# BidSwitch Documentation

***Release 5.3***

**BidSwitch**

**May 03, 2022**

# OPENRTB 2.5 SUPPORT


**1 BidSwitch Buyer Protocol v5.3** **2**

1.1 Protocol Release Notes . . . 2

1.1.1 3rd February 2022 . . . 2
1.1.2 24th September 2021 . . . 3
1.1.3 18th August 2021 . . . 3
1.1.4 31st May 2021 . . . 3
1.1.5 28th May 2021 . . . 3
1.1.6 14th April 2021 . . . 3
1.1.7 30th March 2021 . . . 4
1.1.8 25th February 2021 . . . 4
1.1.9 21st December 2020 . . . 4
1.1.10 7th December 2020 . . . 4
1.1.11 12th November 2020 . . . 5
1.1.12 15th September 2020 . . . 5
1.1.13 7th September 2020 . . . 5
1.1.14 30th April 2020 . . . 6
1.1.15 2nd February 2020 . . . 6
1.1.16 5th January 2020 . . . 6
1.1.17 15th November 2019 . . . 6
1.1.18 15th October 2019 . . . 6
1.1.19 5th September 2020 . . . 7
1.1.20 16th August 2019 . . . 7
1.1.21 25th July 2019 . . . 7
1.1.22 27th May 2019 . . . 7
1.1.23 v5.3 - November 2017 . . . 7
1.1.24 v5.2 - February 2017 . . . 9
1.1.25 v5.1 - October 2016 . . . 10
1.1.26 v5.0 - July 2016 . . . 11
1.1.27 v4.3 - May 2016 . . . 11
1.1.28 v4.2 - March 2016 . . . 12
1.1.29 v4.1 - September 2015 . . . 13
1.1.30 v4.0 - May 2015 . . . 14

1.1.31 v2.6 - February 2015 . . . 15
1.1.32 v2.5 - December 2014 . . . 15
1.1.33 v2.4 - October 2014 . . . 16
1.1.34 v2.3 - September 2014 . . . 17
1.1.35 v2.2 - June 2014 . . . 18
1.1.36 v2.1 - April 2014 . . . 18
1.1.37 v2.0 - February 2014 . . . 19
1.2 Compatibility and Deprecation . . . 20
1.2.1 Deprecation . . . 20
1.2.2 Deprecation Timeframe . . . 20
1.2.3 Backward Compatibility . . . 20
1.3 Protocol Version Differences . . . 21
1.3.1 nurl Response Difference . . . 22
1.3.2 Deprecated 2.x Properties . . . 22
1.3.3 Valid Response Formats . . . 23
1.4 5.x Updating Overview . . . 24
1.4.1 v5.3 . . . 25
1.4.2 v5.0 and v5.1 . . . 26
1.5 **burl** Field Overview . . . 30
1.5.1 What does the **burl** represent? . . . 30
1.5.2 Server to Server Restriction . . . 30
1.5.3 **burl** Response Examples . . . 31
1.6 BidSwitch OpenRTB Differences . . . 33
1.6.1 Bid Request Differences . . . 34
1.6.2 Bid Response Differences . . . 34
1.7 Data Compression (gzip) . . . 35
1.7.1 Sending Bid Responses to BidSwitch . . . 35
1.7.2 Receiving Bid Requests from BidSwitch . . . 35
1.8 Macros . . . 35
1.8.1 Win price macro . . . 35
1.8.2 Supplier Click Tracking URL Macro . . . 36
1.8.3 True Price Macro . . . 37
1.8.4 Total Impressions Macro . . . 38
1.9 Sensitive Categories and Rich Media . . . 39
1.9.1 Supported Rich Media Frameworks . . . 39
1.10 Supplier Custom Categories . . . 40
1.11 Server-to-Server (s2s) Calls . . . 40
1.11.1 Expected s2s Behaviour . . . 41
1.11.2 s2s Headers . . . 41
1.12 Bid Request v5.3 . . . 42
1.12.1 Impression Object . . . 44
1.12.2 Banner Object . . . 50
1.12.3 Format Object . . . 52
1.12.4 Video Object . . . 52
1.12.5 Audio Object . . . 58
1.12.6 Native Object . . . 62
1.12.7 Native Request Object . . . 64
1.12.8 Device Object . . . 72

1.12.9 Geo Object . . . 76
1.12.10 User Object . . . 77
1.12.11 Site Object . . . 84
1.12.12 App Object . . . 86
1.12.13 Content Object . . . 88
1.12.14 TV Object . . . 90
1.12.15 DOOH Object . . . 91
1.12.16 Metric Object . . . 92
1.12.17 Private Marketplace Object . . . 92
1.12.18 Source Object . . . 93
1.12.19 Deal Object . . . 97
1.12.20 Publisher Object . . . 99
1.12.21 Ext Object . . . 100
1.12.22 Regs Object . . . 104
1.12.23 Data Object . . . 106
1.12.24 Bid Request JSON Examples . . . 107
1.13 Bid Response v5.3 . . . 138
1.13.1 Bid Response Ext Object . . . 139
1.13.2 Seat Bid Object . . . 140
1.13.3 Bid Object . . . 140
1.13.4 Native Response Object . . . 150
1.13.5 Supplier Specific Fields . . . 157
1.13.6 Data Response Object . . . 157
1.13.7 Bid Response JSON Examples . . . 158

**Note:** Some sections require login as they are for BidSwitch customers only. These are accessed using IPONWEB LDAP credentials and are denoted with lock symbol. All publicly available sections are also available on the BidSwitch Protocols[1] site.

BidSwitch provides real-time bid/offer access for supply and demand partners across all online media advertising types (display, mobile, video, native, dooh etc) and allows advertising platforms to engage with new Buyers and Suppliers, enter new geographical regions, and utilise new media types by providing a unifying integration layer for parties using many different technologies to run their businesses.

BidSwitch is a technology developed at IPONWEB to simplify the integration of Demand Side Platforms (DSPs) with Sell Side Platforms (SSPs). The key idea behind the service is to enable quick and cheap integration for Buyers with Suppliers without many rounds of specification refinements and development between them.

BidSwitch acts as this intermediary between Buyers and Suppliers by developing custom protocols based on the OpenRTB 2.x standards. This lets all parties integrated with the BidSwitch platform conduct business with each other through a single point of integration. The BidSwitch Protocol[2] section defines the protocol for interaction between DSPs and BidSwitch.

[1] https://protocol.bidswitch.com/
[2] https://protocol.bidswitch.com/standards/standards.html#bsw-proto

# CHAPTER 1

# BIDSWITCH BUYER PROTOCOL V5.3

This guide covers the latest version of the BidSwitch Real-time Bidding Protocol. The BidSwitch bidding protocol is based on the latest OpenRTB Protocol Specification V2.5[3], but an exact match is not possible due to the evolving nature of both the business, and the OpenRTB specification. See the *BidSwitch OpenRTB Differences* (page 33) section for more information.

---

**Note:** It is best practice to always integrate with BidSwitch using the latest version of the protocol, for more information, see the *Compatibility and Deprecation* (page 20) section.

---

#### Version Information

## 1.1 Protocol Release Notes

### 1.1.1 3rd February 2022

Changes made to how headers are passed on s2s BURL calls, now both the *x-forwarded-for* and *x-user-agent* headers are filled when available, see the *Server-to-Server (s2s) Calls* (page 40) section for more details.

[3] http://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf

### 1.1.2 24th September 2021

Added support for the `site|app.ext.inventorypartnerdomain` bid request field. This field was added in the (app-)ads.txt v1.0.3 specification[4] update. See the *App Object* (page 86) or *Site Object* (page 84) section for more details.

### 1.1.3 18th August 2021

Added support for the optional `seatbid.bid.bundle` field in bid responses; see *Bid Object* (page 140) for details.

### 1.1.4 31st May 2021

- `user.ext.xuid` Removed BidSwitch's own cross-platform User ID. This earlier field has been superseded by the industry standard Extended Identifiers field that fulfils the same function, see the *Extended Identifiers* (page 81) section for more details.

### 1.1.5 28th May 2021

Support updated for the SKAdNetwork[5] protocol response extension to include the addition of the `fidelities` object. See the *SkAdNetwork Extension* (page 148) section for more details.

- `seatbid.bid.ext.skadn.fidelities`
- `seatbid.bid.ext.skadn.fidelities.fidelity`
- `seatbid.bid.ext.skadn.fidelities.nonce`
- `seatbid.bid.ext.skadn.fidelities.timestamp`
- `seatbid.bid.ext.skadn.fidelities.signature`

### 1.1.6 14th April 2021

Added support for passing the Federated Learning of Cohorts (FLOC) ID using the Google format, `user.ext.floc`, see the *User Object* (page 77) section for details.

---

**Note:** Due to how the FLoC ID is supported in Prebid[6], its value may also be passed in the `user.ext.eids` field with its source specified as `chrome.com`, see the *Extended Identifiers* (page 81) section or details.

---

[4] https://iabtechlab.com/ads-txt/

[5] https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/extensions/community_extensions/skadnetwork.md

[6] https://github.com/prebid/Prebid.js/pull/6521/

### 1.1.7 30th March 2021

Added support for the following fields:

- `site.ext.amp`
- `geo.accuracy`
- `geo.metro`

### 1.1.8 25th February 2021

Updated support for the IAB SKAdNetwork Specification[7], which resulted in the following bid request additions.

- `imp.ext.skadn.versions`,see the *Impression Ext* (page 46) section for more details
- `imp.ext.skadn.skadnetlist`
- `imp.ext.skadn.skadnetlist.max`
- `imp.ext.skadn.skadnetlist.excl`
- `imp.ext.skadn.skadnetlist.addl`
- `device.ext.atts`, see the *Device Object* (page 72) section for more details

### 1.1.9 21st December 2020

- Support added for `user.ext.eids`. This is the Open RTB Extended User Identifiers[8] community extension. See the *Extended Identifiers* (page 81) section for details.
- Deprecated the `user.ext.xuids` field which duplicates the `eids` functionality, see the *Extended Identifiers* (page 81) section for details

### 1.1.10 7th December 2020

- Added the `native.ext.adchoicesurl_required` request field to indicate when a Buyer must return an adchoices URL in the response, see the *Native Extension Object* (page 62) section for more details.
- Added the `imp.pmp.deals.ext.buyer_wseat` field. This specifies the Advertisers/ Agencies that should have access to this deal in the DSP's system, see the *Deal Extension Object* (page 98) section for more details.
- Added support for the blocked apps `bapp` field, see the *Bid Request v5.3* (page 42) section for details

[7] https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/extensions/community_extensions/skadnetwork.md

[8] https://github.com/InteractiveAdvertisingBureau/openrtb/tree/master/extensions/2.x_official_extensions

### 1.1.11 12th November 2020

Added fields to complete support for the IAB Tech Lab OTT/CTV Store Assigned App Identification Guidelines[9] and the Guidelines for Identifier for Advertising (IFA) on CTV/OTT platforms[10]

- `app.storeid` see the *App Object* (page 86) section for more details
- `device.ext.ifa_type` see the *Device Object* (page 72) section for more details
- `device.ext.truncated_ip` added to support Google's Display & Video 360 OpenRTB Specification[11]

### 1.1.12 15th September 2020

Added support for the content object, see *Content Object* (page 88). This object can be passed in `site.content` or `app.content`

### 1.1.13 7th September 2020

The release adds support for iOS 14 based on the OpenRTB SKAdNetwork Extension[12].

