectively.

Attribute	Type	Description
`id`	string; required	Unique ID of the bid request, provided by the exchange.
`imp`	object array; required	Array of `Imp` objects (Section 3.2.2) representing the impressions offered. At least 1 `Imp` object is required.
`site`	object; recommended	Details via a `Site` object (Section 3.2.6) about the publisher's website. Only applicable and recommended for websites.
`app`	object; recommended	Details via an `App` object (Section 3.2.7) about the publisher's app (i.e., non-browser applications). Only applicable and recommended for apps.
`device`	object; recommended	Details via a `Device` object (Section 3.2.11) about the user's device to which the impression will be delivered.
`user`	object; recommended	Details via a `User` object (Section 3.2.13) about the human user of the device; the advertising audience.
`test`	integer; default 0	Indicator of test mode in which auctions are not billable, where 0 = live mode, 1 = test mode.
`at`	integer; default 2	Auction type, where 1 = First Price, 2 = Second Price Plus. Exchange-specific auction types can be defined using values greater than 500.
`tmax`	integer	Maximum time in milliseconds to submit a bid to avoid timeout. This value is commonly communicated offline.
`wseat`	string array	Whitelist of buyer seats allowed to bid on this impression. Seat IDs must be communicated between bidders and the exchange *a priori*. Omission implies no seat restrictions.
`allimps`	integer; default 0	Flag to indicate if Exchange can verify that the impressions offered represent all of the impressions available in context (e.g., all on the web page, all video spots such as pre/mid/post roll) to support road-blocking. 0 = no or unknown, 1 = yes, the impressions offered represent all that are available.
`cur`	string array	Array of allowed currencies for bids on this bid request using ISO-4217 alpha codes. Recommended only if the exchange accepts multiple currencies.
`bcat`	string array	Blocked advertiser categories using the IAB content categories. Refer to List 5.1.
`badv`	string array	Block list of advertisers by their domains (e.g., "ford.com").

regs	object	A Regs object (Section 3.2.16) that specifies any industry, legal, or governmental regulations in force for this request.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.2 Object: Imp

This object describes an ad placement or impression being auctioned. A single bid request can include multiple `Imp` objects, a use case for which might be an exchange that supports selling all ad positions on a given page. Each `Imp` object has a required ID so that bids can reference them individually.

The presence of `Banner` (Section 3.2.3), `Video` (Section 3.2.4), and/or `Native` (Section 3.2.5) objects subordinate to the `Imp` object indicates the type of impression being offered. The publisher can choose one such type which is the typical case or mix them at their discretion. However, any given bid for the impression must conform to one of the offered types.

Attribute	Type	Description
id	string; required	A unique identifier for this impression within the context of the bid request (typically, starts with 1 and increments.
banner	object	A `Banner` object (Section 3.2.3); required if this impression is offered as a banner ad opportunity.
video	object	A `Video` object (Section 3.2.4); required if this impression is offered as a video ad opportunity.
native	object	A `Native` object (Section 3.2.5); required if this impression is offered as a native ad opportunity.
displaymanager	string	Name of ad mediation partner, SDK technology, or player responsible for rendering ad (typically video or mobile). Used by some ad servers to customize ad code by partner. Recommended for video and/or apps.
displaymanagerver	string	Version of ad mediation partner, SDK technology, or player responsible for rendering ad (typically video or mobile). Used by some ad servers to customize ad code by partner. Recommended for video and/or apps.
instl	integer; default 0	1 = the ad is interstitial or full screen, 0 = not interstitial.
tagid	string	Identifier for specific ad placement or ad tag that was used to initiate the auction. This can be useful for debugging of any issues, or for optimization by the buyer.
bidfloor	float; default 0	Minimum bid for this impression expressed in CPM.
bidfloorcur	string; default “USD”	Currency specified using ISO-4217 alpha codes. This may be different from bid currency returned by bidder if this is allowed by the exchange.
secure	integer	Flag to indicate if the impression requires secure HTTPS URL creative assets and markup, where 0 = non-secure, 1 = secure. If omitted, the secure state is unknown, but non-secure HTTP support can be assumed.
iframebuster	string array	Array of exchange-specific names of supported iframe busters.

pmp	object	A Pmp object (Section 3.2.17) containing any private marketplace deals in effect for this impression.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.3 Object: Banner

This object represents the most general type of impression. Although the term "banner" may have very specific meaning in other contexts, here it can be many things including a simple static image, an expandable ad unit, or even in-banner video (refer to the Video object in Section 3.2.4 for the more generalized and full featured video ad units). An array of Banner objects can also appear within the Video to describe optional companion ads defined in the VAST specification.

The presence of a Banner as a subordinate of the Imp object indicates that this impression is offered as a banner type impression. At the publisher's discretion, that same impression may also be offered as video and/or native by also including as Imp subordinates the Video and/or Native objects, respectively. However, any given bid for the impression must conform to one of the offered types.

Attribute	Type	Description
w	integer; recommended	Width of the impression in pixels. If neither wmin nor wmax are specified, this value is an exact width requirement. Otherwise it is a preferred width.
h	integer; recommended	Height of the impression in pixels. If neither hmin nor hmax are specified, this value is an exact height requirement. Otherwise it is a preferred height.
wmax	integer	Maximum width of the impression in pixels. If included along with a w value then w should be interpreted as a recommended or preferred width.
hmax	integer	Maximum height of the impression in pixels. If included along with an h value then h should be interpreted as a recommended or preferred height.
wmin	integer	Minimum width of the impression in pixels. If included along with a w value then w should be interpreted as a recommended or preferred width.
hmin	integer	Minimum height of the impression in pixels. If included along with an h value then h should be interpreted as a recommended or preferred height.
id	string	Unique identifier for this banner object. Recommended when Banner objects are used with a Video object (Section 3.2.4) to represent an array of companion ads. Values usually start at 1 and increase with each object; should be unique within an impression.
btype	integer array	Blocked banner ad types. Refer to List 5.2.
battr	integer array	Blocked creative attributes. Refer to List 5.3.
pos	integer	Ad position on screen. Refer to List 5.4.

mimes	string array	Content MIME types supported. Popular MIME types may include "application/x-shockwave-flash", "image/jpg", and "image/gif".
topframe	integer	Indicates if the banner is in the top frame as opposed to an iframe, where 0 = no, 1 = yes.
expdir	integer array	Directions in which the banner may expand. Refer to List 5.5.
api	integer array	List of supported API frameworks for this impression. Refer to List 5.6. If an API is not explicitly listed, it is assumed not to be supported.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.4 Object: Video

This object represents an in-stream video impression. Many of the fields are non-essential for minimally viable transactions, but are included to offer fine control when needed. Video in OpenRTB generally assumes compliance with the VAST standard. As such, the notion of companion ads is supported by optionally including an array of Banner objects (refer to the Banner object in Section 3.2.3) that define these companion ads.

The presence of a Video as a subordinate of the Imp object indicates that this impression is offered as a video type impression. At the publisher's discretion, that same impression may also be offered as banner and/or native by also including as Imp subordinates the Banner and/or Native objects, respectively. However, any given bid for the impression must conform to one of the offered types.

Attribute	Type	Description
mimes	string array; required	Content MIME types supported. Popular MIME types may include "video/x-ms-wmv" for Windows Media and "video/x-flv" for Flash Video.
minduration	integer; recommended	Minimum video ad duration in seconds.
maxduration	integer; recommended	Maximum video ad duration in seconds.
protocol	integer; DEPRECATED	*NOTE: Use of protocols instead is highly recommended.* Supported video bid response protocol. Refer to List 5.8. At least one supported protocol must be specified in either the protocol or protocols attribute.
protocols	integer array; recommended	Array of supported video bid response protocols. Refer to List 5.8. At least one supported protocol must be specified in either the protocol or protocols attribute.
w	integer; recommended	Width of the video player in pixels.
h	integer; recommended	Height of the video player in pixels.
startdelay	integer; recommended	Indicates the start delay in seconds for pre-roll, mid-roll, or post-roll ad placements. Refer to List 5.10 for additional generic values.

linearity	integer	Indicates if the impression must be linear, nonlinear, etc. If none specified, assume all are allowed. Refer to List 5.7.
sequence	integer	If multiple ad impressions are offered in the same bid request, the sequence number will allow for the coordinated delivery of multiple creatives.
battr	integer array	Blocked creative attributes. Refer to List 5.3.
maxextended	integer	Maximum extended video ad duration if extension is allowed. If blank or 0, extension is not allowed. If -1, extension is allowed, and there is no time limit imposed. If greater than 0, then the value represents the number of seconds of extended play supported beyond the maxduration value.
minbitrate	integer	Minimum bit rate in Kbps. Exchange may set this dynamically or universally across their set of publishers.
maxbitrate	integer	Maximum bit rate in Kbps. Exchange may set this dynamically or universally across their set of publishers.
boxingallowed	integer; default 1	Indicates if letter-boxing of 4:3 content into a 16:9 window is allowed, where 0 = no, 1 = yes.
playbackmethod	integer array	Allowed playback methods. If none specified, assume all are allowed. Refer to List 5.9.
delivery	integer array	Supported delivery methods (e.g., streaming, progressive). If none specified, assume all are supported. Refer to List 5.13.
pos	integer	Ad position on screen. Refer to List 5.4.
companionad	object array	Array of Banner objects (Section 3.2.3) if companion ads are available.
api	integer array	List of supported API frameworks for this impression. Refer to List 5.6. If an API is not explicitly listed, it is assumed not to be supported.
companiontype	integer array	Supported VAST companion ad types. Refer to List 5.12. Recommended if companion Banner objects are included via the companionad array.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.5 Object: Native

This object represents a native type impression. Native ad units are intended to blend seamlessly into the surrounding content (e.g., a sponsored Twitter or Facebook post). As such, the response must be well-structured to afford the publisher fine-grained control over rendering.

The Native Subcommittee has developed a companion specification to OpenRTB called the Native Ad Specification. It defines the request parameters and response markup structure of native ad units. This object provides the means of transporting request parameters as an opaque string so that the specific parameters can evolve separately under the auspices of the Native Ad Specification. Similarly, the ad markup served will be structured according to that specification.

The presence of a Native as a subordinate of the Imp object indicates that this impression is offered as a native type impression. At the publisher's discretion, that same impression may also be offered as

banner and/or video by also including as `Imp` subordinates the `Banner` and/or `Video` objects, respectively. However, any given bid for the impression must conform to one of the offered types.

Attribute	Type	Description
`request`	string; required	Request payload complying with the Native Ad Specification.
`ver`	string; recommended	Version of the Native Ad Specification to which `request` complies; highly recommended for efficient parsing.
`api`	integer array	List of supported API frameworks for this impression. Refer to List 5.6. If an API is not explicitly listed, it is assumed not to be supported.
`battr`	integer array	Blocked creative attributes. Refer to List 5.3.
`ext`	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.6 Object: Site

This object should be included if the ad supported content is a website as opposed to a non-browser application. A bid request must not contain both a `Site` and an `App` object. At a minimum, it is useful to provide a site ID or page URL, but this is not strictly required.

