Mobile RTB Interface

Specification Version 1.0

OpenRTB Mobile

February 21, 2011

License

Version Designations

This specification is version controlled using “X.Y” notation. The “X” field represents a major version. Changes thereto indicate substantial and possibly disruptive changes or additions to technical content. The “Y” field indicates minor changes or enhancements to technical content that will always be backward compatible to previous versions within the same major version. An “RC” suffix is a temporary indicator of a document being a *Release Candidate* of the version specified.

Table of Contents

1. Introduction 1

1.1 OpenRTB Mobile 1
1.2 Terminology 1
1.3 Document Organization 1

2. Real-Time Bidding Interface 2

2.1 Layer-1 Transport 3
2.2 Layer-2 Representation 3
2.3 Layer-3 Protocol 3
2.3.1 Bid Request 4
2.3.2 Bid Response 4
2.3.2.1 Ad Served in the Bid 5
2.3.2.2 Ad Served on the Win Notice 6
2.3.2.3 Comparison of Ad Serving Methods 6
2.3.3 Substitution Macros 6
2.4 Layer-4 Attribution 7
2.4.1 Bid Request 8
2.4.2 Bid Response 10
2.5 Layer-5 Extension 11

3. Reference Lists 12

3.1 Content Categories 12
3.2 Ad Types 12
3.3 Creative Attributes 13
3.4 Ad Position 13
3.5 Price Units 13
3.6 No-Bid Reason 14

Appendix A. Additional Information 15

1. Introduction

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 between suppliers of mobile publisher inventory (i.e., exchanges) and competitive buyers of that mobile inventory (i.e., bidders).

1.1 OpenRTB Mobile

While there are many similarities between online and mobile advertising, there are also differences. For this reason, the Mobile Subcommittee of OpenRTB has been formed and this specification is a product thereof.

The Mobile Subcommittee also known as OpenRTB Mobile comprises representatives from both the demand and supply sides, each including both pure-play mobile as well as those also engaged in online to achieve a healthy diversity.

1.2 Terminology

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

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.

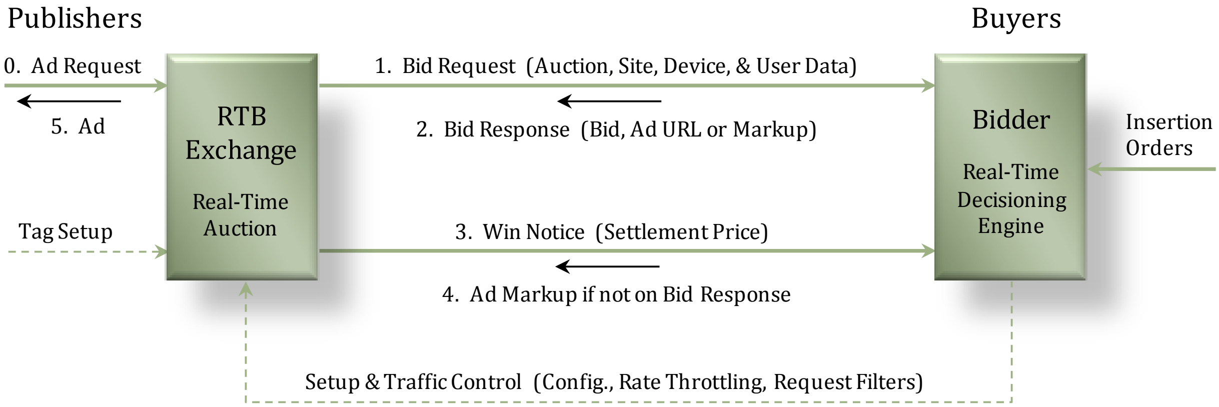
Site: Ad supported content including web and applications unless otherwise specified.