#### Bid Request Fields

- `imp.ext.skadn` See the *Impression Ext* (page 46) section for more details
- `imp.ext.skadn.version`
- `imp.ext.skadn.sourceapp`
- `imp.ext.skadn.skadnetids`
- `imp.ext.skadn.skadnhsh`
- `device.ext.idfv` See the *Device Object* (page 72) for more details
- `user.ext.impdepth` See the *User Object* (page 77) for more details
- `user.ext.sessionduration`
- `device.ifa` Updated validation to support Apple's deprecation of the IDFA as part of iOS 14, this field should be passed using all zeros, rather than removed or an empty string, as all zeros is the expected format, e.g. `"00000000-0000-0000-0000-000000000000"`
- See the *In App iOS 14* (page 114) section for an iOS 14 bid request example

[9] https://iabtechlab.com/wp-content/uploads/2020/08/IAB-Tech-Lab-OTT-store-assigned-App-Identification-Guidelines-2020.pdf
[10] https://iabtechlab.com/wp-content/uploads/2018/12/OTT-IFA-guidelines.final_Dec2018.pdf
[11] https://developers.google.com/display-video/ortb-spec#supported-extension-for-device-object
[12] https://github.com/InteractiveAdvertisingBureau/openrtb

#### Bid Response Fields

- `seatbid.bid.ext.skadn.version` See the *Bid Ext Object* (page 146) section for more details.
- `seatbid.bid.ext.skadn.network`
- `seatbid.bid.ext.skadn.campaign`
- `seatbid.bid.ext.skadn.itunesitem`
- `seatbid.bid.ext.skadn.nonce`
- `seatbid.bid.ext.skadn.sourceapp`
- `seatbid.bid.ext.skadn.timestamp`
- `seatbid.bid.ext.skadn.signature`
- See the *Response to iOS 14* (page 160) section for an iOS 14 response example

### 1.1.14 30th April 2020

- Added `seatbid.bid.ext.data`, which can be used to return arbitrary data from Buyers, if the SSP supports this response field.

### 1.1.15 2nd February 2020

- Added `ext.tgroup`, to indicate the targeting group which the request matched, see the tgroups section for more details

### 1.1.16 5th January 2020

- Added `deals.ext.type`, see *Deal Object* (page 97)

### 1.1.17 15th November 2019

- Added `regs.ext.us_privacy` to support CCPA, see the *Regs Object* (page 104) for more details, or the ccpa section for a CCPA overivew.

### 1.1.18 15th October 2019

Added support for the following fields.

- `source.ext.schain` Adds support for the Supply Chain Object[13]

[13] https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/supplychainobject.md

- `source.ext.omidpv` Adds support for the Open Measurement SDK[14], see the *Source Object* (page 93) section for more details.
- `source.ext.omidpn`
- `imp.ext.wopv` Passes the WhiteOps MediaGuard Prediction ID

#### 1.1.19 5th September 2020

- Removed support for `device.didmd5` and `device.didsha1`

#### 1.1.20 16th August 2019

- Added `user.ext.xuid` to support Cross-platform User IDs (UUIDs), see the *Extended Identifiers* (page 81) section for more details

#### 1.1.21 25th July 2019

- Added `imp.ext.ssai` which signals support for stitching an ad into an audio or video stream, see the *Impression Ext* (page 46) section for more details.

#### 1.1.22 27th May 2019

- Added `imp.ext.notification_type`, see the *Impression Ext* (page 46) section for more details.
- Added `ext.google_query_id`

#### 1.1.23 v5.3 - November 2017

##### New Protocol Fields

Bid Request fields:

- `deals.ext.type`, see *Deal Object* (page 97)
- `regs.ext.us_privacy`
- `regs.ext.gdpr`, `user.ext.consent`, `user.ext.google_consent` and `user.ext.consented_providers_settings` added to support GDPR compliance, see the gdpr section for more details
- `imp.native.request.plcmttype`, see *Native Request Object* (page 64)
- `imp.native.request.privacy`

[14] https://github.com/InteractiveAdvertisingBureau/AdCOM/blob/master/OpenRTB%20support%20for%20OMSDK.md

- `imp.native.request.context`, see *Native Request Object* (page 64) and *Context Type Description* (page 67)
- `imp.native.request.contextsubtype`, see *Native Request Object* (page 64) and *Context Sub-Type ID Description* (page 68)
- `imp.native.request.eventtrackers`, see *Event Tracker Request Object* (page 68)
- `ext.ads_txt` added to support ads-txt
- `ext.ads_txt.status`
- `ext.ads_txt.auth_id`
- `ext.ads_txt.pub_id`
- `ext.ads_txt.supplier_domain`
- `ext.google_query_id`
- `ext.rubicon.ast`
- `ext.media_src`
- `imp.ext.s2s_nurl`
- `imp.ext.ssai`
- `imp.ext.wpov`
- `imp.pmp.deal.ext`
- `imp.pmp.deal.ext.data_src`
- `imp.video.skipmin`
- `imp.video.skipafter`
- `imp.video.ext.rewarded`
- `imp.ext.notification_type`
- `publisher.domain`
- `user.ext.digitrust`
- `user.ext.xuid`
- `source.pchain`
- `source.ext.schain`
- `source.ext.omidpv`
- `source.ext.omidpn`

Bid Response fields:

- `seatbid.bid.ext.native.eventtrackers`, see *Event Tracker Response Object* (page 155)
- `seatbid.bid.ext.native.privacy`, see *Native Response Object* (page 150)

- `nbr` added the ability for Buyers to return a No Bid Reason, see the dsp-nbr section for more details
- `ext.true_price_opt_out`
- `seatbid.bid.ext.at1`
- `seatbid.bid.ext.data`

#### Updated Protocol Fields

- `ext.ssp`

#### Deprecated Protocol Fields

- `ext.s2s_nurl`, replaced by `imp.ext.s2s_nurl`
- Both the `seatbid.bid.ext.native.jstracker` and `seatbid.bid.ext.native.imptracker` fields have been replaced by `seatbid.bid.ext.native.eventtrackers`
- `device.didmd5` and `device.didsha1`

#### Updated or New Sections

- ads-txt
- *Server-to-Server (s2s) Calls* (page 40)
- *5.x Updating Overview* (page 24)
- gdpr

### 1.1.24 v5.2 - February 2017

#### New Protocol Fields

Bid Request fields:

- `imp.video.placement`
- `imp.video.playbackend`
- `imp.video.skip`
- `device.mccmnc`
- `source.fd`
- `imp.native.request.assets.video.ext` see, *Native Request Object* (page 64)
- `imp.native.request.assets.video.ext.playbackmethod`
- `imp.native.ext` see *Native Object* (page 62)
- `imp.native.ext.triplelift`
- `imp.native.ext.triplelift.formats`
- `imp.metric`, see *Metric Object* (page 92)

Bid Response fields:

- `seatbid.bid.burl`, and see the *Using the burl Field* (page 25) section for more details.

- `seatbid.bid.language`
- `seatbid.bid.ext.native.jstracker`
- `seatbid.bid.ext.native.assets.video.ext`, see *Native Asset Video Object* (page 153)
- `seatbid.bid.ext.native.assets.video.ext.playbackmethod`

#### Updated Protocol Fields

- `seatbid.bid.nurl`
- `ext.s2s_nurl`

#### Deprecated Fields

- `imp.ext.viewability` bid request field, replaced by the use of the *Metric Object* (page 92)

#### Recently Added or Updated Sections

- ca-submit, a new section.
- bsw-data-centers, updated with new Data Centre Information.
- *5.x Updating Overview* (page 24), updated with `burl` information.
- Added the *MicroAd 3PAS List* (page **??**) list to the *Supplier Custom Categories* (page 40) section.
- Added the *BidSwitch OpenRTB Differences* (page 33) section.
- Added the ca-overview section.
- Added the *burl Field Overview* (page 30) section.
- Added the *Server-to-Server (s2s) Calls* (page 40) section.

### 1.1.25 v5.1 - October 2016

#### New Protocol Fields

Added the following bid request fields:

- `ext.dooh`
- `video.ext.vast_url_rq`
- `user.data.segment.name`
- `publisher.cat`

#### Updated Protocol Fields

- The `seatbid.bid.ext.vast_url` description, this field can be omitted for some bid requests.

#### New Sections

- The *DOOH Ad Example* (page 135) section.
- The ss-bid-weight section, and the `ext.wt` Bid Request field.

### 1.1.26 v5.0 - July 2016

#### New Sections

- Added the *5.x Updating Overview* (page 24) section
- Added the `video.ext.vast_url_rq` bid request field, see the *Video Ext Object* (page 56) for details.
- Added the `seatbid.bid.ext.daast_url` bid response field.
- Added the *Audio Bid Response* (page 163) example

#### Updated Sections

- Updated the `seatbid.bid.ext.vast_url` description, this field can only be used for video trading.
- Updated the `adm` and `nurl` fields, see the *Bid Object* (page 140) section.
- Updated the *Macros* (page 35) section with added win price macro usage options.
- Updated the *nurl Response Difference* (page 22) section with information about how to use the `nurl` and `adm` fields correctly.

### 1.1.27 v4.3 - May 2016

#### Added

Added the *Compatibility and Deprecation* (page 20) and *Protocol Version Differences* (page 21) sections to help users keep their BidSwitch integration up to date.

Added the following bid response fields:

- `seatbid.bid.ext.native.ext.viewtracker`
- `seatbid.bid.ext.native.ext.adchoiceurl`

Added the following bid request fields:

- `imp.audio`

- `device.geo.utcoffset`
- `video.ext.outstream`
- `ext.tv`
- `imp.banner.format`

Added programmatic TV bid request example.

#### Updated

- Updated the `badv`, `site` and `app` field descriptions.
- Bid response fields `seatbid.bid.ext.native.assets.img.h` and `seatbid.bid.ext.native.assets.img.w` are now required.
- The `imp.pmp.private_auction` field is now optional.
- Deprecated the `imp.banner.ext.extra_sizes` bid request field.
- The `ext.vast_url` field now supports DAAST (audio) creatives.

#### Removed

- Removed fields and objects deprecated in *v4.0 - May 2015* (page 14).
- Removed the `ext.creative_params` and `data` bid request fields.

### 1.1.28 v4.2 - March 2016

#### Added

Added support for the following bid request fields:

- `allimps`
- `ext.s2s_nurl`
- `imp.exp`
- `imp.video.maxextended`
- `imp.video.boxingallowed`
- `imp.video.playbackmethod`
- `imp.video.delivery`
- `imp.video.sequence`
- `device.w`
- `device.h`
- `device.lmt`
- `device.pxratio`
- `site.privacypolicy`
- `app.privacypolicy`
- `user.yob`
- `user.gender`
- `user.data`
- `user.ext.ug`
- `user.ext.cookie_age`

Added the following bid response fields:

- `seatbid.bid.crid`
- `seatbid.bid.ext.agency_id`

- Added the *Compatibility and Deprecation* (page 20) section.
- Added the *Data Compression (gzip)* (page 35) section.