Attribute	Type	Description
`id`	string; recommended	Exchange-specific site ID.
`name`	string	Site name (may be aliased at the publisher's request).
`domain`	string	Domain of the site (e.g., "mysite.foo.com").
`cat`	string array	Array of IAB content categories of the site. Refer to List 5.1.
`sectioncat`	string array	Array of IAB content categories that describe the current section of the site. Refer to List 5.1.
`pagecat`	string array	Array of IAB content categories that describe the current page or view of the site. Refer to List 5.1.
`page`	string	URL of the page where the impression will be shown.
`ref`	string	Referrer URL that caused navigation to the current page.
`search`	string	Search string that caused navigation to the current page.
`mobile`	integer	Mobile-optimized signal, where 0 = no, 1 = yes.
`privacypolicy`	integer	Indicates if the site has a privacy policy, where 0 = no, 1 = yes.
`publisher`	object	Details about the Publisher (Section 3.2.8) of the site.
`content`	object	Details about the Content (Section 3.2.9) within the site.
`keywords`	string	Comma separated list of keywords about the site.
`ext`	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.7 Object: App

This object should be included if the ad supported content is a non-browser application (typically in mobile) as opposed to a website. A bid request must not contain both an `App` and a `Site` object. At a minimum, it is useful to provide an App ID or bundle, but this is not strictly required.

Attribute	Type	Description
`id`	string; recommended	Exchange-specific app ID.
`name`	string	App name (may be aliased at the publisher's request).
`bundle`	string	Application bundle or package name (e.g., com.foo.mygame); intended to be a unique ID across exchanges.
`domain`	string	Domain of the app (e.g., "mygame.foo.com").
`storeurl`	string	App store URL for an installed app; for QAG 1.5 compliance.
`cat`	string array	Array of IAB content categories of the app. Refer to List 5.1.
`sectioncat`	string array	Array of IAB content categories that describe the current section of the app. Refer to List 5.1.
`pagecat`	string array	Array of IAB content categories that describe the current page or view of the app. Refer to List 5.1.
`ver`	string	Application version.
`privacypolicy`	integer	Indicates if the app has a privacy policy, where 0 = no, 1 = yes.
`paid`	integer	0 = app is free, 1 = the app is a paid version.
`publisher`	object	Details about the Publisher (Section 3.2.8) of the app.
`content`	object	Details about the Content (Section 3.2.9) within the app.
`keywords`	string	Comma separated list of keywords about the app.
`ext`	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.8 Object: Publisher

This object describes the publisher of the media in which the ad will be displayed. The publisher is typically the seller in an OpenRTB transaction.

Attribute	Type	Description
`id`	string	Exchange-specific publisher ID.
`name`	string	Publisher name (may be aliased at the publisher's request).
`cat`	string array	Array of IAB content categories that describe the publisher. Refer to List 5.1.
`domain`	string	Highest level domain of the publisher (e.g., "publisher.com").
`ext`	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.9 Object: Content

This object describes the content in which the impression will appear, which may be syndicated or non-syndicated content. This object may be useful when syndicated content contains impressions and does not necessarily match the publisher's general content. The exchange might or might not have knowledge of the page where the content is running, as a result of the syndication method. For example might be a video impression embedded in an iframe on an unknown web property or device.

Attribute	Type	Description
`id`	string	ID uniquely identifying the content.
`episode`	integer	Episode number (typically applies to video content).
`title`	string	Content title. *Video Examples:* "Search Committee" (television), "A New Hope" (movie), or "Endgame" (made for web). *Non-Video Example:* "Why an Antarctic Glacier Is Melting So Quickly" (Time magazine article).
`series`	string	Content series. *Video Examples:* "The Office" (television), "Star Wars" (movie), or "Arby 'N' The Chief" (made for web). *Non-Video Example:* "Ecocentric" (Time Magazine blog).
`season`	string	Content season; typically for video content (e.g., "Season 3").
`producer`	object	Details about the content `Producer` (Section 3.2.10).
`url`	string	URL of the content, for buy-side contextualization or review.
`cat`	string array	Array of IAB content categories that describe the content producer. Refer to List 5.1.
`videoquality`	integer	Video quality per IAB's classification. Refer to List 5.11.
`context`	integer	Type of content (game, video, text, etc.). Refer to List 5.14.
`contentrating`	string	Content rating (e.g., MPAA).
`userrating`	string	User rating of the content (e.g., number of stars, likes, etc.).
`qagmediarating`	integer	Media rating per QAG guidelines. Refer to List 5.15.
`keywords`	string	Comma separated list of keywords describing the content.
`livestream`	integer	0 = not live, 1 = content is live (e.g., stream, live blog).
`sourcerelationship`	integer	0 = indirect, 1 = direct.
`len`	integer	Length of content in seconds; appropriate for video or audio.
`language`	string	Content language using ISO-639-1-alpha-2.
`embeddable`	integer	Indicator of whether or not the content is embeddable (e.g., an embeddable video player), where 0 = no, 1 = yes.
`ext`	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.10 Object: Producer

This object defines the producer of the content in which the ad will be shown. This is particularly useful when the content is syndicated and may be distributed through different publishers and thus when the producer and publisher are not necessarily the same entity.

Attribute	Type	Description
id	string	Content producer or originator ID. Useful if content is syndicated and may be posted on a site using embed tags.
name	string	Content producer or originator name (e.g., "Warner Bros").
cat	string array	Array of IAB content categories that describe the content producer. Refer to List 5.1.
domain	string	Highest level domain of the content producer (e.g., "producer.com").
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.11 Object: Device

This object provides information pertaining to the device through which the user is interacting. Device information includes its hardware, platform, location, and carrier data. The device can refer to a mobile handset, a desktop computer, set top box, or other digital device.

Attribute	Type	Description
ua	string; recommended	Browser user agent string.
geo	object; recommended	Location of the device assumed to be the user's current location defined by a Geo object (Section 3.2.12).
dnt	integer; recommended	Standard "Do Not Track" flag as set in the header by the browser, where 0 = tracking is unrestricted, 1 = do not track.
lmt	integer; recommended	"Limit Ad Tracking" signal commercially endorsed (e.g., iOS, Android), where 0 = tracking is unrestricted, 1 = tracking must be limited per commercial guidelines.
ip	string; recommended	IPv4 address closest to device.
ipv6	string	IP address closest to device as IPv6.
devicetype	integer	The general type of device. Refer to List 5.17.
make	string	Device make (e.g., "Apple").
model	string	Device model (e.g., "iPhone").
os	string	Device operating system (e.g., "iOS").
osv	string	Device operating system version (e.g., "3.1.2").
hwv	string	Hardware version of the device (e.g., "5S" for iPhone 5S).
h	integer	Physical height of the screen in pixels.
w	integer	Physical width of the screen in pixels.

ppi	integer	Screen size as pixels per linear inch.
pxratio	float	The ratio of physical pixels to device independent pixels.
js	integer	Support for JavaScript, where 0 = no, 1 = yes.
flashver	string	Version of Flash supported by the browser.
language	string	Browser language using ISO-639-1-alpha-2.
carrier	string	Carrier or ISP (e.g., "VERIZON"). "WIFI" is often used in mobile to indicate high bandwidth (e.g., video friendly vs. cellular).
connectiontype	integer	Network connection type. Refer to List 5.18.
ifa	string	ID sanctioned for advertiser use in the clear (i.e., not hashed).
didsha1	string	Hardware device ID (e.g., IMEI); hashed via SHA1.
didmd5	string	Hardware device ID (e.g., IMEI); hashed via MD5.
dpidsha1	string	Platform device ID (e.g., Android ID); hashed via SHA1.
dpidmd5	string	Platform device ID (e.g., Android ID); hashed via MD5.
macsha1	string	MAC address of the device; hashed via SHA1.
macmd5	string	MAC address of the device; hashed via MD5.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

BEST PRACTICE: There are currently no prominent open source lists for device makes, models, operating systems, or carriers. Exchanges typically use commercial products or other proprietary lists for these attributes. Until suitable open standards are available, exchanges are highly encouraged to publish lists of their device make, model, operating system, and carrier values to bidders.

BEST PRACTICE: Proper device IP detection in mobile is not straightforward. Typically it involves starting at the left of the `x-forwarded-for` header, skipping private carrier networks (e.g., `10.x.x.x` or `192.x.x.x`), and possibly scanning for known carrier IP ranges. Exchanges are urged to research and implement this feature carefully when presenting device IP values to bidders.

3.2.12 Object: Geo

This object encapsulates various methods for specifying a geographic location. When subordinate to a `Device` object, it indicates the location of the device which can also be interpreted as the user's current location. When subordinate to a `User` object, it indicates the location of the user's home base (i.e., not necessarily their current location).

The `lat`/`lon` attributes should only be passed if they conform to the accuracy depicted in the `type` attribute. For example, the centroid of a geographic region such as postal code should not be passed.

Attribute	Type	Description
lat	float	Latitude from -90.0 to +90.0, where negative is south.
lon	float	Longitude from -180.0 to +180.0, where negative is west.
type	integer	Source of location data; recommended when passing `lat`/`lon`. Refer to List 5.16.
country	string	Country code using ISO-3166-1-alpha-3.

region	string	Region code using ISO-3166-2; 2-letter state code if USA.
regionfips104	string	Region of a country using FIPS 10-4 notation. While OpenRTB supports this attribute, it has been withdrawn by NIST in 2008.
metro	string	Google metro code; similar to but not exactly Nielsen DMAs. See Appendix A for a link to the codes.
city	string	City using United Nations Code for Trade & Transport Locations. See Appendix A for a link to the codes.
zip	string	Zip or postal code.
utcoffset	integer	Local time as the number +/- of minutes from UTC.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.13 Object: User

This object contains information known or derived about the human user of the device (i.e., the audience for advertising). The user `id` is an exchange artifact and may be subject to rotation or other privacy policies. However, this user ID must be stable long enough to serve reasonably as the basis for frequency capping and retargeting.

Attribute	Type	Description
id	string; recommended	Exchange-specific ID for the user. At least one of `id` or `buyeruid` is recommended.
buyeruid	string; recommended	Buyer-specific ID for the user as mapped by the exchange for the buyer. At least one of `buyeruid` or `id` is recommended.
yob	integer	Year of birth as a 4-digit integer.
gender	string	Gender, where "M" = male, "F" = female, "O" = known to be other (i.e., omitted is unknown).
keywords	string	Comma separated list of keywords, interests, or intent.
customdata	string	Optional feature to pass bidder data that was set in the exchange's cookie. The string must be in base85 cookie safe characters and be in any format. Proper JSON encoding must be used to include "escaped" quotation marks.
geo	object	Location of the user's home base defined by a `Geo` object (Section 3.2.12). This is not necessarily their current location.
data	object array	Additional user data. Each `Data` object (Section 3.2.14) represents a different data source.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.14 Object: Data

The data and segment objects together allow additional data about the user to be specified. This data may be from multiple sources whether from the exchange itself or third party providers as specified by the `id` field. A bid request can mix data objects from multiple providers. The specific data providers in use should be published by the exchange *a priori* to its bidders.