1.3 Document Organization

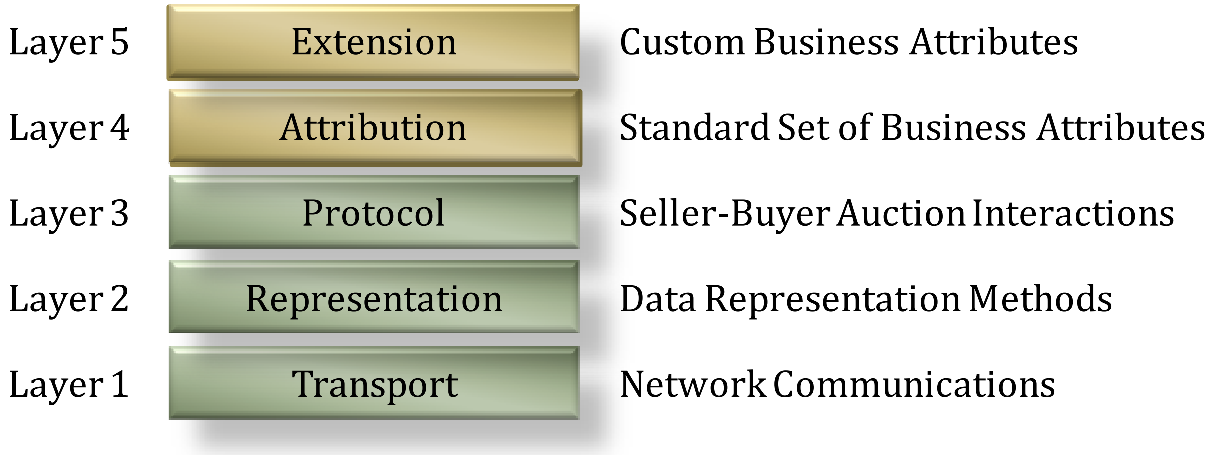
The remainder of this document is organized as follows. Chapter 2 presents the actual RTB Interface specification. Chapter 3 defines several attribute value lists referenced in the foregoing. The document concludes with reference information provided in the Appendix.

2. Real-Time Bidding Interface

The following figure illustrates the RTB interactions between an exchange and its bidders. Ad requests originate at publisher sites; both mobile web and mobile applications. In the case of mobile web, the vast majority of requests originate from publishers' servers rather than browsers. 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. 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.


Publishers
Buyers
0. Ad Request
5. Ad
Tag Setup
RTB Exchange
Real-Time Auction
1. Bid Request (Auction, Site, Device, & User Data)
2. Bid Response (Bid, Ad URL or Markup)
3. Win Notice (Settlement Price)
4. Ad Markup if not on Bid Response
Bidder
Real-Time Decisioning Engine
Insertion Orders
Setup & Traffic Control (Config., Rate Throttling, Request Filters)

The RTB Interface is specified in discrete layers to manage its potential complexity. The Transport layer specifies the raw communications medium between parties. The Representation layer speaks to the formatting or encoding of data being communicated. The Protocol layer defines the essential auction conversation between seller and buyers. The Attribution layer specifies standard business data for making buying decisions while the Extension layer allows custom attributes to be specified. The first three layers can be considered the technical enablers, while the top two are the business payload.


Layer 5
Extension
Custom Business Attributes
Layer 4
Attribution
Standard Set of Business Attributes
Layer 3
Protocol
Seller-Buyer Auction Interactions
Layer 2
Representation
Data Representation Methods
Layer 1
Transport
Network Communications

2.1 Layer-1 Transport

The base protocol between an exchange and its bidder 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. No custom headers are required by this specification and all calls should return HTTP code 200.

SSL (Secure Sockets Layer) manifesting itself as HTTPS 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.

ⓘ **BEST PRACTICE:** *One of the simplest and most effective ways of improving connection performance is to enable HTTP Persistent Connections, also known as HTTP 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 Layer-2 Representation

This layer refers to the formatting of bid request and bid response data payloads. JSON (JavaScript Object Notation) is the required format, chosen for its combination of human readability and compactness. The higher layers of this specification present the data syntax of the JSON representation.

In addition, an exchange can offer multiple optional representations to bidders who may prefer them. These might include a compressed form of JSON, XML, protocol buffers by Google, Thrift being developed as an Apache Software Foundation project, and many others.

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

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.

2.3 Layer-3 Protocol

This layer defines the actual conversation between an exchange and its bidders during an impression auction. It specifies the essential contents of a bid request, bid responses, how the winning bidder is notified of the win and settlement price (i.e., in auction types other than 1st Price, the winner’s bid is not necessarily the price they pay), and options on how the ad markup is served.

2.3.1 Bid Request

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 "imp" (i.e., impression) object. Other attributes are optional since an exchange may establish default values.

The "imp" object describes the ad position 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 as a bundle. Each "imp" object has a required ID so that bids can reference them individually. An exchange can also conduct private auctions by restricting involvement to specific subsets of seats within bidders.