#### Updated

- BidSwitch now supports Compressed JSON as an alternative bid request and bid response data format.
- The `seatbid.bid.adid` bid response field is now optional, and the `seatbid.bid.crid` field may be used instead of it.
- Removed fields and objects deprecated in *v2.6 - February 2015* (page 15).

### 1.1.29 v4.1 - September 2015

#### Added

Added the following bid request fields:

- `site.ref`
- `ext.clktrkrq`
- `ext.gumgum.cat`
- `imp.tagid`
- `imp.native.request.assets.img.mimes`
- `user.keywords`
- `imp.ext.yieldone.cat`
- `imp.bidfloorcur`
- `pmp.deal.bidfloorcur`

#### Updated

- Updated the bid request currency array so that it may contain multiple values.
- Extended enumeration for the `imp.video.ext.player_type` bid request field.
- Updated the `imp.video.mimes` bid request field description.
- Updated the `seatbid.bid.nurl` bid response field description.
- Changed the accepted `seatbid.bid.cat` type bid response field to a strings array.
- Updated the *Supplier Click Tracking URL Macro* (page 36) section.
- Updated the *Bid Request JSON Examples* (page 107) section, added new example requests and responses.

#### Removed

Removed fields and objects deprecated in *v2.5 - December 2014* (page 15).

### 1.1.30 v4.0 - May 2015

#### Added

- Added the *Supported Rich Media Frameworks* (page 39) section.
- Added native inventory support.
- Added the *Data Object* (page 106) and *Segment Object* (page 106).

Added the following bid request fields:

- `site.mobile`
- `device.geo.type`
- `imp.iframebuster`

Added the following bid response fields:

- `seatbid.bid.cat`
- `seatbid.bid.adm`
- `seatbid.bid.ext.native`

#### Updated

- Updated the `imp.banner.btype` bid request field description.
- Updated the `seatbid.bid.nurl` bid response field description.
- Updated the *Macros* (page 35) section, `${CLICK_URL}` is no longer supported.
- Bid response fields `seatbid.seat` and `seatbid.bid.nurl` are now required.
- Bid response `seatbid.bid.iurl` field is now required for banner ads.

#### Removed

- Removed fields and objects deprecated in *v2.4 - October 2014* (page 16).
- Removed the *Creative approval* and *Additional Features* sections.
- Deprecated `site.ext.mobile_site` and `ext.ghostery` bid request fields.
- Deprecated `seatbid.bid.ext.cat` bid response field.

Removed the following bid response fields. See this *Protocol Version Differences* (page 21) section for more details about how this affects the different major versions.

- `seatbid.bid.ext.js_url`
- `seatbid.bid.ext.img_url`
- `seatbid.bid.ext.click_url`

### 1.1.31 v2.6 - February 2015

#### Added

Added the following bid request fields:

- `imp.secure`
- `imp.banner.ext.extra_sizes`
- `imp.ext.viewability`

Added the following bid response fields:

- `seatbid.bid.ext.lpdomain`
- `seatbid.bid.h`
- `seatbid.bid.w`
- `seatbid.bid.iurl bid`

#### Deprecated

- Deprecated the `ext.is_secure` bid request field. The new `imp.secure` field replaces this, see the *Impression Object* (page 44) section for more details.

#### General

- Removed creative preview section.
- Removed fields and objects deprecated in *v2.3 - September 2014* (page 17).
- Updated request and response samples.

### 1.1.32 v2.5 - December 2014

#### Added

Added the following bid request fields:

- `imp.video.ext.player_type`
- `imp.ext.yieldone.allowed_creative_category_id`

Added the following bid response fields:

- `seatbid.bid.ext.yieldone.creative_category_id`
- `ext.protocol`
- `seatbid.bid.attr`
- `seatbid.bid.dealid`

#### Updated

- Click tracking macro is now required in OpenX bids.
- The field `imp.pmp.deal.at` is now optional.

#### Removed

- Removed fields and objects deprecated in *v2.2 - June 2014* (page 18).
- Deprecated the `seatbid.bid.ext.deal` bid response field.

### 1.1.33 v2.4 - October 2014

#### Added

Added the following bid request fields:

- `tmax`
- `app.paid`
- `regs.coppa`
- `device.dnt`
- `device.ipv6`
- `device.js`
- `device.flashver`
- `device.devicetype`
- `site.name`
- `imp.instl`
- `imp.displaymanagerapp`
- `imp.displaymanagerver`
- `imp.ext.yieldone.allowed_creative_types`
- `imp.video.pos`
- `imp.ext.inventory_class`

Added the following bid response field:

- `seatbid.bit.ext.yieldone.creative_type`

#### Updated

Changed the `imp.pmp.deals.at` values to comply with OpenRTB 2.2[15].

#### Removed

Deprecated the `imp.ext.yieldone.inventory_class` bid request field.

[15] http://www.iab.com/wp-content/uploads/2015/06/OpenRTBAPISpecificationVersion2_2.pdf

### 1.1.34 v2.3 - September 2014

#### Added

- Added *Video Object Example* (page 57).
- Added Creative Approval section.

Added the following bid requests fields:

- `site.ext.mobile_site`
- `imp.banner.mimes`

Added the following bid request objects:

- `imp.ext.yieldone`
- `imp.pmp`
- `ext.creative_params`

Added the following bid response fields:

- `seatbid.bid.ext.duration`
- `seatbid.bid.ext.img_url`
- `seatbid.bid.ext.click_url`

#### Updated

- Updated the user-match section.
- The field `imp.video.linearity` is now an optional field.
- Updated the `btype` field description.

---

**Important:** Usage of `nurl`, `ext.js_url` or `ext.liveintent.img_url` bid response fields depends on the `btype` bid request field value.

---

#### Removed

- Deprecated `seatbid.bid.ext.liveintent` and `pmp` objects.

### 1.1.35 v2.2 - June 2014

#### Added

- Added the following bid request fields:
  - `imp.video.protocols`
  - `imp.banner.api`
  - `site.publisher.name`
  - `app.publisher.name`
  - `app.name`
  - `video.companionad.id`
  - `device.os`
  - `device.osv`
  - `device.make`
  - `device.model`
- Added extended sensitive categories, used in `bcat` field.
- Added `imp.ext.google` and `ext.ghostery` bid request objects.

#### Removed

- Deprecated the following bid request fields:
  - `imp.video.protocol`
  - `ext.google.excluded_attribute`
  - `ext.google.allowed_vendor_type`

### 1.1.36 v2.1 - April 2014

#### Added

- Added `seatbid.bid.ext.js_url` bid response field.
- Added the following bid request fields:
  - `pmp.deals.wseat`
  - `imp.banner.expdir`
  - `at`

#### Updated

- Bid response `seatbid.bid.nurl` field is optional now.

### 1.1.37 v2.0 - February 2014

#### Added

- Added application traffic support.
- Added the following bid request fields:
  - `ext.google.excluded_attribute`
  - `ext.google.allowed_vendor_type`
- Added the following bid response fields:
  - `seatbid.bid.ext.google`
  - Added `seatbid.bid.cid`

#### Updated

- Bid request `site` object is optional now.

## 1.2 Compatibility and Deprecation

The BidSwitch protocol is a constantly evolving technology. As such, it regularly undergoes changes which behoove customers to keep their integration with the BidSwitch platform up to date. Along with the *BidSwitch Buyer Protocol v5.3* (page 2) documentation, use this section and the *Protocol Version Differences* (page 21) section to keep your integration up to date.

### 1.2.1 Deprecation

Any part of the specification which gets deprecated will be marked using the following style. In most circumstances a pointer to what you should do to update your implementation of the specification will be provided.

Deprecated since version X.Y.

### 1.2.2 Deprecation Timeframe

Deprecated feature support in the codebase is guaranteed for five specification releases after deprecation. Starting with the sixth specification release after deprecation, any deprecated feature is used at the at Buyers's own risk.

### 1.2.3 Backward Compatibility

All specifications with the same major version number are backward compatible. Specifications with different major version numbers are not guaranteed to be backward compatible. You can compare previous specifications against the current implementation using the following PDFs.

- BidSwitch 2.1 PDF
- BidSwitch 2.2 PDF
- BidSwitch 2.3 PDF
- BidSwitch 2.4 PDF
- BidSwitch 2.5 PDF
- BidSwitch 2.6 PDF
- BidSwitch 4.0 PDF
- BidSwitch 4.1 PDF
- BidSwitch 4.2 PDF
- BidSwitch 4.3 PDF
- BidSwitch 5.0 PDF
- BidSwitch 5.1 PDF
- BidSwitch 5.2 PDF
- BidSwitch 5.3 PDF

## 1.3 Protocol Version Differences

**Note:**

- This section covers the difference between 2.x and 4.x. For 5.x information see the *5.x Updating Overview* (page 24) section.
- The differences between each protocol affect only the format and interpretation of the bid response. Bid requests work exactly the same.

The BidSwitch protocol has three major versions, 2.X, 4.X, and 5.x. Certain fields and the interpretation of data in them differ between each version. This section outlines the differences between 2.x and 4.x.

- The bid response `nurl` field is interpreted differently, see the *nurl Response Difference* (page 22) section for details.
- Ensure that the protocol version field, which is used to define how the response will be interpreted by BidSwitch, is set to the correct protocol version. See the *Bid Response Ext Object* (page 139) section for more details.
- In 4.x the `adm` field is required for display ads, see the *Bid Object* (page 140) section for more details.
- The following bid extension fields are valid in 2.X, but not in 4.X. For more details, see the *Deprecated 2.x Properties* (page 22) section.
  - `img_url`
  - `js_url`
  - `click_url`

### 1.3.1 nurl Response Difference

The `nurl` bid response field is handled differently between each version. Use the table below to understand the expected response format for each version. For more information, see the *Bid Object* (page 140) section.

Table 1: nurl Version Differences

| Version | Properties |
|---|---|
| 2.x | The win notice URL.<br>• For banner impressions the ad markup should be returned via this URL. Ad markup should be in HTML format.<br>• For video impressions the URL should contain the URL of impression pixel.<br>• The URL may contain substitution macros, see the *Macros* (page 35) section.<br>• For application banner inventory this field will be IGNORED, use `ext.js_url` instead. |
| 4.x | The win notice URL.<br>• This field should **not** be used for submitting creative markup.<br>• The URL should contain the win price macro, see the *Macros* (page 35) section.<br>• As this URL is called from the user's browser, it should be SSL-compliant for requests with `imp.secure` set to `1`.<br>• This URL will be called by a server to server call if the bid request has the `ext.s2s_nurl` field value set to `1`. |
| 5.x | The win notice URL.<br>• Similar to 4.x, except that the win price macro can only be used in one field per bid response. Either the `adm` or `nurl` field. See the *5.x Updating Overview* (page 24) or *Bid Object* (page 140) section for details. |

### 1.3.2 Deprecated 2.x Properties

The following fields are not valid in a 4.x response, but are part of the 2.x protocol. See the *Bid Ext Object* (page 146) section for the latest protocol standards.