Attribute	Type	Description
id	string	Exchange-specific ID for the data provider.
name	string	Exchange-specific name for the data provider.
segment	object array	Array of Segment (Section 3.2.15) objects that contain the actual data values.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.15 Object: Segment

Segment objects are essentially key-value pairs that convey specific units of data about the user. The parent Data object is a collection of such values from a given data provider. The specific segment names and value options must be published by the exchange *a priori* to its bidders.

Attribute	Type	Description
id	string	ID of the data segment specific to the data provider.
name	string	Name of the data segment specific to the data provider.
value	string	String representation of the data segment value.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.16 Object: Regs

This object contains any legal, governmental, or industry regulations that apply to the request. The coppa flag signals whether or not the request falls under the United States Federal Trade Commission's regulations for the United States Children's Online Privacy Protection Act ("COPPA"). Refer to Section 7.1 for more information.

Attribute	Type	Description
coppa	integer	Flag indicating if this request is subject to the COPPA regulations established by the USA FTC, where 0 = no, 1 = yes.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.17 Object: Pmp

This object is the private marketplace container for direct deals between buyers and sellers that may pertain to this impression. The actual deals are represented as a collection of Deal objects. Refer to Section 7.2 for more details.

Attribute	Type	Description
private_auction	integer	Indicator of auction eligibility to seats named in the Direct Deals object, where 0 = all bids are accepted, 1 = bids are restricted to the deals specified and the terms thereof.
deals	object array	Array of Deal (Section 3.2.18) objects that convey the specific deals applicable to this impression.

ext	object	Placeholder for exchange-specific extensions to OpenRTB.

3.2.18 Object: Deal

This object constitutes a specific deal that was struck *a priori* between a buyer and a seller. Its presence with the Pmp collection indicates that this impression is available under the terms of that deal. Refer to Section 7.2 for more details.

Attribute	Type	Description
id	string; required	A unique identifier for the direct deal.
bidfloor	float; default 0	Minimum bid for this impression expressed in CPM.
bidfloorcur	string; default "USD"	Currency specified using ISO-4217 alpha codes. This may be different from bid currency returned by bidder if this is allowed by the exchange.
at	integer	Optional override of the overall auction type of the bid request, where 1 = First Price, 2 = Second Price Plus, 3 = the value passed in bidfloor is the agreed upon deal price. Additional auction types can be defined by the exchange.
wseat	string array	Whitelist of buyer seats allowed to bid on this deal. Seat IDs must be communicated between bidders and the exchange *a priori*. Omission implies no seat restrictions.
wadomain	string array	Array of advertiser domains (e.g., advertiser.com) allowed to bid on this deal. Omission implies no advertiser restrictions.
ext	object	Placeholder for exchange-specific extensions to OpenRTB.

4. Bid Response Specification

RTB responses contain bids that reference specific impressions within a bid request. Bids are in essence an offer to buy. The bid response consists of the top-level bid response object and optional objects that depict the specific bids. An empty HTTP response constitutes a no-bid and is in fact the most bandwidth friendly form of this signal. A malformed response or a response that contains no actual bids will also be interpreted as no-bid.

4.1 Object Model

Following is the object model for the bid response. The top-level object (i.e., in JSON the unnamed outer object) is denoted as `BidResponse` in the model. A bid response may contain bids from multiple "seats" (i.e., the buying entity upstream from the actual bidder). In fact a response may contain multiple bids from the same seat; typically but not necessarily from different campaigns. This can improve the seat's chances of winning since most exchanges enforce various block lists on behalf of their publishers.

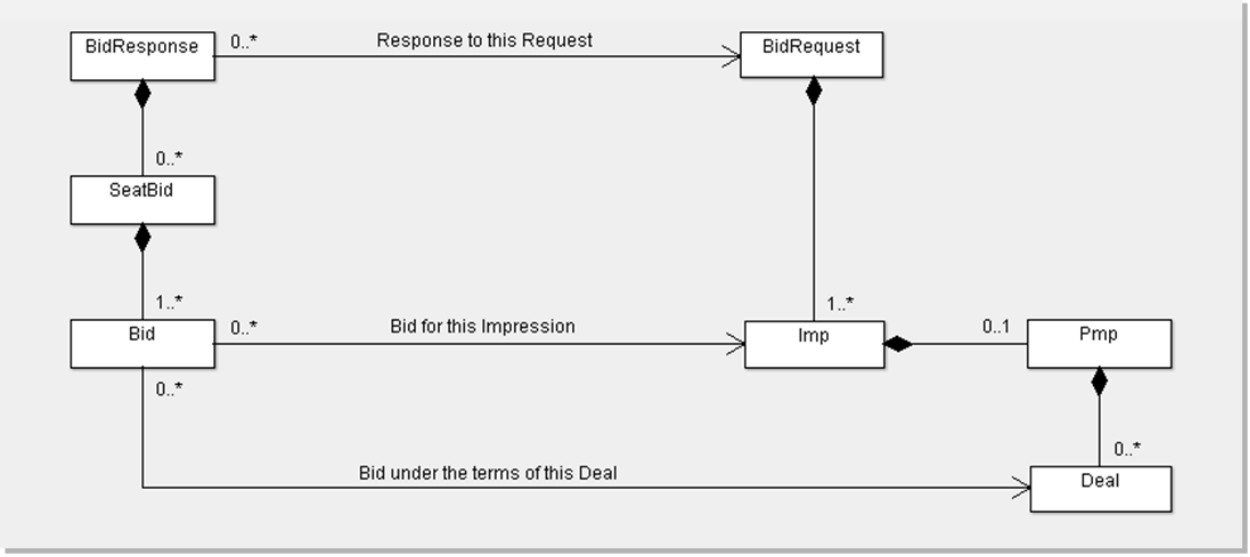

Figure 5: Bid Response object model.

Referring to the figure, the actual response objects are shown on the left, specifically the `BidResponse` top level object the seat specific `SeatBid` collections of `Bid` objects. The other objects shown are those objects from the bid request to which response objects related. Specifically, `BidResponse` includes the `BidRequest` ID for positive tracking purposes, and since a request can include multiple impressions `Bid` includes the ID of the `Imp` for which the bid is an offer to purchase. If a bid is made under the terms of a private marketplace deal, the `Bid` also includes the ID of the specific `Deal` object.

Not shown in the model figure is an extensions object. This is an object of undefined structure that can be added to any other object to convey bidder-specific extensions to the standard. Bidders using these objects are responsible for publishing their extensions to their exchanges.

The following table summarizes the objects in the Bid Response model and serves as an index into the detailed definitions in the subsections that follow.

Object	Section	Description
BidResponse	4.2.1	Top-level object.
SeatBid	4.2.2	Collection of bids made by the bidder on behalf of a specific seat.
Bid	4.2.3	An offer to buy a specific impression under certain business terms.

4.2 Object Specifications

The subsections that follow define each of the objects in the bid response model. Several conventions are used throughout:

- Attributes are "required" if their omission would technically break the protocol.
- Some optional attributes are denoted "recommended" due to their elevated business importance.
- Unless a default value is explicitly specified, an omitted attribute is interpreted as "unknown".

4.2.1 Object: BidResponse

This object is the top-level bid response object (i.e., the unnamed outer JSON object). The `id` attribute is a reflection of the bid request ID for logging purposes. Similarly, `bidid` is an optional response tracking ID for bidders. If specified, it can be included in the subsequent win notice call if the bidder wins. At least one `seatbid` object is required, which contains at least one bid for an impression. Other attributes are optional.

To express a "no-bid", the options are to return an empty response with HTTP 204. Alternately if the bidder wishes to convey to the exchange a reason for not bidding, just a `BidResponse` object is returned with a reason code in the `nbr` attribute.

Attribute	Type	Description
id	string; required	ID of the bid request to which this is a response.
seatbid	object array	Array of seatbid objects; 1+ required if a bid is to be made.
bidid	string	Bidder generated response ID to assist with logging/tracking.
cur	string; default "USD"	Bid currency using ISO-4217 alpha codes.
customdata	string	Optional feature to allow a bidder to set data in the exchange's cookie. The string must be in base85 cookie safe characters and be in any format. Proper JSON encoding must be used to include "escaped" quotation marks.
nbr	integer	Reason for not bidding. Refer to List 5.19.
ext	object	Placeholder for bidder-specific extensions to OpenRTB.

4.2.2 Object: SeatBid

A bid response can contain multiple `SeatBid` objects, each on behalf of a different bidder seat and each containing one or more individual bids. If multiple impressions are presented in the request, the `group` attribute can be used to specify if a seat is willing to accept any impressions that it can win (default) or if it is only interested in winning any if it can win them all as a group.

Attribute	Type	Description
bid	object array; required	Array of 1+ Bid objects (Section 4.2.3) each related to an impression. Multiple bids can relate to the same impression.
seat	string	ID of the bidder seat on whose behalf this bid is made.
group	integer; default 0	0 = impressions can be won individually; 1 = impressions must be won or lost as a group.
ext	object	Placeholder for bidder-specific extensions to OpenRTB.

4.2.3 Object: Bid

A SeatBid object contains one or more Bid objects, each of which relates to a specific impression in the bid request via the impid attribute and constitutes an offer to buy that impression for a given price.

Attribute	Type	Description
id	string; required	Bidder generated bid ID to assist with logging/tracking.
impid	string; required	ID of the Imp object in the related bid request.
price	float; required	Bid price expressed as CPM although the actual transaction is for a unit impression only. Note that while the type indicates float, integer math is highly recommended when handling currencies (e.g., BigDecimal in Java).
adid	string	ID of a preloaded ad to be served if the bid wins.
nurl	string	Win notice URL called by the exchange if the bid wins; optional means of serving ad markup.
adm	string	Optional means of conveying ad markup in case the bid wins; supersedes the win notice if markup is included in both.
adomain	string array	Advertiser domain for block list checking (e.g., "ford.com"). This can be an array of for the case of rotating creatives. Exchanges can mandate that only one domain is allowed.
bundle	string	Bundle or package name (e.g., com.foo.mygame) of the app being advertised, if applicable; intended to be a unique ID across exchanges.
iurl	string	URL without cache-busting to an image that is representative of the content of the campaign for ad quality/safety checking.
cid	string	Campaign ID to assist with ad quality checking; the collection of creatives for which iurl should be representative.
crid	string	Creative ID to assist with ad quality checking.
cat	string array	IAB content categories of the creative. Refer to List 5.1.
attr	integer array	Set of attributes describing the creative. Refer to List 5.3.
dealid	string	Reference to the deal.id from the bid request if this bid pertains to a private marketplace direct deal.
h	integer	Height of the creative in pixels.
w	integer	Width of the creative in pixels.

ext	object	Placeholder for bidder-specific extensions to OpenRTB.