Top-Level Bid Request Object ▶ Required ▷ Optional

```
{
    "id": "string40",                 ▶  Unique ID of the bid request (i.e., the overall auction ID).
    "at": n,                          ▷  Auction type; "1" indicates 1st Price, others denote alternate rules.
    "tmax": n,                        ▷  Maximum time in mS that the exchange will wait for bid responses.

    "imp": [                          ▶  1 object per impression being offered for bid.
        {
            "impid": "string40",      ▶  Unique ID of the impression.
            "wseat": […],             ▷  Array of allowed bidder seats; default is unrestricted.
            "h": n,                   ▷  Height in pixels.
            "w": n,                   ▷  Width in pixels.
            "pos": n,                 ▷  Position of the impression on the page (see Section 3.4).
            "instl": n,               ▷  "1" if the ad is interstitial or full screen; else "0" (i.e., no).
            "btype": […],             ▷  Array of blocked ad types (see Section 3.2).
            "battr": […]              ▷  Array of blocked creative attributes (see Section 3.3).
        },

        …                             ▷  Additional "imp" objects (optional).
    ],

    …                                 ▷  Business objects defined in Layer-4.
}
```

ⓘ **BEST PRACTICE:** *Exchanges are highly encouraged to publish their auction rules to bidders for any auction types other than simple 1st Price. Similarly, the default maximum auction time should also be published if not included in bid requests.*

2.3.2 Bid Response

An empty bid response indicates no-bid. Otherwise, the top-level bid response object is defined below. 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.

A bid response can contain multiple "seatbid" objects, each on behalf of a different bidder seat. Since a bid request can include multiple impressions, each "seatbid" object can contain multiple bids each pertaining to a different impression. Thus, each “bid” object must include the impression ID to which it pertains as well as the bid price. 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 (i.e., all or nothing).

Top-Level Bid Response Object ▶ Required ▷ Optional

```
{
    “id”: “string40”,                ▶  ID of the bid request.
    “bidid”: “string40”,             ▷  Bid response ID to assist tracking for bidders.
    “nbr”: n,                        ▷  Reason for not bidding (see Section 3.6); default is “0” (unknown).
    “cur”: “string4”,                ▷  Bid currency using ISO-4217 alphabetic codes; default is “USD”.
    “units”: n,                      ▷  Bid units (see Section 3.5); default is “0” (CPM).

    “seatbid”: [                     ▷  0 or more sets of bids by seat.
        {
            “seat”: “string40”,      ▷  ID of the bidder seat on whose behalf this bid is made.
            “group”: n,              ▷  “1” means impressions must be won-lost as a group; default is “0”.
            “bid”: [                 ▶  0 or more bid objects; related to “imp” objects in the bid request.
                {
                    “impid”: “string40”,       ▶  ID of the impression to which this bid applies.
                    “price”: x,                ▶  Bid price in the currency and units specified.
                    “adid”: “string40”,        ▷  ID that references the ad to be served if the bid wins.
                    “nurl”: “string256”,       ▷  Win notice URL.
                    “adm”: “string1024”,       ▷  Actual XHTML ad markup.

                    …                          ▷  Business attributes defined in Layer-4.
                },

                …                    ▷  Additional “bid” objects (optional).
            ]
        },

        …                            ▷  Additional “bidset” objects (optional).
    ]
}
```

For each bid, the “nurl” attribute contains the win notice URL. If the bidder wins the impression, the exchange calls this notice URL a) to inform the bidder of the win and b) to convey certain information using substitution macros (see Section 2.3.3).

ⓘ **BEST PRACTICE:** *Substitution macros may allow a bidder to use a static notice URL for all of its bids. Thus, exchanges should offer the option of a default notice URL that can be pre-configured per bidder to reduce redundant data transfer.*

2.3.2.1 Ad Served in the Bid

There are two methods by which the winning bidder can return ad markup to the exchange. One method is to serve the ad markup in the bid request itself. This is accomplished via the “adm” attribute in the “bid” object. If both methods are used, this method will take precedence in that if the “adm” attribute is passed and has a non-empty value, its value will be accepted as the ad markup.

2.3.2.2 Ad Served on the Win Notice

The alternate method of returning ad markup to the exchange is via the win notice. In this case, the response body of the win notice call (e.g., invoking the “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 “adm” attribute in the “bid” object must be omitted.