Table 2: Bid Ext 2.X Objects

| Field | Value | Description |
|---|---|---|
| *img_url** | *string* | The URL of the creative image. In order to receive the user cookie and win price, this url should point to the Buyer handler and redirect to the actual creative location. The url may contain the win price macro, e.g. `${AUCTION_PRICE}`, but not the click macro.<br>If this field is present, the `nurl` field of the bid response will be ignored. |
| *click_url** | *string* | The creative click URL. Required if the `img_url` field is present. |
| *js_url** | *string* | A Javascript-based win notice URL.<br>• For in-app inventory, the ad markup should be returned using this URL.<br>• For website or video inventory this field may be used as a substitute for the `nurl` field.<br>• Ad markup should be in JavaScript format.<br>• The URL may contain macros, see the *Macros* (page 35) section for more details. |

### 1.3.3 Valid Response Formats

For valid 4.x response examples, see the examples section in the 4.2[16] guide. If you wish to check for valid 2.x response formats see the 2.x guides, for example 2.6[17] or 2.5[18].

The following diff examples show how the expected JSON response has changed between version. The green represents the changes in 4.x, and the red shows version 2.x syntax.

#### Example Response diff

```
   {
      "id":"1234567890",
      "ext":{
-        "protocol":"2.5"
+        "protocol":"4.2"
      },
      "seatbid":[
         {
            "bid":[
               {
                  "id":"1",
```

(continues on next page)

[16] https://docs.bidswitch.com/_downloads/BidSwitch_specs_4.2.pdf
[17] https://docs.bidswitch.com/_downloads/BidSwitch_specs_2.6.pdf
[18] https://docs.bidswitch.com/_downloads/BidSwitch_specs_2.5.pdf

(continued from previous page)

```
                    "impid":"102",
                    "price":9.43,
                    "adid":"314",
                    "cid":"42",
                    "cat":[
                       "IAB12"
                    ],
+                   "adm":"<a href=\"http://adserver.com/click?adid=12345&tracker=$
↪{CLICK_URL:URLENCODE}\"><img src=\"http://image1.cdn.com/impid=102\"/></a>",
-                   "nurl":"http://adserver.com/winnotice?impid=102&winprice=$
↪{AUCTION_PRICE}&click_url=${CLICK_URL:URLENCODE}",
+                   "nurl":"http://adserver.com/winnotice?impid=102&winprice=$
↪{AUCTION_PRICE}",
                    "iurl":"http://adserver.com/preview?crid=314",
                    "adomain":[
                       "advertiserdomain.com"
                    ],
                    "ext":{
                       "advertiser_name":"Coca-Cola",
                       "language":"en",
                       "cat":"IAB12",
-                      "img_url":"http://adserver.com/imp?impid=102&winprice=$
↪{AUCTION_PRICE}",
-                      "click_url":"http://adserver.com/click?impid=102",
                       "agency_name":"CC-advertising"
                    }
                 }
              ]
              ],
+             "seat":"4"
          }
       ]
    }
```

## 1.4 5.x Updating Overview

Use the information in the following sections to understand the slight differences between each version of the v5.x protocol.

- *v5.3* (page 25)
  - *Using the burl Field* (page 25)
- *v5.0 and v5.1* (page 26)
  - *URL Formatting* (page 26)

– *Server to Server Restriction* (page 27)
– *Upgrading from 4.x* (page 27)
– *Upgrading from 2.x* (page 27)

### 1.4.1 v5.3

#### Using the `burl` Field

#### Display

- The `burl` field is only valid from v5.2 onwards.
- The `burl` should contain the `${AUCTION_PRICE}` macro.
- The `${AUCTION_PRICE}` macro should be a part of a valid URL.
- If the `burl` field is given, the `nurl` field becomes optional. Both fields are called, unless the Bid Request specifies a s2s notification only by setting the `imp.ext.s2s_nurl` field set to `1`.
- If the Bid Request sets the `imp.ext.s2s_nurl` field value to `1`, only the `burl` field will be called. Therefore, use the `burl` field to pass the win price macro.
- It is allowed to respond with a non-secure `burl` for secure bid requests.
- When using the `${AUCTION_PRICE}` macro in the `burl` field, it may be simultaneously present in either the `adm` or `nurl` field, not both. Using `${AUCTION_PRICE}` in the `nurl` and `adm` at the same time is not allowed. If using either of these fields, you should **NOT** omit the `${AUCTION_PRICE}` macro in the `burl` field.
- No more than one win price macro can be used in the `adm` field, otherwise BidSwitch records multiple impression events.

#### Video/Native

Video and Native use the same logic as Display.

- If the Bid Request specifies a s2s notification only, using `ext.s2s_nurl` set to `1`, then only one of either the `nurl` or `burl` fields may be used. As it is an s2s call only, if the the `burl` field is given, the `nurl` field is ignored. Therefore, you should set the `${AUCTION_PRICE}` macro in the field that will be called.
- If the Bid Request does not explicitly set an s2s call, then `nurl`, `burl`, or both can be used and both will be called. In this scenario, both fields can contain the `${AUCTION_PRICE}` macro.

```
"burl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_PRICE}",
```

### 1.4.2 v5.0 and v5.1

The main difference between 5.x and previous versions (v2.x and v4.x) is that it allows the `adm` bid response field to contain the win price macro, `${AUCTION_PRICE}`. This simplifies the protocol by removing some extended fields, making it more compliant with OpenRTB standards, and also letting Buyers know the clearing price at the time the creative markup is formed by placing the price macro in the `iframe` source URL.

Upgrading to this version from v2.x will also grant users access to features added in v4.x of the protocol such as native, audio, and programmatic TV.

There is a number of usage limitations around the price macro being placed in the `adm` field:

- No more than one win price macro can be used in the `adm` field, otherwise BidSwitch records multiple impression events.
- The `${AUCTION_PRICE}` should be a part of a valid URL.
- The `${AUCTION_PRICE}` should be present in only one of either the `adm` field or `nurl` field, never both.

---

**Note:** Only one price macro per bid response is allowed. You should not use it in the `adm` and `nurl` fields simultaneously.

---

#### URL Formatting

The price macro must be placed within a valid URL and located in the `src` attribute of an HTML tag (e.g. `<iframe>`, `<script>` or `<img>`). Use the following examples to ensure you use the correct syntax.

```
# valid URLs
"adm": "<iframe src=\"http://dsp.com/imp?bc=12345aaabbb&price=${AUCTION_PRICE}\">
↪<\/iframe>"
"adm": "<script src=\"http://dsp.com/js_imp?bc=12345aaabbb&price=${AUCTION_PRICE}\
↪"><\/script>

# invalid URL
"adm":"<img src=\"http://cdn.com/img?id=123\" onclick=\"impfn(${AUCTION_PRICE})\"/
↪>"
```

### Server to Server Restriction

Including the price macro in the `adm` field is not allowed for bid requests that use server to server impression notification. These are marked with the `ext.s2s_nurl = 1` flag.

For server to server notifications, it is required to provide the win price macro in the `nurl` field, or from v5.2 onwards the `burl` field.

### Upgrading from 4.x

The 5.x protocol is fully compatible with v4.x, so to upgrade you only need to set the bid response `ext.protocol` field to `"5.3"`

```
{
  "id": "1234567890",
  "ext": {
    "protocol": "5.3"
  },
}
```

- If you are currently using the 4.x protocol, then no other changes are required.

---

**Note:**

- In the case of expired impressions or price errors, BidSwitch forwards impression calls originating from the `adm` field to the Buyer with a clearing price of zero. If the impression call originates from the `nurl` field then BidSwitch responds with a 1x1 pixel.
- You still have the option of using the `nurl` field to send the impression clearing price as supported in 4.x versions.

---

### Upgrading from 2.x

### Upgrading Context

- The `seatbid.bid.iurl` response field is required for banner bids starting from version 4.0, so it has to be supported to migrate properly from 2.x to 5.x
- In 2.x, BidSwitch was building the actual ad markup using the parameters provided in the Buyer Bid Response (`nurl` field etc), whereas in 5.x the Buyer is fully responsible for the Ad Markup and BidSwitch keeps its interference to a minimum

### Upgrading Steps

To upgrade your implementation to 5.x and start its feature, use the following steps.

1. Set the bid response `ext.protocol` field to `"5.3"`
2. Move the impression/ad serving URL from whichever of the following valid 2.x fields it is currently used in (`nurl` | `ext.js_url` | `ext.img_url`) to the `adm` field
3. Wrap it with the appropriate HTML tag. Examples using the `<iframe>`, `script` and `img` tags are given below.

**Example 1:** Replacing the `nurl` field with `adm` and wrapping the impression link with the `<iframe>` tag.

---

**Note:** If you wrap the the impression URL using an `<iframe>` tag, then you should also define the frame width and height, as in the above example. This is to ensure better rendering of the ad on the user's device.

---

```
{
   "seatbid": {
      "bid": {
-         "nurl": "http://dsp.com/imp?bc=12345aaabbb&price=${AUCTION_PRICE}"
+         "adm": "<iframe src=\"http://dsp.com/imp?bc=12345aaabbb&price=${AUCTION_
↪PRICE}\ "height=\"300\" width=\"300\"><\/iframe>"
      }
   },
   "ext": {
-      "protocol": "2.6"
+      "protocol": "5.3"
   }
}
```

**Example 2:** Replacing the `ext.js_url` field with `adm` and wrapping the impression link with the `<script>` tag.

```
{
   "seatbid":{
      "bid":{
-       "ext":{
-            "js_url":"http://dsp.com/js_imp?bc=12345aaabbb&price=${AUCTION_PRICE}"
-         }
+       "adm": "<script src=\"http://dsp.com/js_imp?bc=12345aaabbb&price=${AUCTION_
↪PRICE}\"><\/script>"
    }

      }
   },
```

(continues on next page)

(continued from previous page)

```
   "ext":{
-       "protocol": "2.6"
+       "protocol": "5.3"
   }
}
```

**Example 3:** Replacing the `ext.img_url` field with `adm` and wrapping the impression link with the `<img>` tag.

```
{
   "seatbid":{
      "bid":{
-          "ext":{
-             "img_url":"http://dsp.com/img_imp?bc=12345aaabbb&price=${AUCTION_
↪PRICE}",
-             "click_url":"http://dsp.com/click?bc=12345aaabbb"
-         }
+          "adm": "<a href=\"http://dsp.com/click?bc=12345aaabbb\">
+                  <img src=\"http://dsp.com/img_imp?bc=12345aaabbb&price=${AUCTION_
↪PRICE}\"/><\/a>"
      }
   },
   "ext":{
-       "protocol": "2.6"
+       "protocol": "5.3"
   }
}
```

**Note:**

- In the case of expired impressions or price errors, BidSwitch forwards impression calls originating from the `adm` field to the Buyer with a clearing price of zero. If the impression call originates from the `nurl` field then BidSwitch responds with a 1x1 pixel.
- You still have the option of using the `nurl` field to send the impression clearing price as supported in 4.x versions.
- Click tracking is also different from v2.x to v5.x, to learn more about it see the *Supplier Click Tracking URL Macro* (page 36) section.