For each bid, the nurl attribute contains the win notice URL. If the bidder wins the impression, the exchange calls this notice URL to inform the bidder of the win and to convey certain information using substitution macros (see Section 4.4) such as the clearing price. The win notice return or the adm attribute can be used to serve markup (see Section 4.3). In either case, the exchange will also apply the aforementioned substitution to any macros found in the markup.

Several other attribute are used for ad quality checks or enforcing publisher restrictions. These include the advertiser domain via adomain, a non-cache-busted URL to an image representative of the content of the campaign via iurl, an ID of the campaign and of the creative within the campaign via cid and crid respective, an array of creative attribute via attr, and the dimensions via h and w. If the bid pertains to a private marketplace deal, the dealid attribute is used to reference that agreement from the bid request.

4.3 Ad Serving Options

The fulfilment of an RTB transaction within the scope of this OpenRTB specification lies in the delivery of markup. Depending on the impression and other ad type constraints, this markup can be XHTML, XHTML with embedded JavaScript, a VAST document for video, a Native ad unit structure, and potentially other formats in the future.

The OpenRTB specification does not require any processing of the ad markup by the exchange other than macro substitution (refer to Section 4.4) and delivery to the supply-side. There are, however, multiple standard methods for transferring markup from the bidder to the exchange. The method used is at the discretion of the bidder, but an OpenRTB compliant exchange is expected to support all methods as defined in the subsections that follow.

4.3.1 Markup Served on the Win Notice

In this method, ad markup is returned to the exchange is via the win notice. In this case, the response body of the win notice call (i.e., invoking the bid.nurl attribute) contains the ad markup and only the ad markup; there must be no other structured data in the response body. Using this method, the bid.adm attribute must be omitted.

4.3.2 Markup Served in the Bid

In this method, ad markup is returned directly in the bid itself. This is accomplished via the bid.adm attribute. If both the adm attribute and win notice return data, the adm contents will take precedence.

4.3.3 Comparison of Ad Serving Approaches

Each of the ad serving methods has its own advantages that may be of varying importance to either the exchange or the bidder.

Ad Served on the Win Notice

- *Reduced Bandwidth Costs:* Serving ad markup only upon winning can save large amounts of bandwidth usage, the costs for which can be large at high volumes especially for exchanges.

- *Additional Bidder Flexibility:* Bidders may typically know the ad they will serve at the time of bid, but this provides an additional optional decision point after the clearing price has been established.

Ad Served in the Bid

- *Reduced Risk of Forfeiture:* A forfeit is the scenario in which a bidder wins, but forfeits due to failure to serve the ad markup. The risk of an additional HTTP failure (e.g., calling the win notice) is mitigated by this method.
- *Potential Concurrency:* The exchange can choose to return that ad markup and call the win notice concurrently, thereby improving user experience.

4.4 Substitution Macros

The win notice URL and its format are defined by the bidder. In order for the exchange to convey certain information to the winning bidder (e.g., the clearing price), a number of substitution macros can be inserted into the win notice URL definition. Prior to calling a win notice URL, the exchange will search the specified URL for any of the defined macros and replace them with the appropriate data. Note that the substitution is simple in the sense that wherever a legal macro is found, it will be replaced without regard for syntax correctness. Furthermore, if the source value is an optional parameter that was not specified, the macro will simply be removed (i.e., replaced with a zero-length string).

These same substitution macros can also be placed in the ad markup. The exchange will perform the same data substitutions as in the win notice URL. This occurs irrespective of whether the markup is returned on the win notice or passed in the `bid.adm` attribute of the bid response. A use case for macros in the ad markup might be when a bidder prefers to receive its official win notification from the device itself. To accomplish this, the bidder would include a tracking pixel in the ad markup, the URL for which would include any of the available macros.

Macro	Description
`${AUCTION_ID}`	ID of the bid request; from `BidRequest.id` attribute.
`${AUCTION_BID_ID}`	ID of the bid; from `BidResponse.bidid` attribute.
`${AUCTION_IMP_ID}`	ID of the impression just won; from `imp.id` attribute.
`${AUCTION_SEAT_ID}`	ID of the bidder seat for whom the bid was made.
`${AUCTION_AD_ID}`	ID of the ad markup the bidder wishes to serve; from `bid.adid` attribute.
`${AUCTION_PRICE}`	Settlement price using the same currency and units as the bid.
`${AUCTION_CURRENCY}`	The currency used in the bid (explicit or implied); for confirmation only.

Prior to substitution, macro data values can be encoded for security purposes using various obfuscation or encryption algorithms. This may be of particular interest for use cases such as the foregoing where price information is carried beyond the exchange, through the publisher, and into the device browser via a tracking pixel in the markup.

To specify that a particular macro is to be encoded, the suffix "`:X`" should be appended to the macro name, where X is a string that indicates the algorithm to be used. Algorithms choices are not defined by this specification and must be mutually agreed upon between exchange and bidder. As an example,

suppose that the price macro is to be encoded using Base64 and that its code is "`B64`". The macro would then be written as follows:

```
${AUCTION_PRICE:B64}
```

BEST PRACTICE: Encoding of macro data should be used sparingly due to the additional processing overhead. For communications strictly between exchange and bidder (e.g., a win notice called from the exchange), encoding is generally considered unnecessary.

5. Enumerated Lists Specification

All reference lists are actively maintained by the IAB on the OpenRTB website. As such, implementers should ensure they are working from the latest lists and enumerations.

5.1 Content Categories

The following list represents the IAB's contextual taxonomy for categorization. Standard IDs have been adopted to easily support the communication of primary and secondary categories for various objects.

This OpenRTB table has values derived from the IAB Quality Assurance Guidelines (QAG). Practitioners should keep in sync with updates to the QAG values as published on IAB.net.

Value	Description
IAB1	**Arts & Entertainment**
IAB1-1	Books & Literature
IAB1-2	Celebrity Fan/Gossip
IAB1-3	Fine Art
IAB1-4	Humor
IAB1-5	Movies
IAB1-6	Music
IAB1-7	Television
IAB2	**Automotive**
IAB2-1	Auto Parts
IAB2-2	Auto Repair
IAB2-3	Buying/Selling Cars
IAB2-4	Car Culture
IAB2-5	Certified Pre-Owned
IAB2-6	Convertible
IAB2-7	Coupe
IAB2-8	Crossover
IAB2-9	Diesel
IAB2-10	Electric Vehicle
IAB2-11	Hatchback
IAB2-12	Hybrid
IAB2-13	Luxury
IAB2-14	Minivan
IAB2-15	Motorcycles
IAB2-16	Off-Road Vehicles
IAB2-17	Performance Vehicles

IAB2-18	Pickup
IAB2-19	Road-Side Assistance
IAB2-20	Sedan
IAB2-21	Trucks & Accessories
IAB2-22	Vintage Cars
IAB2-23	Wagon
IAB3	**Business**
IAB3-1	Advertising
IAB3-2	Agriculture
IAB3-3	Biotech/Biomedical
IAB3-4	Business Software
IAB3-5	Construction
IAB3-6	Forestry
IAB3-7	Government
IAB3-8	Green Solutions
IAB3-9	Human Resources
IAB3-10	Logistics
IAB3-11	Marketing
IAB3-12	Metals
IAB4	**Careers**
IAB4-1	Career Planning
IAB4-2	College
IAB4-3	Financial Aid
IAB4-4	Job Fairs
IAB4-5	Job Search
IAB4-6	Resume Writing/Advice
IAB4-7	Nursing
IAB4-8	Scholarships
IAB4-9	Telecommuting
IAB4-10	U.S. Military
IAB4-11	Career Advice
IAB5	**Education**
IAB5-1	7-12 Education
IAB5-2	Adult Education
IAB5-3	Art History
IAB5-4	College Administration
IAB5-5	College Life

IAB5-6	Distance Learning
IAB5-7	English as a 2nd Language
IAB5-8	Language Learning
IAB5-9	Graduate School
IAB5-10	Homeschooling
IAB5-11	Homework/Study Tips
IAB5-12	K-6 Educators
IAB5-13	Private School
IAB5-14	Special Education
IAB5-15	Studying Business
IAB6	**Family & Parenting**
IAB6-1	Adoption
IAB6-2	Babies & Toddlers
IAB6-3	Daycare/Pre School
IAB6-4	Family Internet
IAB6-5	Parenting - K-6 Kids
IAB6-6	Parenting teens
IAB6-7	Pregnancy
IAB6-8	Special Needs Kids
IAB6-9	Eldercare
IAB7	**Health & Fitness**
IAB7-1	Exercise
IAB7-2	ADD
IAB7-3	AIDS/HIV
IAB7-4	Allergies
IAB7-5	Alternative Medicine
IAB7-6	Arthritis
IAB7-7	Asthma
IAB7-8	Autism/PDD
IAB7-9	Bipolar Disorder
IAB7-10	Brain Tumor
IAB7-11	Cancer
IAB7-12	Cholesterol
IAB7-13	Chronic Fatigue Syndrome
IAB7-14	Chronic Pain
IAB7-15	Cold & Flu
IAB7-16	Deafness

IAB7-17	Dental Care
IAB7-18	Depression
IAB7-19	Dermatology
IAB7-20	Diabetes
IAB7-21	Epilepsy
IAB7-22	GERD/Acid Reflux
IAB7-23	Headaches/Migraines
IAB7-24	Heart Disease
IAB7-25	Herbs for Health
IAB7-26	Holistic Healing
IAB7-27	IBS/Crohn's Disease
IAB7-28	Incest/Abuse Support
IAB7-29	Incontinence
IAB7-30	Infertility
IAB7-31	Men's Health
IAB7-32	Nutrition
IAB7-33	Orthopedics
IAB7-34	Panic/Anxiety Disorders
IAB7-35	Pediatrics
IAB7-36	Physical Therapy
IAB7-37	Psychology/Psychiatry
IAB7-38	Senior Health
IAB7-39	Sexuality
IAB7-40	Sleep Disorders
IAB7-41	Smoking Cessation
IAB7-42	Substance Abuse
IAB7-43	Thyroid Disease
IAB7-44	Weight Loss
IAB7-45	Women's Health
IAB8	**Food & Drink**
IAB8-1	American Cuisine
IAB8-2	Barbecues & Grilling
IAB8-3	Cajun/Creole
IAB8-4	Chinese Cuisine
IAB8-5	Cocktails/Beer
IAB8-6	Coffee/Tea
IAB8-7	Cuisine-Specific