2.3.2.3 Comparison of Ad Serving Methods

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 in the Bid

- ❑ *Potential Concurrency:* The exchange can choose to return that ad markup and call the win notice concurrently, thereby improving user experience.
- ❑ *Reduced Risk of Forfeiture:* A forfeit is the scenario in which a bidder wins, but forfeits due to technical failure serving the ad. This can occur when serving on the win notice (e.g., win notice call failure), but is mitigated by including the ad in the bid.

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 mount up over high volumes.
- ❑ *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 settlement price has been established.

2.3.3 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 settlement 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 "adm" attribute of the bid response. A use case for macros in the ad markup might be when a bidder prefers to receive its win notice 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 "id" attribute.
${AUCTION_BID_ID}	ID of the bid; from "bidid" attribute.
${AUCTION_IMP_ID}	ID of the impression just won; from "impid" attribute.
${AUCTION_SEAT_ID}	ID of the bidder's seat for whom the bid was made.
${AUCTION_AD_ID}	ID of the ad markup the bidder wishes to serve; from "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.
${AUCTION_UNITS}	The units 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, but must be mutually agreed upon between exchange and bidder. As an example, suppose that the price macro is to be obfuscated 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 unnecessary.*

2.4 Layer-4 Attribution

This layer adds business information to Layer-3 enabling effective buying decisions on the part of bidders and quality controls within an exchange. Unless otherwise specified, an omitted attribute generally indicates that its value is unknown. In this layer, all objects and attributes thereof are technically optional although the availability of these attributes will factor directly into the competitive value of a given exchange or bidder.

ⓘ **BEST PRACTICE:** *Exchanges and bidders are highly encouraged to share guidance with each other as to the availability and estimated frequency of attributes.*

2.4.1 Bid Request

The bid request as defined in Layer-3 contains a top-level object that includes the attributes required for auction mechanics as well as an array of “imp” (i.e., impression) objects on which to bid. This layer adds to this top level object a set of business objects that enable bidders to evaluate the value of these impressions. Unless otherwise, specified, a bid request can contain 0 or 1 of each of these objects.

Bid Request Object: “site” ▶ Required ▷ Optional

```
“site”: {
    “sid”: “string40”,             ▷  Site ID on the exchange.
    “name”: “string64”,            ▷  Site name; may be masked at publisher’s request.
    “domain”: “string64”,          ▷  Domain of the site (e.g., “foo.com”).
    “pid”: “string40”,             ▷  Publisher ID of the site.
    “pub”: “string64”,             ▷  Publisher name; may be masked at publisher’s request.
    “pdomain”: “string64”,         ▷  Domain of the publisher (e.g., “foopub.com”).
    “cat”: […],                    ▷  Array of content categories of the site or page (see Section 3.1).
    “keywords”: “string256”,       ▷  Comma separated list of keywords related to site content.
    “page”: “string256”,           ▷  URL of the current page.
    “ref”: “string256”,            ▷  Referrer URL that caused navigation to the current page.
    “search”: “string256”          ▷  Search string that caused navigation to the current page.
}
```

A “site” object should be included if the ad supported content is part of a mobile website (as opposed to an application). A bid request must not contain both a “site” object and an “app” object.

Bid Request Object: “app” ▶ Required ▷ Optional

```
“app”: {
    “aid”: “string40”,             ▷  Application ID on the exchange.
    “name”: “string64”,            ▷  Application name; may be masked at publisher’s request.
    “domain”: “string64”,          ▷  Domain of the application (e.g., “mygame.foo.com”).
    “pid”: “string40”,             ▷  Publisher ID of the application.
    “pub”: “string64”,             ▷  Publisher name; may be masked at publisher’s request.
    “pdomain”: “string64”,         ▷  Domain of the publisher (e.g., “foopub.com”).
    “cat”: […],                    ▷  Array of content categories of the application (see Section 3.1).
    “keywords”: “string256”,       ▷  Comma separated list of keywords related to application content.
    “ver”: “string16”,             ▷  Application version.
    “bundle”: “string64”,          ▷  Application bundle (e.g., com.foo.mygame).
    “paid”: n                      ▷  “1” if the application is a paid version; else “0” (i.e., free).
}
```

An “app” object should be included if the ad supported content is part of a mobile application (as opposed to a mobile website). A bid request must not contain both an “app” object and a “site” object.