## 1.5 burl Field Overview

The `burl` field was introduced with v5.2 of the BidSwitch protocol to comply with the latest updates to the OpenRTB specification. The idea behind the introduction of the `burl` field is to standardise the billing mechanism for Server-to-Server (s2s) notifications sent to Buyers. As such, it has the following properties when used with BidSwitch:

- It must contain the win price macro
- It is always a server-to-server (S2S) call
- It can be used in conjunction with the `adm` or `nurl` field, as well as used independently. See the *burl Response Examples* (page 31), and the *5.x Updating Overview* (page 24) section for more information.

### 1.5.1 What does the burl represent?

The `burl` call signifies that BidSwitch has recorded an impression on its side. How that represents what happened on the Supplier side differs from Supplier to Supplier, specifically in the following ways:

1. Some Suppliers offer a separate call and affirm that it represents a true impression. In this case, the number of impressions may be less than the number of pure pixel calls from the creative markup, as this model covers creative pre-caching.
2. For the rest of the Suppliers, BidSwitch places a pixel in the creative markup and records the impression when this pixel is called by the browser.
3. For video and native impressions, BidSwitch receives the standard impression tracking call. For video this means the VAST impression call, and for Native it means the impression tracker. These are true impression calls by definition. For more information about Video and Native impression tracking, see the imp-track-deliver section.

### 1.5.2 Server to Server Restriction

Including the price macro in the `adm` field is not allowed for bid requests that use server to server impression notification. These are marked with the `ext.s2s_nurl = 1` flag.

For server to server notifications, it is required to provide the win price macro in the `nurl` field, or from v5.2 onwards the `burl` field.

#### Using the `burl` Field

#### Display

- The `burl` field is only valid from v5.2 onwards.
- The `burl` should contain the `${AUCTION_PRICE}` macro.
- The `${AUCTION_PRICE}` macro should be a part of a valid URL.
- If the `burl` field is given, the `nurl` field becomes optional. Both fields are called, unless the Bid Request specifies a s2s notification only by setting the `imp.ext.s2s_nurl` field set to `1`.
- If the Bid Request sets the `imp.ext.s2s_nurl` field value to `1`, only the `burl` field will be called. Therefore, use the `burl` field to pass the win price macro.
- It is allowed to respond with a non-secure `burl` for secure bid requests.
- When using the `${AUCTION_PRICE}` macro in the `burl` field, it may be simultaneously present in either the `adm` or `nurl` field, not both. Using `${AUCTION_PRICE}` in the `nurl` and `adm` at the same time is not allowed. If using either of these fields, you should **NOT** omit the `${AUCTION_PRICE}` macro in the `burl` field.
- No more than one win price macro can be used in the `adm` field, otherwise BidSwitch records multiple impression events.

#### Video/Native

Video and Native use the same logic as Display.

- If the Bid Request specifies a s2s notification only, using `ext.s2s_nurl` set to `1`, then only one of either the `nurl` or `burl` fields may be used. As it is an s2s call only, if the the `burl` field is given, the `nurl` field is ignored. Therefore, you should set the `${AUCTION_PRICE}` macro in the field that will be called.
- If the Bid Request does not explicitly set an s2s call, then `nurl`, `burl`, or both can be used and both will be called. In this scenario, both fields can contain the `${AUCTION_PRICE}` macro.

```
"burl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_PRICE}",
```

### 1.5.3 burl Response Examples

As of v5.2, when responding to Bid Requests, a win price macro can be given in the `burl` field, and the `adm` or `nurl` field.

---

**Note:**

- The win price macro cannot be given in the `adm` and `nurl` field simultaneously. See the *5.x Updating Overview* (page 24) section for more details.

- The examples used in this section are extracts from a complete Bid Response, see the *Bid Response JSON Examples* (page 158) section for full details.

### burl Response

```
{
  "seatbid":[
    {
      "bid":[
        {
          "language":"en",
          "burl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
          "adm":"<a href=\"https://adserver.com/clickadid=12345&tracker=${CLICK_
↪URL:URLENCODE}\"><img src=\"https://image1.cdn.com/impid=102\"/></a>",
          "cat":[
            "IAB12"
          ]
        }
      ]
    }
  ]
}
```

### burl + adm Response

In this response, both the `burl` and `adm` fields contain the win price macro. The `nurl` field is not given as it is optional with the use of the `burl` field..

```
{
  "seatbid":[
    {
      "bid":[
        {
          "language":"en",
          "burl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
          "cat":[
            "IAB12"
          ],
          "adm":"<iframe src=\"http://dsp.com/imp?bc=12345aaabbb&price=${AUCTION_
↪PRICE}\"><\/iframe>"
        }
      ]
```

(continues on next page)

(continued from previous page)

```
    }
  ]
}
```

#### burl + nurl Response

In this response, both the `burl` and `nurl` fields contain the win price macro. The `adm` field does not.

```
{
  "seatbid":[
    {
      "bid":[
        {
          "language":"en",
          "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}",
          "cat":[
            "IAB12"
          ],
          "adm":"<a href=\"https://adserver.com/clickadid=12345&tracker=${CLICK_
↪URL:URLENCODE}\"><img src=\"https://image1.cdn.com/impid=102\"/></a>",
          "nurl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}"
        }
      ]
    }
  ]
}
```

## 1.6 BidSwitch OpenRTB Differences

Even though the BidSwitch Real-time Bidding Protocol v5.3 is based on the latest OpenRTB Protocol Specification V2.5[19], differences exist between the two. Those differences are outlined in this section.

[19] https://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf

### 1.6.1 Bid Request Differences

The BidSwitch protocol differs in the following ways.

- `bcat` accepts additional non-IAB Blocked Advertiser Categories.
- `native.request` is an object. The OpenRTB version uses a string and passes a serialized object.
- `user.buyeruid` uses a lowercase IDFA, or Android ID for in-app traffic, if the user cookie ID is not known.

### 1.6.2 Bid Response Differences

The BidSwitch protocol differs in the following ways.

- `ext.protocol` is required.
- `seatbid.bid` array can have a maximum of two elements.
- `seatbid.seat` is required if `wseat` is present in the bid request.
- `seatbid.bid.adm` may or may not contain the win price macro depending on the protocol version, bid request parameters, and `nurl` field usage. See the *5.x Updating Overview* (page 24) section for details. The field may not be used for submitting VAST or DAAST documents.
- `seatbid.bid.iurl` is required for banner bids.
- `seatbid.bid.price` is required.
- One of either `seatbid.bid.adid` or `seatbid.bid.crid` is required.
- `seatbid.bid.adomain` is required.
- `seatbid.bid.cid` is required for a number of Suppliers.
- `seatbid.bid.cat` is required for a number of Suppliers.
- `seatbid.bid.ext.advertiser_name` is required for a number of Suppliers.
- `seatbid.bid.ext.agency_name` is required for a number of Suppliers.
- `seatbid.bid.ext.duration` is required for a number of Suppliers.
- `seatbid.bid.ext.vast_url` may be used to supply the VAST document in any video bid. It is required to be used for video bids if the `video.ext.vast_url_rq` bid request field is set to `1`. The VAST document should not contain win price macros
- `seatbid.bid.nurl`
  - Should NOT be used for submitting creative markup, and it cannot contain a win price macro if `seatbid.bid.adm` contains one.
  - May be used to supply the VAST document if the `video.ext.vast_url_rq` bid request field is set to `1`. The VAST document should NOT contain win price macros.

**Protocol Information**

## 1.7 Data Compression (gzip)

BidSwitch supports JSON and Compressed JSON (gzip) as data formats for bid requests and bid responses, as indicated using the `Accept-Encoding: gzip` header. It is recommended to use Compressed JSON to minimize the amount of data exchanged between BidSwitch and the Buyer, as this reduces latency times between servers without any additional costs. Here is what BidSwitch has observed with gzip traffic:

- When enabled, gzip compression ratios tend to be in the range of x1.4 to x2.0 (depending on the size of the request/response)
- There is no tangible extra CPU load due to compressing/decompressing traffic, i.e. there is no extra cost

### 1.7.1 Sending Bid Responses to BidSwitch

Sending bid responses in Compressed JSON does not require special configuration. The bidder is only required to send a `Content-Encoding: gzip` header in the responses where the compression is used.

### 1.7.2 Receiving Bid Requests from BidSwitch

In order for the Buyer to receive Compressed JSON bid requests, please contact BidSwitch support at support@bidswitch.com.

Once enabled, BidSwitch starts sending bid requests in Compressed JSON. All such bid requests carry an extra HTTP header `Content-Encoding: gzip`. It is recommended for the bidder to check for this HTTP header to distinguish between JSON and Compressed JSON bid request formats, as even when compression is enabled some Bid Requests may arrive uncompressed. In this case the `Content-Encoding: gzip` header is absent.

## 1.8 Macros

### 1.8.1 Win price macro

In order for the exchange to convey certain information to the winning bidder (e.g., the settlement price), some substitution macros can be inserted into the win notice URL. Prior to calling a win notice URL, BidSwitch will search the specified URL for any of the defined macros and replace them with the appropriate data.

---

**Note:** The substitution is simple in the sense that wherever a legal macro is found, it will be replaced without regard to syntax correctness.

---

Table 3: Win Price Macro Description

| Value | Description |
|---|---|
| *${AUCTION_PRICE}* | Settlement price for the auction. The substituted value will be defined as CPM. See the *Bid Object* (page 140) section for usage details. This macro can be used, but with restrictions per protocol version.<br>• In version 4.x, the win price macro should be used in `nurl` field, and the click tracking macro may be used in the `adm` field<br>• In version 5.x, the win price can be used in the `adm` or `nurl` field, with restrictions. See the *5.x Updating Overview* (page 24) section for details.<br>• In version 5.2, the win price can also be used in the `burl` field, with certain restrictions. See the *Using the burl Field* (page 25) section for details. |

Table 4: Valid Use of Win Price Macro Quick Reference

| | v4.x | v5.0 & v5.1 | v5.2 & v5.3 |
|---|---|---|---|
| Browser Notification | `nurl` | `nurl \| adm` | `nurl \| adm` |
| u2s or Browser Notification | `nurl` | `nurl \| adm` | `(nurl \| adm) + burl` |
| s2s notification only, specified in the Bid Request using `imp.ext.s2s_nurl = 1` | `nurl` | `nurl` | `nurl \| burl` ** |

**Note:**

- Please see the field descriptions in the *Bid Object* (page 140) section for more precise details about each, and the *5.x Updating Overview* (page 24) and *Using the burl Field* (page 25) sections for more details.
- ** As it is an s2s call only, if the the `burl` field is given, the `nurl` field is ignored. Therefore, you should set the `${AUCTION_PRICE}` macro in the field that will be called. See the *Using the burl Field* (page 31) section for complete details.