IAB8-8	Desserts & Baking
IAB8-9	Dining Out
IAB8-10	Food Allergies
IAB8-11	French Cuisine
IAB8-12	Health/Low-Fat Cooking
IAB8-13	Italian Cuisine
IAB8-14	Japanese Cuisine
IAB8-15	Mexican Cuisine
IAB8-16	Vegan
IAB8-17	Vegetarian
IAB8-18	Wine
IAB9	**Hobbies & Interests**
IAB9-1	Art/Technology
IAB9-2	Arts & Crafts
IAB9-3	Beadwork
IAB9-4	Bird-Watching
IAB9-5	Board Games/Puzzles
IAB9-6	Candle & Soap Making
IAB9-7	Card Games
IAB9-8	Chess
IAB9-9	Cigars
IAB9-10	Collecting
IAB9-11	Comic Books
IAB9-12	Drawing/Sketching
IAB9-13	Freelance Writing
IAB9-14	Genealogy
IAB9-15	Getting Published
IAB9-16	Guitar
IAB9-17	Home Recording
IAB9-18	Investors & Patents
IAB9-19	Jewelry Making
IAB9-20	Magic & Illusion
IAB9-21	Needlework
IAB9-22	Painting
IAB9-23	Photography
IAB9-24	Radio
IAB9-25	Roleplaying Games

IAB9-26	Sci-Fi & Fantasy
IAB9-27	Scrapbooking
IAB9-28	Screenwriting
IAB9-29	Stamps & Coins
IAB9-30	Video & Computer Games
IAB9-31	Woodworking
IAB10	**Home & Garden**
IAB10-1	Appliances
IAB10-2	Entertaining
IAB10-3	Environmental Safety
IAB10-4	Gardening
IAB10-5	Home Repair
IAB10-6	Home Theater
IAB10-7	Interior Decorating
IAB10-8	Landscaping
IAB10-9	Remodeling & Construction
IAB11	**Law, Government, & Politics**
IAB11-1	Immigration
IAB11-2	Legal Issues
IAB11-3	U.S. Government Resources
IAB11-4	Politics
IAB11-5	Commentary
IAB12	**News**
IAB12-1	International News
IAB12-2	National News
IAB12-3	Local News
IAB13	**Personal Finance**
IAB13-1	Beginning Investing
IAB13-2	Credit/Debt & Loans
IAB13-3	Financial News
IAB13-4	Financial Planning
IAB13-5	Hedge Fund
IAB13-6	Insurance
IAB13-7	Investing
IAB13-8	Mutual Funds
IAB13-9	Options
IAB13-10	Retirement Planning

IAB13-11	Stocks
IAB13-12	Tax Planning
IAB14	**Society**
IAB14-1	Dating
IAB14-2	Divorce Support
IAB14-3	Gay Life
IAB14-4	Marriage
IAB14-5	Senior Living
IAB14-6	Teens
IAB14-7	Weddings
IAB14-8	Ethnic Specific
IAB15	**Science**
IAB15-1	Astrology
IAB15-2	Biology
IAB15-3	Chemistry
IAB15-4	Geology
IAB15-5	Paranormal Phenomena
IAB15-6	Physics
IAB15-7	Space/Astronomy
IAB15-8	Geography
IAB15-9	Botany
IAB15-10	Weather
IAB16	**Pets**
IAB16-1	Aquariums
IAB16-2	Birds
IAB16-3	Cats
IAB16-4	Dogs
IAB16-5	Large Animals
IAB16-6	Reptiles
IAB16-7	Veterinary Medicine
IAB17	**Sports**
IAB17-1	Auto Racing
IAB17-2	Baseball
IAB17-3	Bicycling
IAB17-4	Bodybuilding
IAB17-5	Boxing
IAB17-6	Canoeing/Kayaking

IAB17-7	Cheerleading
IAB17-8	Climbing
IAB17-9	Cricket
IAB17-10	Figure Skating
IAB17-11	Fly Fishing
IAB17-12	Football
IAB17-13	Freshwater Fishing
IAB17-14	Game & Fish
IAB17-15	Golf
IAB17-16	Horse Racing
IAB17-17	Horses
IAB17-18	Hunting/Shooting
IAB17-19	Inline Skating
IAB17-20	Martial Arts
IAB17-21	Mountain Biking
IAB17-22	NASCAR Racing
IAB17-23	Olympics
IAB17-24	Paintball
IAB17-25	Power & Motorcycles
IAB17-26	Pro Basketball
IAB17-27	Pro Ice Hockey
IAB17-28	Rodeo
IAB17-29	Rugby
IAB17-30	Running/Jogging
IAB17-31	Sailing
IAB17-32	Saltwater Fishing
IAB17-33	Scuba Diving
IAB17-34	Skateboarding
IAB17-35	Skiing
IAB17-36	Snowboarding
IAB17-37	Surfing/Body-Boarding
IAB17-38	Swimming
IAB17-39	Table Tennis/Ping-Pong
IAB17-40	Tennis
IAB17-41	Volleyball
IAB17-42	Walking
IAB17-43	Waterski/Wakeboard

IAB17-44	World Soccer
IAB18	**Style & Fashion**
IAB18-1	Beauty
IAB18-2	Body Art
IAB18-3	Fashion
IAB18-4	Jewelry
IAB18-5	Clothing
IAB18-6	Accessories
IAB19	**Technology & Computing**
IAB19-1	3-D Graphics
IAB19-2	Animation
IAB19-3	Antivirus Software
IAB19-4	C/C++
IAB19-5	Cameras & Camcorders
IAB19-6	Cell Phones
IAB19-7	Computer Certification
IAB19-8	Computer Networking
IAB19-9	Computer Peripherals
IAB19-10	Computer Reviews
IAB19-11	Data Centers
IAB19-12	Databases
IAB19-13	Desktop Publishing
IAB19-14	Desktop Video
IAB19-15	Email
IAB19-16	Graphics Software
IAB19-17	Home Video/DVD
IAB19-18	Internet Technology
IAB19-19	Java
IAB19-20	JavaScript
IAB19-21	Mac Support
IAB19-22	MP3/MIDI
IAB19-23	Net Conferencing
IAB19-24	Net for Beginners
IAB19-25	Network Security
IAB19-26	Palmtops/PDAs
IAB19-27	PC Support
IAB19-28	Portable

IAB19-29	Entertainment
IAB19-30	Shareware/Freeware
IAB19-31	Unix
IAB19-32	Visual Basic
IAB19-33	Web Clip Art
IAB19-34	Web Design/HTML
IAB19-35	Web Search
IAB19-36	Windows
IAB20	**Travel**
IAB20-1	Adventure Travel
IAB20-2	Africa
IAB20-3	Air Travel
IAB20-4	Australia & New Zealand
IAB20-5	Bed & Breakfasts
IAB20-6	Budget Travel
IAB20-7	Business Travel
IAB20-8	By US Locale
IAB20-9	Camping
IAB20-10	Canada
IAB20-11	Caribbean
IAB20-12	Cruises
IAB20-13	Eastern Europe
IAB20-14	Europe
IAB20-15	France
IAB20-16	Greece
IAB20-17	Honeymoons/Getaways
IAB20-18	Hotels
IAB20-19	Italy
IAB20-20	Japan
IAB20-21	Mexico & Central America
IAB20-22	National Parks
IAB20-23	South America
IAB20-24	Spas
IAB20-25	Theme Parks
IAB20-26	Traveling with Kids
IAB20-27	United Kingdom
IAB21	**Real Estate**

IAB21-1	Apartments
IAB21-2	Architects
IAB21-3	Buying/Selling Homes
IAB22	**Shopping**
IAB22-1	Contests & Freebies
IAB22-2	Couponing
IAB22-3	Comparison
IAB22-4	Engines
IAB23	**Religion & Spirituality**
IAB23-1	Alternative Religions
IAB23-2	Atheism/Agnosticism
IAB23-3	Buddhism
IAB23-4	Catholicism
IAB23-5	Christianity
IAB23-6	Hinduism
IAB23-7	Islam
IAB23-8	Judaism
IAB23-9	Latter-Day Saints
IAB23-10	Pagan/Wiccan
IAB24	**Uncategorized**
IAB25	**Non-Standard Content**
IAB25-1	Unmoderated UGC
IAB25-2	Extreme Graphic/Explicit Violence
IAB25-3	Pornography
IAB25-4	Profane Content
IAB25-5	Hate Content
IAB25-6	Under Construction
IAB25-7	Incentivized
IAB26	**Illegal Content**
IAB26-1	Illegal Content
IAB26-2	Warez
IAB26-3	Spyware/Malware
IAB26-4	Copyright Infringement

5.2 Banner Ad Types

The following table indicates the types of ads that can be accepted by the exchange unless restricted by publisher site settings.

Value	Description
1	XHTML Text Ad (usually mobile)
2	XHTML Banner Ad. (usually mobile)
3	JavaScript Ad; must be valid XHTML (i.e., Script Tags Included)
4	iframe

5.3 Creative Attributes

The following table specifies a standard list of creative attributes that can describe an ad being served or serve as restrictions of thereof.

Value	Description
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 (e.g., Over, Under, or Upon Exit)
9	Provocative or Suggestive Imagery
10	Shaky, Flashing, Flickering, Extreme Animation, Smileys
11	Surveys
12	Text Only
13	User Interactive (e.g., Embedded Games)
14	Windows Dialog or Alert Style
15	Has Audio On/Off Button
16	Ad Can be Skipped (e.g., Skip Button on Pre-Roll Video)

5.4 Ad Position

The following table specifies the position of the ad as a relative measure of visibility or prominence.

This OpenRTB table has values derived from the IAB Quality Assurance Guidelines (QAG). Practitioners should keep in sync with updates to the QAG values as published on IAB.net. Values "3" – "6" apply to apps per the mobile addendum to QAG version 1.5.

Value	Description
0	Unknown
1	Above the Fold
2	DEPRECATED - May or may not be initially visible depending on screen size/resolution.
3	Below the Fold
4	Header
5	Footer
6	Sidebar
7	Full Screen

5.5 Expandable Direction

The following table lists the directions in which an expandable ad may expand, given the positioning of the ad unit on the page and constraints imposed by the content.

Value	Description
1	Left
2	Right
3	Up
4	Down
5	Full Screen

5.6 API Frameworks

The following table is a list of API frameworks supported by the publisher. Note that MRAID-1 is a subset of MRAID-2. In OpenRTB 2.1 and prior, value "3" was "MRAID". However, not all MRAID capable APIs understand MRAID-2 features and as such the only safe interpretation of value "3" is MRAID-1. In OpenRTB 2.2, this was made explicit and MRAID-2 has been added as value "5".

Value	Description
1	VPAID 1.0
2	VPAID 2.0
3	MRAID-1
4	ORMMA
5	MRAID-2

5.7 Video Linearity

The following table indicates the options for video linearity. "In-stream" or "linear" video refers to pre-roll, post-roll, or mid-roll video ads where the user is forced to watch ad in order to see the video content. "Overlay" or "non-linear" refer to ads that are shown on top of the video content.