Bid Request Object: “device” ▶ Required ▷ Optional

```
“device”: {
    “did”: “string42”,            ▷  SHA1 hashed device ID; IMEI when available, else MEID or ESN.
    “dpid”: “string42”,           ▷  SHA1 hashed platform-specific ID (e.g., Android ID or UDID for iOS).
    “ip”: “string15”,             ▷  IP address closest to device (typically a carrier IP).
    “country”: “string4”,         ▷  Country derived from the IP address using ISO-3166-1 Alpha-3.
    “carrier”: “string64”,        ▷  Carrier or ISP derived from the IP address.
    “ua”: “string256”,            ▷  Device user agent string.
    “make”: “string32”,           ▷  Device make (e.g., “Apple”).
    “model”: “string32”,          ▷  Device model (e.g., “iPhone”).
    “os”: “string32”,             ▷  Device operating system (e.g., “iOS”).
    “osv”: “string32”,            ▷  Device operating system version (e.g., “3.1.2”).
    “js”: n,                      ▷  “1” if the device supports JavaScript; else “0”.
    “loc”: “string16”             ▷  Lat/Long as “-999.99,-999.99” (i.e., south and west are negative).
}
```

The “device” object provides information pertaining to the mobile device including its hardware, platform, location, and carrier.

ⓘ **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.*

Bid Request Object: “user” ▶ Required ▷ Optional

```
“user”: {
    “uid”: “string40”,            ▷  Unique consumer ID of this user on the exchange.
    “yob”: n,                     ▷  Year of birth as a 4-digit integer.
    “gender”: “string1”,          ▷  Gender as “M” male, “F” female, “O” other.
    “zip”: “string16”,            ▷  Home zip code if USA; else postal code.
    “country”: “string4”,         ▷  Home country; using ISO-3166-1 Alpha-3.
    “keywords”: “string256”       ▷  Comma separated list of keywords of consumer interests or intent.
}
```

The “user” object contains information known or derived about the human user of the device. Note that the user ID is an exchange artifact (refer to the “device” object for hardware or platform derived IDs) and may be subject to rotation policies. However, this user ID must be stable long enough to serve reasonably as the basis for frequency capping.

Bid Request Object: “restrictions” ▶ Required ▷ Optional

```
“restrictions”: {
    “bcat”: […],               ▷  Array of blocked content categories (see Section 3.1).
    “badv”: […]                ▷  Array of blocked advertiser domains.
}
```

The “restrictions” object allows certain block lists to be passed on the bid request. This technique is useful in several cases including a) when bidders have not download these lists from the exchange offline, b) when block lists are highly dynamic, or c) when passing critical blocked items to ensure they are not missed in less frequent offline downloads. Per the latter case, if a block list is both passed in this object and synchronized offline, bidders must combine them via union (i.e., one does not supersede the other).

2.4.2 Bid Response

The bid response as defined in Layer-3 contains a top-level object that includes the attributes required for auction mechanics as well as an array of 0 or more “bid” objects (i.e., 0 or 1 “bid” object per “imp” object in the bid request). This layer defines an additional set of business attributes that can be added to each “bid” object for use by an exchange. These additional attributes are optional, but are highly useful for ensuring quality control with respect to ad delivery. The importance of this to an exchange may be such that it may place auction value on their inclusion such as favoring a bid that includes them when resolving winning tie bids.

Bid Response Object: “bid” ▶ Required ▷ Optional

```
“bid”: [
    {
        …                        ▶  Attributes defined in Layer-3.

        “adomain”: “string”,     ▷  Advertiser’s primary or top-level domain for advertiser checking.
        “iurl”: “string”,        ▷  Sample image URL (without cache busting) for content checking.
        “cid”: “string”,         ▷  Campaign ID or similar that appears within the ad markup.
        “crid”: “string”,        ▷  Creative ID for reporting content issues or defects.
        “attr”: […]              ▷  Array of creative attributes (see Section 3.3).
    },
    …                            ▷  Additional “bid” objects (optional).
]
```

The “adomain” attribute can be used to check advertiser block list compliance. The “iurl” attribute can provide a link to an image that is representative of the campaign’s content (irrespective of whether the campaign may have multiple creatives). This enables human review for spotting inappropriate content. The “cid” attribute can be used to block ads that were previously identified as inappropriate; essentially a safety net beyond the block lists. The “crid” attribute can be helpful in reporting creative issues back to bidders. Finally, the “attr” array indicates the creative attributes that describe the ad to be served.