### 1.8.2 Supplier Click Tracking URL Macro

Some Suppliers require the Buyer to add a Supplier click URL macro to all creatives for click reporting. Buyers wishing to bid on such bid requests should add the click tracking macro to the creative markup in the `adm` field. The Supplier will replace the macro with the Supplier click tracking URL. If the Supplier doesn't support click tracking macros, then BidSwitch will replace the tracking macro with an empty string.

**Note:**

- This is a strict requirement if the bid request contains `ext.clktrkrq` field and is strongly recommended in other cases.
- In the case of video and native inventory, click tracking is handled on the Supplier side and the click macro is not used, as well as the `adm` field.

Table 5: Click Tracking Macro

| Value | Type |
|---|---|
| *${CLICK_URL:URLENCODE}* | A placeholder for the Supplier click tracking URL in URL encoded form. Required for bids to Suppliers that support click tracking. No more than one click tracking macro can be used in the `bid.adm` field. Only single-encoded click tracking URLs are supported. |

The Supplier click URL (if present) should be inserted before the landing page in the creative. The landing page URL should be single-escaped. For example, if the Buyer click-URL contains

```
http://dsp.com/click?bc=dnJD723&sspclick=${CLICK_URL:URLENCODE}
```

The macro is replaced and the user clicks the resulting URL

```
http://dsp.com/click?bc=dnJD723&sspclick=http%3A%2F%2Fssp.com%2Fclick%3Fic%3DbKk4
↪%26lp%3D
```

The Buyer unescapes the `sspclick` parameter and redirects to the target URL while adding the landing page at the end

```
http://ssp.com/click?ic=bKks3k4&lp=http%3A%2F%2Fadvertiser.com%2Fhomepage
```

**Note:** Some Suppliers may keep the `/` and `:` characters unencoded, thus the resulting click URL may take a form such as the following

```
http://dsp.com/click?bc=dnJD723&sspclick=http://ssp.com%2Fclick%3Fic%3DbKk4%26lp%3D
```

### 1.8.3 True Price Macro

The true price macro can used in the win notice URL. This macro will indicate whether the Bid Response was processed by True Price, and it should be used in conjunction with the `${AUCTION_PRICE}` macro.

Table 6: True Price Macro

| Value | Type |
|---|---|
| *${AUCTION_TP_MODE}* | A placeholder which BidSwitch will populate with information indicating if True Price processed the Bid Response. The field can be populated with the following values:<br>• `1` No True Price processing applied to this Bid Response. This is considered the control group used to benchmark its effectiveness<br>• `2` True Price treated this Bid Response<br>• An empty string. Indicates that the response did not go through True Price. This is returned if you used the `ext.true_price_opt_out` field to bypass True Price, see the *Bid Response Ext Object* (page 139) section for more details. |

```
# syntax to include in Bid Response
http://adserver.com/winnotice?impid=102&winprice=${AUCTION_PRICE}&trueprice=$
↪{AUCTION_TP_MODE}

# Example Return
http://adserver.com/winnotice?impid=102&winprice=4&trueprice=2
```

### 1.8.4 Total Impressions Macro

Some Suppliers support the `${total_imp}` macro in DAAST and VAST markup. For those Suppliers, BidSwitch will append this macro to the VAST impression URL, which is filled with the total audience size when the Vast or Daast URL is played.

When trading DOOH, the usage of this macro may result in a different tally than calculations based upon the `dooh.impmultiply` request field. Both parties should agree on which number to base their billing and discrepancy counts to prevent mismatches. The following Suppliers currently support this macro:

- AdScale (a.k.a Ströer)

**Example VAST Usage**

```
<Impression><![CDATA[http://x.bidswitch.net/imp/${AUCTION_PRICE}/BSWhttp_A_B_
↪Bdsp1.com_Bimp_Cprice_R_I_WAUCTION__PRICE_X/foo//${TOTAL_IMP}]]></Impression>
```

## 1.9 Sensitive Categories and Rich Media

For greater coverage of sensitive categories BidSwitch extends the standard IAB list with additional categories. These categories may be used in the following fields.

Table 7: Sensitive Categories Fields

| Bid Requests | Bid Response |
|---|---|
| • `bcat`<br>• `site.cat`<br>• `app.cat` | `seatbid.bid.cat` |

Table 8: BidSwitch Sensitive Categories

| ID | Category |
|---|---|
| BSW1 | Alcohol |
| BSW2 | Gambling |
| BSW3 | Tobacco and smoking |
| BSW4 | Firearms and weapons |
| BSW5 | Sexual & Reproductive Health |
| BSW6 | Ringtones & Downloadable |
| BSW7 | Drugs & Supplements |
| BSW8 | Get Rich Quick |
| BSW9 | Free Gifts, Quizzes, & Surveys |
| BSW10 | Nudity |
| BSW11 | Cosmetic Procedures & Body Modification |

### 1.9.1 Supported Rich Media Frameworks

Rich Media Framework information will be sent to Buyers using the `imp.iframebuster` field in bid requests, see the *Impression Object* (page 44) section for more information.

Table 9: Supported Rich Media Frameworks

| Description | Value |
|---|---|
| Any framework is accepted | ALL |
| Adcentric | ac |
| Adinterax | ad |
| Adform | af |
| Atlas | at |
| Apivid | av |
| DoubleClick | dc |
| Eyeblaster | eb |
| EyeReturn | er |
| EyeWonder | ew |
| Flashtalking | ft |
| Klipmart | km |
| Kpsule | ks |
| MediaMind | mm |
| Mediaplex | mp |
| Piximedia | pm |
| PointRoll | pr |
| Pictela | pt |
| Rockabox | rb |
| Smart Adserver | sa |
| Silence Media | sm |
| Unicast | ui |
| Undertone | ut |
| Viewpoint | vp |
| Weborama | wo |

## 1.10 Supplier Custom Categories

See the online reference page for these categories, *Supplier Custom Categories* (page 40)

## 1.11 Server-to-Server (s2s) Calls

BidSwitch calls the Buyer endpoint using a s2s call in the following cases:

- The initial call from the Supplier/Publisher side was done using a s2s call
- BidSwitch is calling the Buyer `burl`
- BidSwitch is calling the Buyer `nurl` as the BidSwitch Bid Request had the `imp.ext.s2s_nurl` field set to `1`.

An easy way to determine whether the call from BidSwitch is s2s is to check the user agent. For s2s calls it is always `"BidSwitch/1.0"`. One advantage of s2s impression calls is increased control of the

call delivery, and thus minimizing any discrepancies between partners' reporting and/or invoicing.

### 1.11.1 Expected s2s Behaviour

- BidSwitch expects HTTP `200`, `204`, or `302` responses to s2s calls, if you have recorded a valid impression.
- If BidSwitch receives a response code > `399`, i.e. *HTTP 4xx, or 5xx*, it denotes the impression as not properly recorded on the Buyer side, and will lead to discrepancies between BidSwitch and the Buyer.
- If the response code is >= `400`, or there was a timeout, BidSwitch makes 1 retry to check if a valid impression was recorded on the Buyer side.
- BidSwitch will bill based on its numbers, regardless of Buyer response. See the discrepancy-api-buy section for details about managing discrepancies.

### 1.11.2 s2s Headers

When sending s2s impression calls, BidSwitch uses the following headers.

- `User-Agent:` Indicates that the impression call is relayed by BidSwitch, this is always `BidSwitch/1.0`.
- `X-User-Agent:` Specifies the application type, operating system, software vendor or software version of the original request.
- `X-Forwarded-For:` Identifies the originating IP address.
- `X-BSW-ServerUpstream:` A value of `1` indicates that the original impression call to BidSwitch was from a server.

```
# Example of s2s headers from burl imp request
X-Forwarded-For: 77.16.215.3, 18.196.6.164
X-BSW-ServerUpstream: 1
User-Agent: BidSwitch/1.0
X-User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 11_2_1 like Mac OS X) AppleWebKit/
↪604.4.7 (KHTML, like Gecko) Mobile/15C153 [FBAN/FBIOS;FBAV/179.0.0.50.82;FBBV/
↪116150041;FBDV/iPhone8,1;FBMD/iPhone;FBSN/iOS;FBSV/11.2.1;FBSS/2;FBCR/Telenor;
↪FBID/phone;FBLC/nb_NO;FBOP/5;FBRV/0]
```

Use the following table to compare the s2s vs u2s headers BidSwitch applies in each circumstance.

Table 10: BidSwitch s2s and u2s HTTP Headers

| SSP to DSP | s2s:s2s_nurl in request or adslot | s2s:burl in response | u2s |
|---|---|---|---|
| s2s | `User-Agent: BidSwitch/1.0`<br>`X-BSW-ServerUpstream: 1` | `User-Agent: BidSwitch/1.0`<br>`X-BSW-ServerUpstream: 1`<br>`X-User-Agent:` (applied from user in most cases)<br>`X-Forwarded-For:` | **n/a**; s2s forwarded as s2s only |
| u2s | `User-Agent: BidSwitch/1.0`<br>`X-Forwarded-For:`<br>`X-User-Agent:` (applied from user) | `User-Agent: BidSwitch/1.0`<br>`X-Forwarded-For:`<br>`X-User-Agent:` (applied from user) | `X-Forwarded-For:`<br>`X-User-Agent:` **See u2s note** |

**Note:** **u2s:** In all other scenarios which the table outlines, BidSwitch applies its own headers. Except in the u2s case, where it forwards the initial u2s call without any alteration.

### Protocol

## 1.12 Bid Request v5.3

This is the top level object that is sent to the Buyer. Each bid request sent from BidSwitch to a Buyer will contain the following fields.