This field is optional. The following is the interpretation of the bidder based upon the presence or absence of the field in the bid request:

- If no value is set, any ad (linear or not) can be present in the response.
- If a value is set, only ads of the corresponding type can be present in the response.

Note to the reader: This OpenRTB table has values derived from the IAB Quality Assurance Guidelines (QAG). Practitioners should keep in sync with updates to the QAG values as published on IAB.net.

Value	Description
1	Linear / In-Stream
2	Non-Linear / Overlay

5.8 Video Bid Response Protocols

The following table lists the options for video bid response protocols that could be supported by an exchange.

Value	Description
1	VAST 1.0
2	VAST 2.0
3	VAST 3.0
4	VAST 1.0 Wrapper
5	VAST 2.0 Wrapper
6	VAST 3.0 Wrapper

5.9 Video Playback Methods

The following table lists the various video playback methods.

Value	Description
1	Auto-Play Sound On
2	Auto-Play Sound Off
3	Click-to-Play
4	Mouse-Over

5.10 Video Start Delay

The following table lists the various options for the video start delay. If the start delay value is greater than 0, then the position is mid-roll and the value indicates the start delay.

Value	Description
> 0	Mid-Roll (value indicates start delay in second)
0	Pre-Roll
-1	Generic Mid-Roll
-2	Generic Post-Roll

5.11 Video Quality

The following table lists the options for the video quality. These values are defined by the IAB – http://www.iab.net/media/file/long-form-video-final.pdf.

Value	Description
0	Unknown
1	Professionally Produced
2	Prosumer
3	User Generated (UGC)

5.12 VAST Companion Types

The following table lists the options to indicate markup types allowed for video companion ads. This table is derived from IAB VAST 2.0+. Refer to www.iab.net/vast/ for more information.

Value	Description
1	Static Resource
2	HTML Resource
3	iframe Resource

5.13 Content Delivery Methods

The following table lists the various options for the delivery of video content.

Value	Description
1	Streaming
2	Progressive

5.14 Content Context

The following table lists the various options for indicating the type of content in which the impression will appear.

This OpenRTB table has values derived from the IAB Quality Assurance Guidelines (QAG). Practitioners should keep in sync with updates to the QAG values as published on IAB.net.

Value	Description
1	Video (a video file or stream that is being watched by the user, including (Internet) television broadcasts)
2	Game (an interactive software game that is being played by the user)
3	Music (an audio file or stream that is being listened to by the user, including (Internet) radio broadcasts)
4	Application (an interactive software application that is being used by the user)
5	Text (a document that is primarily textual in nature that is being read or viewed by the user, including web page, eBook, or news article)
6	Other (content type unknown or the user is consuming content which does not fit into one of the categories above)
7	Unknown

5.15 QAG Media Ratings

The following table lists the media ratings used in describing content based on the QAG categorization. Refer to http://www.iab.net/ne_guidelines for more information.

Value	Description
1	All Audiences
2	Everyone Over 12
3	Mature Audiences

5.16 Location Type

The following table lists the options to indicate how the geographic information was determined.

Value	Description
1	GPS/Location Services
2	IP Address
3	User provided (e.g., registration data)

5.17 Device Type

The following table lists the type of device from which the impression originated.

OpenRTB version 2.2 of the specification added distinct values for Mobile and Tablet. It is recommended that any bidder adding support for 2.2 treat a value of 1 as an acceptable alias of 4 & 5.

This OpenRTB table has values derived from the IAB Quality Assurance Guidelines (QAG). Practitioners should keep in sync with updates to the QAG values as published on IAB.net.

Value	Description	Notes
1	Mobile/Tablet	Version 2.0
2	Personal Computer	Version 2.0
3	Connected TV	Version 2.0
4	Phone	New for Version 2.2
5	Tablet	New for Version 2.2
6	Connected Device	New for Version 2.2
7	Set Top Box	New for Version 2.2

5.18 Connection Type

The following table lists the various options for the type of device connectivity.

Value	Description
0	Unknown
1	Ethernet
2	WIFI
3	Cellular Network – Unknown Generation
4	Cellular Network – 2G
5	Cellular Network – 3G
6	Cellular Network – 4G

5.19 No-Bid Reason Codes

The following table lists the options for a bidder to signal the exchange as to why it did not offer a bid for the impression.

Value	Description
0	Unknown Error
1	Technical Error
2	Invalid Request

3	Known Web Spider
4	Suspected Non-Human Traffic
5	Cloud, Data center, or Proxy IP
6	Unsupported Device
7	Blocked Publisher or Site
8	Unmatched User

6. Bid Request/Response Samples

6.1 Github Repository

The official OpenRTB Github repository now contains a set of validated example requests. This repository should be considered the canonical examples for implementers.

https://github.com/openrtb/examples

6.2 Validator

A public validator of request and response objects is available at the link below. The validator is a developer utility for ensuring the validity of bid request and response payloads expressed in JSON. The validator can be used with all final versions of OpenRTB specification. To use the validator, select the version of OpenRTB that is relevant to your data sample, select whether that data sample is a Bid Request or a Bid Response, then paste the JSON data sample into the field indicated, and press the Validate button.

http://www.nexage.com/resources/technical-resources/openrtb-validator

The underlying code for the actual validator is also available. It is distributed freely under a BSD-3 open source license at the below URL.

http://github.com/openrtb/openrtb2x/tree/2.0/openrtb-validator

6.3 Bid Requests

6.3.1 Example 1 – Simple Banner

Following is a basic example of a bid request for a banner ad. Some optional parameters are included in this example.

```
{
    "id": "80ce30c53c16e6ede735f123ef6e32361bfc7b22",
    "at": 1, "cur": [ "USD" ],
    "imp": [
        {
            "id": "1", "bidfloor": 0.03,
            "banner": {
                "h": 250, "w": 300, "pos": 0
            }
        }
    ],
    "site": {
        "id": "102855",
        "cat": [ "IAB3-1" ],
        "domain": "www.foobar.com",
        "page": "http://www.foobar.com/1234.html ",
        "publisher": {
```

```
            "id": "8953", "name": "foobar.com",
            "cat": [ "IAB3-1" ],
            "domain": "foobar.com"
        }
    },
    "device": {
        "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/537.13
(KHTML, like Gecko) Version/5.1.7 Safari/534.57.2",
        "ip": "123.145.167.10"
    },
    "user": {
        "id": "55816b39711f9b5acf3b90e313ed29e51665623f"
    }
}
```

6.3.2 Example 2 – Expandable Creative

This example builds the first and adds parameters to describe support for an expandable creative, and passes data about the user from “Data Provider 1”.

```
{
    "id": "123456789316e6ede735f123ef6e32361bfc7b22",
    "at": 2, "cur": [ "USD" ],
    "imp": [
        {
            "id": "1", "bidfloor": 0.03,
            "iframebuster": [ "vendor1.com", "vendor2.com" ],
            "banner": {
                "h": 250, "w": 300, "pos": 0,
                "battr": [ 13 ],
                "expdir": [ 2, 4 ]
            }
        }
    ],
    "site": {
        "id": "102855",
        "cat": [ "IAB3-1" ],
        "domain": "www.foobar.com",
        "page": "http://www.foobar.com/1234.html",
        "publisher": {
            "id": "8953", "name": "foobar.com",
            "cat": [ "IAB3-1" ],
            "domain": "foobar.com"
        }
    },
    "device": {
        "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10 6 8) AppleWebKit/537.13
(KHTML, like Gecko) Version/5.1.7 Safari/534.57.2",
        "ip": "123.145.167.10"
    },
    "user": {
        "id": "55816b39711f9b5acf3b90e313ed29e51665623f",
```

```
        "buyeruid": "5456787654678765678987654678987654",
        "data": [
            {
                "id": "6", "name": "Data Provider 1",
                "segment": [
                    {
                        "id": "12341318394918", "name": "auto intenders"
                    },
                    {
                        "id": "1234131839491234", "name": "auto enthusiasts"
                    },
                    {
                        "id": "23423424", "name": "data-provider1-age",
                        "value": "30-40"
                    }
                ]
            }
        ]
    }
}
```

6.3.3 Example 3 – Mobile

This example uses a device object to reflect a mobile device, and an app object to reflect a request from a mobile application.

```
{
    "id": "IxexyLDIIk",
    "at": 2,
    "bcat": [ "IAB25", "IAB7-39", "IAB8-18", "IAB8-5", "IAB9-9" ],
    "badv": [ "apple.com", "go-text.me", "heywire.com" ],
    "imp": [
        {
            "id": "1", "bidfloor": 0.5, "instl": 0,
            "tagid": "agltb3B1Yi1pbmNyDQsSBFNpdGUY7fD0FAw",
            "banner": {
                "w": 728, "h": 90, "pos": 1,
                "btype": [ 4 ],
                "battr": [ 14 ],
                "api": [ 3 ]
            }
        }
    ],
    "app": {
        "id": "agltb3B1Yi1pbmNyDAsSA0FwcBiJkfIUDA", "name": "Yahoo Weather",
        "cat": [ "IAB15", "IAB15-10" ],
        "ver": "1.0.2",
        "bundle": "com.yahoo.wxapp",
        "storeurl": "https://itunes.apple.com/id628677149",
        "publisher": {
            "id": "agltb3B1Yi1pbmNyDAsSA0FwcBiJkfTUCV", "name": "yahoo",
            "domain": "www.yahoo.com"
```

```
        }
    },
    "device": {
        "dnt": 0,
        "ua": "Mozilla/5.0 (iPhone; CPU iPhone OS 6_1 like Mac OS X)
AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3",
        "ip": "123.145.167.189",
        "ifa": "AA000DFE74168477C70D291f574D344790E0BB11",
        "carrier": "VERIZON",
        "language": "en",
        "make": "Apple", "model": "iPhone",
        "os": "iOS", "osv": "6.1",
        "js": 1,
        "connectiontype": 3,
        "devicetype": 1,
        "geo": {
            "lat": 35.012345, "lon": -115.12345,
            "country": "USA",
            "metro": "803",
            "region": "CA", "city": "Los Angeles", "zip": "90049"
        }
    },
    "user": {
        "id": "ffffffd5135596709273b3a1a07e466ea2bf4fff",
        "yob": 1984, "gender": "M"
    }
}
```

6.3.4 Example 4 – Video

The following example illustrates a bid request for a video impression with two companion ad slots (1 expandable). Additionally, the video content itself is described in the "content" object. A few notes about specific fields in the example:

- `protocol`: Only VAST 2.0 and 3.0 are allowed. Note that a wrapper response is not allowed in this example.
- `sequence`: It is not explicitly included so the default of “1” should be assumed.
- `battr`: User interactive and alert type ads (value “13” and “14”, respectively) are explicitly being blocked for both the video and its companions.
- `pos`: Indicates this opportunity is “above the fold”.
- `api`: Indicates that VPAID 1.0 containers are explicitly supported. As such, the mime types supported for VPAID are only “application/x-shockwave-flash” and “application/javascript”. Note that there is an implicit restriction as to which protocol is allowed in which mime type. JavaScript support was not specified until VPAID 2.0, while Flash supports both VPAID 1.0 and 2.0.
- `companiontype`: Indicates only static or HTML resources are allowed.

```
{
    "id": "1234567893",
    "at": 2, "tmax": 120,
```

```
"imp": [
    {
        "id": "1", "bidfloor": 0.03,
        "video": {
            "w": 640, "h": 480, "pos": 1,
            "startdelay": 0, "minduration": 5, "maxduration": 30,
            "maxextended": 30,
            "minbitrate": 300, "maxbitrate": 1500,
            "api": [ 1, 2 ],
            "protocols": [ 2, 3 ],
            "mimes": [
                "video/x-flv",
                "video/mp4",
                "application/x-shockwave-flash",
                "application/javascript"
            ],
            "linearity": 1,
            "boxingallowed": 1,
            "playbackmethod": [ 1, 3 ],
            "delivery": [ 2 ],
            "battr": [ 13, 14 ],
            "companionad": [
                {
                    "id": "1234567893-1",
                    "w": 300, "h": 250, "pos": 1,
                    "battr": [ 13, 14 ],
                    "expdir": [ 2, 4 ]
                },
                {
                    "id": "1234567893-2",
                    "w": 728, "h": 90, "pos": 1,
                    "battr": [ 13, 14 ]
                }
            ],
            "companiontype": [ 1, 2 ]
        }
    }
],
"site": {
    "id": "1345135123", "name": "Site ABCD",
    "domain": "siteabcd.com",
    "cat": [ "IAB2-1", "IAB2-2" ],
    "page": "http://siteabcd.com/page.htm",
    "ref": "http://referringsite.com/referringpage.htm",
    "privacypolicy": 1,
    "publisher": {
        "id": "pub12345", "name": "Publisher A"
    },
    "content": {
        "id": "1234567",
        "series": "All About Cars",
        "season": "2", "episode": 23, "title": "Car Show",
        "cat": [ "IAB2-2" ],
        "keywords": "keyword-a,keyword-b,keyword-c"
```

```
        }
    },
    "device": {
        "ip": "64.124.253.1",
        "ua": "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; en-US; rv:1.9.2.16)
Gecko/20110319 Firefox/3.6.16",
        "os": "OS X",
        "flashver": "10.1", "js": 1
    },
    "user": {
        "id": "456789876567897654678987656789",
        "buyeruid": "5456787654678765678987656789876543",
        "data": [
            {
                "id": "6", "name": "Data Provider 1",
                "segment": [
                    {
                        "id": "12341318394918", "name": "auto intenders"
                    },
                    {
                        "id": "1234131839491234", "name": "auto enthusiasts"
                    }
                ]
            }
        ]
    }
}
```

6.3.5 Example 5 – PMP with Direct Deal

Following is a basic example of a bid request for a banner ad with a direct deal. Some optional parameters are included in this example.

```
{
    "id": "80ce30c53c16e6ede735f123ef6e32361bfc7b22",
    "at": 1, "cur": [ "USD" ],
    "imp": [
        {
            "id": "1", "bidfloor": 0.03,
            "banner": {
                "h": 250, "w": 300, "pos": 0
            },
            "pmp": {
                "private_auction": 1,
                "deals": [
                    {
                        "id":"AB-Agency1-0001",
                        "at": 1, "bidfloor": 2.5,
                        "wseat": [ "Agency1" ]
                    },
                    {
                        "id":"XY-Agency2-0001",
```

```
                    "at": 2, "bidfloor": 2,
                    "wseat": [ "Agency2" ]
                }
            ]
        }
    }
  ],
  "site": {
    "id": "102855",
    "domain": "www.foobar.com",
    "cat": [ "IAB3-1" ],
    "page": "http://www.foobar.com/1234.html",
    "publisher": {
      "id": "8953", "name": "foobar.com",
      "cat": [ "IAB3-1" ],
      "domain": "foobar.com"
    }
  },
  "device": {
    "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/537.13
(KHTML, like Gecko) Version/5.1.7 Safari/534.57.2",
    "ip": "123.145.167.10"
  },
  "user": {
    "id": "55816b39711f9b5acf3b90e313ed29e51665623f"
  }
}
```

6.3.6 Example 6 – Native Ad

Following is a basic example of a bid request for a Native ad; similar otherwise to the simple banner example in Section 6.3.1. Notice the `request` attribute in the `Native` object contains an encoded string of a native ad request that conforms to the Native Specification, specifically version 1.0 as indicated by the `ver` attribute.

```
{
  "id": "80ce30c53c16e6ede735f123ef6e32361bfc7b22",
  "at": 1, "cur": [ "USD" ],
  "imp": [
    {
      "id": "1", "bidfloor": 0.03,
      "native": {
        "request": "...Native Spec request as an encoded string...",
        "ver": "1.0",
        "api": [ 3 ], "battr": [ 13, 14 ]
      }
    }
  ],
  "site": {
    "id": "102855",
    "cat": [ "IAB3-1" ],
```

```
        "domain": "www.foobar.com",
        "page": "http://www.foobar.com/1234.html ",
        "publisher": {
            "id": "8953", "name": "foobar.com",
            "cat": [ "IAB3-1" ],
            "domain": "foobar.com"
        }
    },
    "device": {
        "ua": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/537.13
(KHTML, like Gecko) Version/5.1.7 Safari/534.57.2",
        "ip": "123.145.167.10"
    },
    "user": {
        "id": "55816b39711f9b5acf3b90e313ed29e51665623f"
    }
}
```

6.4 Bid Responses

6.4.1 Example 1 – Ad Served on Win Notice

Following is an example of a bid response with the ad served on win notice. The bid for this impression is a $9.43 CPM.

```
{
    "id": "1234567890", "bidid": "abc1123", "cur": "USD",
    "seatbid": [
        {
            "seat": "512",
            "bid": [
                {
                    "id": "1", "impid": "102", "price": 9.43,
                    "nurl": "http://adserver.com/winnotice?impid=102",
                    "iurl": "http://adserver.com/pathtosampleimage",
                    "adomain": [ "advertiserdomain.com" ],
                    "cid": "campaign111",
                    "crid": "creative112",
                    "attr": [ 1, 2, 3, 4, 5, 6, 7, 12 ]
                }
            ]
        }
    ]
}
```

6.4.2 Example 2 – VAST XML Document Returned Inline

Following is an example of a bid response that returns the VAST document inline to be served. A few notes about specific fields in the example:

- The bid for this impression is a $3.00 CPM.
- Note that since there both a win notice URL and an inline VAST document in the `adm` attribute, which constitutes the ad markup. The win notice is still called, but if it were to return markup it would be ignored in favor of the contents of the `adm` attribute.

```
{
   "id": "123",
   "seatbid": [
      {
         "bid": [
            {
               "id": "12345", "impid": "2", "price": 3.00,
               "nurl": "http://example.com/winnoticeurl",
               "adm": "%3C%3Fxml%20version%3D%221.0%22%20encoding%3D%22utf-
8%22%3F%3E%0A%3CVAST%20version%3D%222.0%22%3E%0A%20%20%20%20%3CAd%20id%3D%2212345%22
%3E%0A%20%20%20%20%20%20%20%20%3CInLine%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%3C
AdSystem%20version%3D%221.0%22%3ESpotXchange%3C%2FAdSystem%3E%0A%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%3CAdTitle%3E%3C!%5BCDATA%5BSample%20VAST%5D%5D%3E%3C%2F
AdTitle%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CImpression%3Ehttp%3A
%2F%2Fsample.com%3C%2FImpression%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%3CDescription%3E%3C!%5BCDATA%5BA%20sample%20VAST%20feed%5D%5D%3E%3C%2FDescription
%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CCreatives%3E%0A%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CCreative%20sequence%3D%221%22%20i
d%3D%221%22%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%3CLinear%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%3CDuration%3E00%3A00%3A30%3C%2FDuration%3E%0A%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CTrackingEvents%
3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%3C%2FTrackingEvents%3E%20%20%20%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%3CVideoClicks%3E%0A%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CClickThro
ugh%3E%3C!%5BCDATA%5Bhttp%3A%2F%2Fsample.com%2Fopenrtbtest%5D%5D%3E%3C%2FClickThroug
h%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%3C%2FVideoClicks%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%3CMediaFiles%3E%0A%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3CMediaFile%20delivery
%3D%22progressive%22%20bitrate%3D%22256%22%20width%3D%22640%22%20height%3D%22480%22%
20type%3D%22video%2Fmp4%22%3E%3C!%5BCDATA%5Bhttp%3A%2F%2Fsample.com%2Fvideo.mp4%5D%5
D%3E%3C%2FMediaFile%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%3C%2FMediaFiles%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3C%2FLinear%3E%0A%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3C%2FCreative%3E%0A%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%3C%2FCreatives%3E%0A%20%20%20%20%20%20%20%20%3C%2FInLine%3E%0A%20%20%2
0%20%3C%2FAd%3E%0A%3C%2FVAST%3E"
            }
         ]
      }
   ]
}
```

6.4.3 Example 3 – Direct Deal Ad Served on Win Notice

Following is an example of a bid response with the ad served on win notice. The bid for this impression is a $5.00 CPM against a direct deal.

```
{
   "id": "1234567890", "bidid": "abc1123", "cur": "USD",
   "seatbid": [
      {
         "seat": "512",
         "bid": [
            {
               "id": "1", "impid": "102", "price": 5.00,
               "dealid": "ABC-1234-6789",
               "nurl": "http: //adserver.com/winnotice?impid=102",
               "adomain": [ "advertiserdomain.com" ],
               "iurl": "http: //adserver.com/pathtosampleimage",
               "cid": "campaign111",
               "crid": "creative112",
               "adid": "314",
               "attr": [ 1, 2, 3, 4 ]
            }
         ]
      }
   ]
}
```

6.4.4 Example 4 – Native Markup Returned Inline

Following is an example of a bid response that returns a native ad inline to be served. The `adm` attribute contains an encoded string of a native ad request that conforms to the Native Specification and specifically the same version as that used for the request string. Alternatively, the `adm` attribute could have been omitted in favor of returning the native ad markup in the response to the win notice `nurl`.

```
{
   "id": "123",
   "seatbid": [
      {
         "bid": [
            {
               "id": "12345", "impid": "2", "price": 3.00,
               "nurl": "http://example.com/winnoticeurl",
               "adm": "...Native Spec response as an encoded string..."
            }
         ]
      }
   ]
}
```

7. Implementation Notes

The following section will provide brief notes on how certain objects and fields are to be interpreted and implemented.