2.5 Layer-5 Extension

Exchanges may offer additional attributes that bidders would have the option of consuming or ignoring. Any of the standard objects may include these extended attributes defined by an exchange. Extended objects may be included in addition to the standard objects. Similarly, an exchange may also define extended attributes or objects that it can consume on bid responses.

It is highly recommended that an Exchange distinguish in some way the names of extended attributes or objects from those defined as standard in Layers 3 and 4. This could be accomplished by prefixing names with an underscore or an abbreviated company name. For example, the Foo, Inc. exchange might name an attribute describing a user’s marital status as “_marital”, “fooMarital”, “foo_marital”, etc., as long as the convention a) is distinctive from the standard attributes and b) is used consistently.

Domain names are not recommended due to their potentially significant size impact over a large number of bid requests. Furthermore, there is little or no impact of name collisions across exchanges such that the additional size would be warranted. The convention is simply to assist buy-side integrators, who may be dealing with multiple exchanges, to easily distinguishing standard vs. non-standard entities.

Notwithstanding the ability to extend this specification, standard objects and attributes must be used if possible. For example, including the user’s gender is optional. But if passed, it must be expressed using the standard “gender” attribute as defined. Omitting this attribute as optional and creating an extended attribute to convey the user’s gender would not be considered compliant.

3. Reference Lists

This chapter presents lists that are referenced in this specification as legal values for specific attributes.

3.1 Content Categories

The following table details the content categories used to describe site, application, and advertising content. These codes and descriptions are derived from the IAB and have been previously adopted by OpenRTB.

VALUE	DESCRIPTION	VALUE	DESCRIPTION
IAB1	Arts & Entertainment	IAB14	Society
IAB2	Automotive	IAB15	Science
IAB3	Business	IAB16	Pets
IAB4	Careers	IAB17	Sports
IAB5	Education	IAB18	Style & Fashion
IAB6	Family & Parenting	IAB19	Technology & Computing
IAB7	Health & Fitness	IAB20	Travel
IAB8	Food & Drink	IAB21	Real Estate
IAB9	Hobbies & Interests	IAB22	Shopping
IAB10	Home & Garden	IAB23	Religion & Spirituality
IAB11	Law Government & Politics	IAB24	Uncategorized
IAB12	News	IAB25	Non-Standard Content
IAB13	Personal Finance	IAB26	Illegal Content

3.2 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.
2	XHTML banner ad.
3	JavaScript ad; must be valid XHTML (i.e., script tags included).

3.3 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

3.4 Ad Position

The following table specifies the position of the ad as a relative measure of visibility or prominence. The default is “0” indicating a prominent position such as a header.

VALUE	DESCRIPTION
1	Definitely visible without scrolling (i.e., “above the fold”).
2	May or may not be immediately visible depending on screen size and resolution.
3	High likelihood that the ad will initially appear off-screen (i.e., “below the fold”).

3.5 Price Units

The following table indicates the options for units used on bids and settlement prices. This is essentially a multiplier applied to the currency in effect. The default is “0” indicating CPM.

VALUE	DESCRIPTION
0	CPM basis (i.e., the cost if one were to buy 1,000 impressions at the unit price).
1	Per unit impression. For example, $1 CPM would be expressed as 0.001.
2	Micros per unit impression. For example, $1 CPM would be expressed as 1,000.

3.6 No-Bid Reason

The following table defines the reference values for a bidder expressing to the exchange the reason for not bidding. This is optional feedback, but may help the exchange tune the types of requests being sent to the bidder. The default is “0” indicating reason unknown.

VALUE	DESCRIPTION
0	Reason unknown.
1	Impression is not needed by bidder (e.g., no matching campaigns at this time).
2	Impression violates bidder’s filter settings (i.e., never should have been sent).
3	Technical error.

Appendix A. Additional Information

- ❑ Creative Commons / Attribution No-Derivatives License
  http://creativecommons.org/licenses/by-nd/3.0
- ❑ IAB (Interactive Advertising Bureau)
  http://www.iab.net
- ❑ IAB / Networks & Exchanges QA Guidelines / Content Categories
  http://www.iab.net/media/file/NE-QA-Guidelines-Final-Release-0610.pdf
- ❑ JavaScript Object Notation (JSON)
  http://www.json.org
- ❑ MMA (Mobile Marketing Association)
  http://mmaglobal.com
- ❑ OpenRTB Project
  http://code.google.com/p/openrtb