**Note:** Fields marked with asterisk (*) are optional.

Table 11: Bid Request Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | Unique ID of the bid request, provided by BidSwitch, for example, `"b5ba5ed2-547e-4e86-8a84-34a440dad6db"` |
| *imp* | *array of objects* | Array of objects representing the impressions offered, for more information, see the *Impression Object* (page 44) section. |

continues on next page

Table 11 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *device* | *object* | Device object with details about the device to which the impression will be delivered, for more information, see the *Device Object* (page 72) section. |
| *user* | *object* | User Object which describes the user, for more information, see the *User Object* (page 77) section.<br>**Note**: In the even that a user has opted-out under the GDPR or CCPA, this will be passed as an empty object, e.g. `"user":{}` |
| *tmax* | *integer* | Maximum time in milliseconds the exchange allows for bids to be received to avoid timeout, including internet latency, for example, `120`. |
| *cur* | *array of strings* | Array of allowed currencies for bids on this bid request using ISO-4217[20] alpha codes, for example, `["USD", "EUR"]` |
| *ext* | *object* | Ext Object used for Supplier specific properties, for more information, see the *Ext Object* (page 100) section. |
| *at** | *integer* | Auction type, the default value is `2`.<br>• `1`: the first price auction.<br>• `2`: the second price auction. |
| *source** | *object* | Indicates the entity responsible for the final impression sale decision, see the *Source Object* (page 93). |
| *site** | *object* | The *Site Object* (page 84) describing the site. One of these objects should be present in the request: `site`, `app`, `ext.tv`, or `ext.dooh`. |
| *app** | *object* | The *App Object* (page 86) describing the mobile application. One of these objects should be present in the request: `site`, `app`, `ext.tv`, or `ext.dooh`. |
| *bcat** | *array of strings* | Blocked Advertiser Categories, using the IAB taxonomy, and extended with additional sensitive categories listed in the *Sensitive Categories and Rich Media* (page 39) section. Creatives belonging to at least one of the listed categories are not permitted for bidding in the current bid request, for example `["IAB10-1", "IAB25", "BSW3"]` |
| *badv** | *array of strings* | Array of strings of blocked top-level domains of advertisers, for example, `["mysite.com", "mysite2.com"]` |
| *bapp** | *array of strings* | A list of applications blocked from being advertised, specified using their their platform-specific identifiers, i.e. the app bundle ID. On Android, these should be app bundle and on iOS numeric IDs e.g. `["com.app.example", "012987652"]` |

continues on next page

Table 11 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *wseat** | *array of strings* | An array of Buyer seats allowed to bid on this auction, for example `[58, 61, 99]`. If this field is present, the specified seat IDs may be supplied using BidSwitch or Supplier taxonomy. A bid request may contain multiple seat IDs using the Supplier taxonomy. The bid response should contain the appropriate seat value corresponding to one of the values of the `wseat` field, for more details see the *Seat Bid Object* (page 140) section. |
| *bseat** | *array of strings* | An array of Buyer seats disallowed to bid on this auction, for example `["58", "61", "99"]`. If this field is present, it can be assumed that the Seat IDs refer to agency buyers, because if the BidSwitch Buyer ID is set in the original Supplier request, it is not sent to the DSP. The `wseat` field takes precedence, so if the same value is in both fields the request is sent to the Buyer. |
| *allimps** | *integer* | A flag to indicate if the Supplier 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.<br>• `0` = no or unknown<br>• `1` = yes, the impressions offered represent all that are available. |
| *regs** | *object* | A regulations object that specifies any industry, legal, or governmental regulations in force for this request, for more information, see the *Regs Object* (page 104) section. |

### 1.12.1 Impression Object

**Note:** Fields marked with asterisk (*) are optional.

Table 12: Impression Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | ID of the impression being shown, unique within the bid request, for example `"1"` |
| *banner** | *object* | The *Banner Object* (page 50) describes the ad properties. Required for banner impressions. |
| *video** | *object* | The *Video Object* (page 52) describes the ad properties. Required for video impressions. |

continues on next page

[20] https://www.iso.org/iso-4217-currency-codes.html

Table 12 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *audio** | *object* | The *Audio Object* (page 58) describes the ad properties. Required for audio impressions. |
| *native** | *object* | The *Native Object* (page 62) describes the ad properties. Required for native impressions. |
| *bidfloor** | *float* | Bid floor in CPM as set by the Supplier, for example, `0.01080` |
| *bidfloorcur** | *string* | Bid floor currency specified using ISO-4217 alpha codes, for example, `"USD"`. |
| *instl** | *integer* | Specifies if the ad is an interstitial.<br>• `0` = not interstitial, the default value.<br>• `1` = the ad is interstitial or full screen |
| *tagid** | *string* | Identifier for specific ad placement or ad tag that was used to initiate the auction. This value is prefixed with the exchange name followed by underscore, for example, if Rubicon sends a bid request for a placement ID "43256", the value becomes `"rubicon_43256"` |
| *secure** | *integer* | Specifies if the page is SSL compliant:<br>• `0`: for insecure pages, the default value.<br>• `1`: for secure pages. Creative assets for secure pages should be SSL-compliant. |
| *iframebuster** | *array of strings* | Array of names of supported iframe busters, for example, `["dc", "rb"]`, for more information, see the *Supported Rich Media Frameworks* (page 39) section. |
| *pmp** | *object* | The *Private Marketplace Object* (page 92), used for direct deals between Buyers and Suppliers. |
| *displaymanager** | *string* | Name of the ad mediation partner, SDK technology, or native player responsible for rendering the ad (typically video or mobile), for example, `"SOMA"` |
| *displaymanagerver** | *string* | Version of the ad mediation partner, SDK technology, or native player responsible for rendering the ad (typically video or mobile), for example, `"1.1"` |
| *exp** | *integer* | Impression expiry timeout, in seconds, for example, `"300"`. An impression will be considered expired if it is registered later than `imp.exp` seconds after the auction. |
| *metric** | *array of objects* | The object that is associated with an impression as an array of metrics, see the *Metric Object* (page 92) section. |
| *ext** | *object* | See the *Impression Ext* (page 46) section. |

### Example Impression Object JSON

```
{
  "imp":[
    {
      "id":"1",
      "metric":[
        {
          "type":"viewability",
          "value":0.85
        }
      ],
      "bidfloor":0.426,
      "banner":{
        "w":300,
        "h":250,
        "pos":1,
        "topframe":0,
        "expdir":[
          1,
          3
        ]
      }
    }
  ]
}
```

### Impression Ext

Table 13: Impression Extension Properties

| Value | Type | Description |
|---|---|---|
| *wopv* | *str* | Passes the WhiteOps MediaGuard Prediction ID, e.g. "abc-123" |
| *google* | *object* | See *Impression Ext Google Object* (page 49) |
| *yieldone** | *object* | See *Impression Ext Yieldone Object* (page 50) |
| *skadn** | *object* | Apple Ad Network Object, this will be used to pass app data from iOS 14 and newer releases. See *SkAdNetwork Extension* (page 48) |
| *inventory_class** | *integer* | Inventory class according to the Supplier classification. The Supplier classification is maintained by each Buyer and is not a BidSwitch list, for example `1` |

continues on next page

Table 13 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *notification_type** | *int* | Indicates the Supplier's supported notification type(s):<br>• `1` The SSP only sends browser (u2s) notifications<br>• `2` The SSP can only send Server-to-Server (s2s) notification<br>• `3` The SSP can response with either u2s or s2s depending on the Buyer response.<br>You can read more about notification types in the *Server-to-Server (s2s) Calls* (page 40) section |
| viewability* | integer | Deprecated since version 5.2.<br>Replaced by using the *Metric Object* (page 92)<br>This is an estimate of the likelihood that this slot will be viewable by the end user. Estimation is based on Supplier-specific data (e.g. historical viewability data, real-time data etc.). Expressed as a percentage in the range of [0, 100]. If the field is not present in the bid request then the viewability estimate is not available. For example, `100` |
| *ssai** | *int* | Indicates if server-side ad insertion (e.g., stitching an ad into an audio or video stream) is in use and the impact of this on asset and tracker retrieval. It can take the following values:<br>• `0` = status unknown<br>• `1` = all client-side (i.e., not server-side)<br>• `2` = assets stitched server-side but tracking pixels fired client-side<br>• `3` = all server-side. |
| *s2s_nurl** | integer | `1` indicates an impression object that only uses a server-to-server win notification for won auctions. This has the following implications for the `${AUCTION_PRICE}` macro:<br>• The `adm` field cannot contain the `${AUCTION_PRICE}`. For v5.1 and v5.2 the `nurl` field should contain the macro.<br>• As of v5.2, the `burl` field should contain the `${AUCTION_PRICE}`<br>• BidSwitch expects that `nurl` calls for Bid Requests with `s2s_nurl` set to `1` should return a HTTP status `200`, `204`, or `30x`, see the *Server-to-Server (s2s) Calls* (page 40) section for more information.<br>See the *5.x Updating Overview* (page 24) and *burl Field Overview* (page 30) sections for more information. |

### SkAdNetwork Extension

Table 14: skadn Extension Properties

| Value | Type | Description |
|---|---|---|
| *version** | *str* | Version of skadnetwork supported. Always `"2.0"` or higher. Dependent on both the OS version and the SDK version., e.g. `"2.0"` https://developer.apple.com/documentation/storekit/skadnetwork |
| *versions** | *array of strings* | An array of strings containing the supported skadnetwork versions. Always `"2.0"` or higher. Dependent on both the OS version and the SDK version. |
| *sourceapp** | *str* | ID of publisher app in Apple's App Store. Should match "app.bundle" |
| *skadnetids** | *array of string* | A subset of SKAdNetworkItem entries in the publisher app's `info.plist` that are relevant to the DSP, e.g. `["cDkw7geQsH.skadnetwork", "qyJfv329m4.skadnetwork"]` https://developer.apple.com/documentation/bundleresources/information_property_list |
| *skadnetlist** | *object* | An object containing the IAB Tech Lab (IABTL) list definition. See the *skadnetlist object* (page 49) table for details. You can also find out more about the IABTL List from their release blog post[21] and at https://tools.iabtechlab.com/skadnetwork |

[21] https://iabtechlab.com/blog/register-now-for-iab-tech-lab-skadnetwork-id-list/

### skadnetlist object

Table 15: skadnetlist Properties

| Value | Type | Description |
|---|---|---|
| *max** | *integer* | A list of IABTL IDs containing the max entry ID on the list up to which you wish to include. The skadnetids associated with all IABTL IDs numerically lower than this are included as subset of SKAdNetworkItem entries in the publisher app's `info.plist` that are relevant to the DSP, e.g. `42`. |
| *excl** | *array of integers* | A list of IABTL registration IDs to be excluded, i.e. those numerically lower than the max value but which should not be included, e.g. `[12, 14]` |
| *addl** | *array of strings* | A list of raw lowercase SKAdNetworkItem entries in the publisher app's `info.plist` that are relevant to the DSP, e.g. `["cDkw7geQsH.skadnetwork", "qyJfv329m4.skadnetwork"]` https://developer.apple.com/documentation/bundleresources/information_property_list **Note:** The intention of this field is to replace the `skadn.skadnetids` field, it is also recommended that this list not exceed 10. |

### Impression Ext Google Object

Table 16: Google Impression Extension Properties

| Value | Type | Description |
|---|---|---|
| *excluded_attribute** | *array of integers* | List of excluded creative attributes as defined by Google, for example, `[70, 28, 30, 32, 22]` |
| *allowed_vendor_type** | *array of integers* | List of allowed vendor types as defined by Google. |

**Note:** If the `site.publisher.id` or `app.publisher.id` field value equals "`google_1`" then the vendors listed at the following link are also allowed to bid:

#### Impression Ext Yieldone Object

Table 17: Yieldone Impression Extension Properties

| Value | Type | Description |
|---|---|---|
| *allowed_creative_types* | *array of strings* | List of allowed creative types as defined by YieldOne, for example, `["HTML", "FLASH"]` |
| *allowed_creative_category_id** | *array of integers* | List of allowed creative categories as defined by YieldOne, for example, `[70, 71, 72]` |
| *cat** | *array of integers* | List of site categories as defined by YieldOne, for example, `[5, 16]` |
| *inventory_class** | *integer* | Inventory class according to the YieldOne classification. Deprecated since version 2.4: Use `imp.ext.inventory_class` instead. |