7.1 COPPA Regulation Flag

The United States Federal Trade Commission has changed the compliance rules for the Children's Online Privacy Protection Act ("COPPA"), effective July 1, 2013. The proposal effects websites, and associated services), that have been identified as: (1) directed to users under 13 years of age; or (2) collecting information from users actually known to be under 13 (collectively "Children's Sites").

The FTC has written a comprehensive FAQ on the change here:

http://business.ftc.gov/documents/Complying-with-COPPA-Frequently-Asked-Questions

Steve Bellovin, CTO of the FTC, argued for a standardized signaling protocol in a blog posted dated January 2013:

http://techatftc.wordpress.com/2013/01/02/coppa-and-signaling/

Impacts

The FAQ specifically calls out these areas relevant for OpenRTB as "Personal Information" that is not to be collected:

- Geo-location information sufficient to identify street name and name of a city or town.
- Persistent identifiers when they can be used to recognize a user over time and across different Web sites or online services.

Recommendations to Implementers

OpenRTB Exchanges and Bidders should:

- Provide a facility for sites to be declared as "child directed".
- Implement the regulations object extension.
- Provide facilities within campaigns to target for and against this signal.
- Degrade the Geographic information to be less exact prior to logging or transmission.
- Suppress the assignment and synchronization of identifiers, depending on usage.

It is recommended that when `regs.coppa` = 1, the exchange should additionally manipulate the OpenRTB bid request object as follows:

Device Object

- Suppress `didmd5` and `didsha1` device ID fields.
- Truncate `ip` field - remove lowest 8 bits.
- Truncate `ipv6` field - remove lowest 32 bits.

Geo Object

- Suppress `lat` and `lon` fields.
- Suppress `metro`, `city`, and `zip` fields.

User Object

- Suppress `id`, `buyerid`, `yob`, and `gender` fields.

7.2 PMP & Direct Deals

Best Practice Bidding Logic

```
Receive request and parse;
Create empty bid list for response;

If request contains the impression[].pmp object;
   match bids against each pmp.deals[];
   enforce targeting for dealID and seatID;
   append best M matching bids to response;

If pmp.private_auction = False;
   match open auction bids against the request;
   append top N bids by price to response;

Return response list to exchange;
```

Recommendations

- M >= 1, preferably one per matching Deal ID.
- N >= 2 to assist with blocking rate issues.
- Minimum viable is “1+1” bidding.
- Ideal is “M+N” bidding.

Warning

Returning only one bid when both Deal ID and open auction bids are valid creates problems. The exchange side may be configured by a publisher to prioritize all Deal ID bids above open auction bids, or to force a price auction between them with different floors by class of bid. There are multiple common practices that depend on how the publisher prefers to sell inventory with Deal ID.

Policy Recommendations

- A Deal ID should be utilized for any situation where the auction may be awarded to a bid not on the basis of price alone. Any prioritization of bids other than by price should have a Deal ID.
- A Deal ID is recommended for all situations where a preferential floor may be assigned to a seat entity.

Anti-Patterns

The below is a set of anti-patterns that OpenRTB supporting platforms have observed in various attempts to implement Deal ID bidding logic.

Subjecting Deal ID Bids to an internal auction on price

The ideal bidding logic describes a process of being liberal about sending bids. Deal ID bids may not be subject to a classic price auction. There may be an expectation that the buyer and seller want prioritization to achieve a larger objective: complete delivery of the Deal represented by the Deal ID. Thus any bidding logic that sorts Deal ID bids by price (with or without open marketplace bids) and truncates the list too aggressively can endanger the fulfillment of the Deal.

Associating Deal ID to the wrong Object

A Deal ID should be treated as a "targeting token" associated to orders, line-items or campaigns. If the Deal ID is associated to a Seat/Buyer it may create an undesired application of the Deal ID too many active campaigns. Alternatively if it is associated to the Advertiser it may limit that entity to only a single Deal ID.

Improper Handling of the Private vs Open Market Flag

The `pmp.private_auction` flag indicates that the seller is willing or not willing to accept open market bids (i.e., "all bidders are welcome"). If this flag is not read and interpreted correctly, bid responses may be invalid. Open market bids sent to a private impression auction may be rejected and should not have been exposed to all bidders.

Improper handling of Seat IDs

If Seat IDs are treated as a filter of eligible demand partners on an open market impression, this defeats the "all bidders are welcome" intention.

Silently Applying Margin Discounts to Deal ID Bids

With Deal ID buyers are sellers are communicating directly. The Exchange and Bidder become third-party automation platforms. If there are any automatic or silent discounts of bid prices (based upon margins or fees) set by either the exchange or the bidder, then the Deal may fail to function correctly.

Use cases

Case-1: Open Trading Agreement with Buyer

- Between publisher and buying entity.
- Publisher sets an access rule defining the price floor for a specific buyer.
- Locked to the buyer.
- Broadcast price floor.
- Public/open inventory.
- No Deal ID needed (Deal ID is optional).
- No named advertiser(s).

- No prioritization of bids.
- Daily total or frequency caps optional on publisher/exchange side.
- All placements or limited to specific placements.
- Targeting is up to the buyer/bidder.

Case-2: Open Trading Agreement with Buyer with Named Advertisers

- As Case-1 with a list of named advertisers.

Case-3: Open Bidding with Deal ID as Value-added Markers

- Between publisher and buying entity.
- Publisher sets a price floor for URL masked inventory.
- Public/open inventory (i.e., all buyers welcome).
- Deal ID represents “Package Tokens”.
- Each Deal ID signals that the impression falls into various content and placement categories.
- Floor is associated to each Deal ID to signal cost for usage of that token.
- Winner is decided by bid price.
- Execution of targeting is up to the buyer/bidder.

Case-4: First Look Trading Agreement

- Between publisher and buying entity.
- Publisher sets an access rule defining the price floor for the buyer.
- Locked to the buyer.
- Known price floor.
- Deal ID needed.
- Optional named advertiser list.
- Prioritization of bids expected.
- Daily total or frequency caps optional on publisher/exchange side.
- All placements or limited to specific placements.
- Targeting is up to the buyer/bidder.

Case-5: Direct Option Deal with Advertiser via RTB

- Between Publisher and Advertiser or their representative.
- Publisher sets a rule defining a price floor and prioritization for specific advertiser(s).
- Fill rate is expected to be greater than or equal to X%.
- Locked to the buyer.
- Private/exclusive inventory.
- Limited to a set list of advertiser names (generally variants of one name).
- Known price floor.
- Deal ID needed.
- Prioritization of bids expected.
- Daily total or frequency caps will apply on bidder side; optional on Exchange side.
- Limited to specific placements.
- Targeting is mostly enforced by buyer/bidder.

Case-6: Direct Option Deal with Advertiser via RTB with Private Data

- Same as Case-4.

- Deal ID represents some combination of private first-party data from the Publisher.

Case-7: Full-Fill Direct Deal with Advertiser via RTB

- Same as Case-4.
- Fill rate is expected to be 100% or nearly so.

Case-8: Full-Fill Direct Deal with Advertiser via RTB with Private Data

- Same as Case-6.
- Deal ID represents some combination of private first-party data from the Publisher.

7.3 No-Bid Signaling

This section covers best practices for using the optional no-bid signaling. See the List 5.19 for the enumerated list of no-bid reason codes.

Many exchanges support multiple response types as a no-bid:

- HTTP 204 "No Content" from the bidder *(most economical in terms of bandwidth)*.
- An empty JSON object:
  ```
  {}
  ```
- A well-formed no bid response:
  ```
  {"id": "1234567890", "seatbid": []}
  ```
- A well-formed no bid response with a reason code:
  ```
  {"id": "1234567890", "seatbid": [], "nbr": 2}
  ```

An important issue in RTB is when impressions are triggered by software robots mimicking web browsers. Such robots may be implicitly or explicitly driving these false transactions. The following represents a set of symmetric best practices for exchanges and bidders to help recognize and reject these events.

Responsibility of the exchange

Make best effort to classify and reject "non-human traffic" requests for ads to the exchange via the following best practices:

- (Recommended) Filter impressions from known spiders via user-agent classification.
- (Recommended) Filter impressions from suspected NHT via a "detector".

Responsibility of the bidder

- (Recommended) no-bid impressions from known spiders via user-agent classification.
- (Recommended) no-bid impressions from suspected NHT via a "detector".
- Specify a no-bid reason code in either case.

Where:

- For exchanges, filtering the impression means that the exchange should respond to the "ad call" with either a blank HTTP 204 response or an unpaid ad (PSA) and not offered to any bidders.
- For bidders, filtering the impression means that the bidder should respond with a no-bid.

- For both exchanges and bidders, the impression transaction records should be clearly marked in any logging systems and be removed from contributing to any event counts associated with planning, forecasting, and reporting systems.

Appendix A. Additional Information

- Creative Commons / Attribution License
  http://creativecommons.org/licenses/by/3.0
- IAB (Interactive Advertising Bureau)
  http://www.iab.net
- IAB Quality Assurance Guidelines (QAG):
  http://www.iab.net/ne_guidelines
- JavaScript Object Notation (JSON)
  http://www.json.org
- MMA (Mobile Marketing Association)
  http://mmaglobal.com
- OpenRTB Project on Github
  http://github.com/openrtb/OpenRTB/
- Apache Avro
  http://avro.apache.org
- Protocol Buffers (Protobuf)
  http://code.google.com/p/protobuf
- Google Metro Codes
  http://code.google.com/apis/adwords/docs/appendix/metrocodes.html
- U.N. Code for Trade and Transport Locations:
  http://www.unece.org/cefact/locode/service/location.htm

Appendix B. Specification Change Log

This appendix serves as an index to specification changes made in Version 2.3 and 2.3.1 relative to Version 2.2. These changes pertain only to the substance of the specification and not routine document formatting, organization, or content without technical impact.

Ref.	Sections(s)	Description
1	3.2.2, 3.2.5	Native Ad Units: Added new `Native` object under `Imp`.
2	3.2.1	Test Mode: Added `BidRequest.test`.
3	3.2.6	Mobile Web Signal: Added `site.mobile`.
4	3.2.11	Additional Device Attributes: Added `h`, `w`, `pxratio`, `ppi`, and `hwv` to `Device`.
5	3.2.11	DNT vs. Limit Ad Tracking: Added `device.lmt`.
6	3.2.12	UTC Offset: Added `geo.utcoffset`.
7	4.2.3	Block Check Support in Bids: Added `bid.cat` and `bid.bundle`.
8	3.2.6, 3.2.7, 3.2.9, 3.2.13	Fix Keywords: Reverted to original string type from the dual "string or string array" description.
9	3.2.9	Fix Content Context: Corrected enumerated type from string to integer.
10	3.2.11	Fix Device Carrier: Clarified description of usage.
11	3.2.13	Update from version 2.3: In the user object, the buyer ID attribute has been corrected to `buyeruid`.
12	4.4	Update from version 2.3: The `${AUCTION_BID_ID}` has been corrected to be substituted with the `BidResponse.bidid` attribute.