### 1.12.2 Banner Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 18: Banner Object Properties

| Value | Type | Description |
|---|---|---|
| *w* | integer | Width of the impression in pixels, for example, `300` |
| *h* | *integer* | Height of the impression in pixels, for example `250` |
| *battr** | *array of integers* | Blocked creative attributes as defined in the OpenRTB protocol, for example, `[1, 23]` |
| *btype* | *array of integers* | Blocked banner ad types as defined in the OpenRTB protocol, for example, `[4, 21]` |
| *pos** | *integer* | Ad Position as defined in the OpenRTB protocol, for example, `1` |
| *topframe** | *integer* | Indicates if the banner is in the top frame as opposed to an iframe.<br>• `0` = no<br>• `1` = yes. |
| *mimes** | *array of strings* | Specifies the content MIME types supported, common MIME types include `"text/html"`, `"application/x-shockwave-flash"`, and `"image/gif"`. For example: `[ "video/mp4", "image/jpg"]` |
| *expdir** | *array of integers* | Possible expansion directions for an expandable ad, for example, `[2,5]`. This can take the following values:<br>• `1`: Left<br>• `2`: Right<br>• `3`: Up<br>• `4`: Down<br>• `5`: Full screen<br>If the field is not present, expandable creatives are not allowed. |
| *format** | *array or objects* | An array of format objects, see *Format Object* (page 52), denoting the alternative sizes that may be used for bidding. If one of the alternative ad sizes is used in the bid response, then the `seatbid.bid.h` and `seatbid.bid.w` fields are required in the bid response. |
| *api** | *array of integers* | List of supported API frameworks for this impression as defined in the OpenRTB, for example `[3, 5]`. If an API is not explicitly listed, it is assumed not to be supported. |

### Banner JSON Example

```
{
   "banner":{
      "w":300,
      "h":250,
      "pos":1,
      "topframe":0,
```

(continues on next page)

(continued from previous page)

```
      "btype":[
         2,
         3
      ],
      "mimes":[
         "text/html",
         "application/x-shockwave-flash"
      ],
      "format":[
         {
            "h":50,
            "w":300
         }
      ]
   }
}
```

### 1.12.3 Format Object

Table 19: Format Object Properties

| Value | Type | Description |
|---|---|---|
| *h* | *integer* | Height of the impression in pixels, for example 500 |
| *w* | *integer* | Width of the impression in pixels, for example 340 |

```
{
   "format":[{
      "w":300,
      "h":250
   }]
}
```

### 1.12.4 Video Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 20: Video Object Properties

| Value | Type | Description |
|---|---|---|
| *mimes* | *array of strings* | Content MIME types supported.<br>**Note:** For Google bids all the specified MIME type media files are required in the VAST response, for example, `["video/mpeg", "video/mp4"]` |
| *minduration* | *integer* | Minimum video ad duration in seconds, for example, `2` |
| *maxduration* | *integer* | Maximum video ad duration in seconds, for example, `15` |
| *linearity** | *integer* | Indicates if the impression must be linear or nonlinear, for example, `1`. If none is specified, it is assumed all are allowed<br>• `1`: Linear/In-stream<br>• `2`: Non-Linear/Overlay |
| *placement** | *integer* | Placement type for the impression, for example `2`. This can take the following values:<br>• `1`: In-stream. Played before, during or after the streaming video content that the consumer has requested (e.g., Pre-roll, Mid-roll, Post-roll).<br>• `2`: In-banner. Exists within a web banner that leverages the banner space to deliver a video experience as opposed to another static or rich media format. The format relies on the existence of display ad inventory on the page for its delivery.<br>• `3`: In-article. Loads and plays dynamically between paragraphs of editorial content; existing as a standalone branded message.<br>• `4`: In-feed. Found in content, social, or product feeds.<br>• `5`: Interstitial/Slider/Floating. Covers the entire or a portion of screen area, but is always on screen while displayed (i.e. cannot be scrolled out of view). Note that a full-screen interstitial (e.g., in mobile) can be distinguished from a floating/slider unit by the `imp.instl` field. |
| *playbackend** | *integer* | The event that causes playback to end, for example `2`. This field can take the following values:<br>• `1`: On Video Completion or when Terminated by User.<br>• `2`: On Leaving Viewport or when Terminated by User.<br>• `3`: On Leaving Viewport Continues as a Floating/Slider Unit until Video Completion or when Terminated by User. |
| *protocols* | *array of integers* | Accepted video bid response protocols as defined in OpenRTB, for example `[2,5]` |
| *pos** | *integer* | Ad Position as defined in OpenRTB, for example `1` |
| *w** | *integer* | Width of the player in pixels, for example, `600` |
| *h** | *integer* | Height of the player in pixels, for example `400` |

continues on next page

Table 20 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *startdelay** | *integer* | Indicates the start delay in seconds. If the start delay value is greater than 0, then the position is mid-roll and the value indicates the start delay.<br>• `> 0`: Mid-Roll (value indicates start delay in second)<br>• `0`: Pre-roll<br>• `-1`: Generic mid-roll<br>• `-2`: Generic post-roll |
| *battr** | *array of integers* | Blocked creative attributes as defined in OpenRTB, for example, `[6]` |
| *minbitrate** | *integer* | Minimum bit rate in Kbps, for example `680` |
| *maxbitrate** | *integer* | Maximum bit rate in Kbps, for example `990` |
| *api** | *array of integers* | List of supported API frameworks for this impression as defined in OpenRTB, for example, `[1,2]`. If an API is not explicitly listed, it is assumed not to be supported. |
| *maxextended** | *integer* | Maximum extended video ad duration if extension is allowed.<br>• Blank or `0`, extension is not allowed.<br>• `-1`, extension is allowed, and there is no time limit imposed.<br>• Greater than `0`, then the value represents the number of seconds of extended play supported beyond the `maxduration` value. |
| *boxingallowed** | *integer* | Indicates if letter-boxing of 4:3 content into a 16:9 window is allowed:<br>• 0 = no<br>• 1 = yes. |
| *playbackmethod** | *array of integers* | Allowed playback methods as defined in the OpenRTB, for example `[1, 2]`. If none are specified, it is assumed all are allowed. |
| *delivery** | *array of integers* | Supported delivery methods (e.g., streaming, progressive) as defined in OpenRTB. If none specified, assume all are supported, for example, `[1, 2]` |
| *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, for example, `2`. |
| *companionad** | *object array* | Array of Banner objects if companion ads are available. See the *Banner Object* (page 50) section for more information. |

continues on next page

Table 20 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *companion-type** | *array of integers* | List of allowed companion ad types, for example `[1, 2]` Possible values:<br>• `1`: Static Resource<br>• `2`: HTML Resource<br>• `3`: iframe Resource |
| *ext** | *object* | See the *Video Ext Object* (page 56) section |
| *skip** | *integer* | Indicates if the player will allow the video to be skipped, where 0 = no, 1 = yes. |
| *skipmin** | *integer* | Videos of a total duration greater than this value (seconds) can be skippable; only applicable if the ad is skippable. |
| *skipafter** | *integer* | Number of seconds a video must play before skipping is enabled; only applicable if the ad is skippable. |

### Video Ext Object

Table 21: Video Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *skippable** | *integer* | Indicates whether the video ad may be skippable (i.e. contain a skip button). The values are:<br>• `0`: allow skippable<br>• `1`: require skippable<br>• `2`: deny skippable |
| *player_type** | *integer* | Video player type based on detected player size:<br>• `1`: Small player ( Less than 300px)<br>• `2`: Medium player ( between 300px and 600px)<br>• `3`: Large player ( greater than 600px) |
| *outstream** | *integer* | Indicates that the video is outstream, for example `1` |
| *rewarded** | *integer* | Indicates whether the ad is being rendered as part of a rewarded / incentivised user experience, where:<br>• `0` = non-rewarded<br>• `1` = rewarded<br>• If omitted, non-rewarded can be assumed |
| *vast_url_rq** | *integer* | Used to specify if the `bid.ext.vast_url` field is required in the bid response. It is recommended that you use this field to pass the VAST URL in all video responses.<br>• `1`: Indicates the `bid.ext.vast_url` field is required in the bid response.<br>• `0`: If this field is missing, or set as not required using `0`, Buyers should still pass the VAST URL in this field as it works for all video requests.<br>Only if it is deemed extremely necessary should you pass the VAST URL in the `bid.nurl` field. The price macro will be replaced in that field as usual. In this scenario BidSwitch will not send an impression notification as the VAST impression event should be used for that. |

### Video Object Example

```
{
   "id":"1",
   "bidfloor":0.03,
   "video":{
      "w":640,
      "h":480,
      "pos":1,
      "startdelay":0,
      "minduration":5,
      "maxduration":30,
      "maxextended":30,
      "minbitrate":300,
      "maxbitrate":1500,
      "skip":1,
      "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
      ],
      "btype": [
         4,
         21
      ],
      "battr":[
         13,
         14
```

(continues on next page)

(continued from previous page)

```
      ],
      "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
      ]
   }
}
```

### 1.12.5 Audio Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 22: Audio Object Properties

| Value | Type | Description |
|---|---|---|
| *mimes* | *array of strings* | Content MIME types supported, for example `["audio/mp4", "audio/mpeg"]` |
| *minduration* | *integer* | Minimum audio ad duration in seconds, for example, `2` |
| *maxduration* | *integer* | Maximum audio ad duration in seconds, for example, `15` |
| *protocols* | *array of integers* | Accepted audio bid response protocols as defined in OpenRTB, for example `[9,10]` |
| *startdelay** | *integer* | Indicates the start delay in seconds, or generic values below:<br>• `0`: Pre-roll<br>• `-1`: Generic mid-roll<br>• `-2`: Generic post-roll |
| *battr** | *array of integers* | Blocked creative attributes as defined in OpenRTB, for example, `[6]` |
| *minbitrate** | *integer* | Minimum bit rate in Kbps, for example `32` |
| *maxbitrate** | *integer* | Maximum bit rate in Kbps, for example `320` |
| *api** | *array of integers* | List of supported API frameworks for this impression as defined in the OpenRTB guide, for example, `[1,2]`. If an API is not explicitly listed, it is assumed not to be supported. |
| *maxextended** | *integer* | Maximum extended audio ad duration if extension is allowed.<br>• Blank or `0`, extension is not allowed.<br>• `-1`, extension is allowed, and there is no time limit imposed.<br>• Greater than `0`, then the value represents the number of seconds of extended play supported beyond the *maxduration* value. |
| *delivery** | *array of integers* | Supported delivery methods (e.g., streaming, progressive) as defined in OpenRTB. If none specified, assume all are supported, for example, `[1, 2]` |
| *maxseq** | *integer* | The maximum number of ads that can be played in an ad pod, for example, `1` |
| *feed** | *integer* | Type of audio feed, for example, `1` |
| *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, for example, `2` |
| *stitched** | *integer* | Indicates if the ad is stitched with audio content or delivered independently, for example, `1` |
| *nvol** | *integer* | Volume normalization mode as defined in OpenRTB, for example, `1` |
| *companionad** | *array of objects* | Array of Banner objects if companion ads are available. See the *Banner Object* (page 50) section for more information. |

continues on next page

Table 22 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *companion-type** | *array of integers* | Supported DAAST companion ad types, for example `[1, 2]` Possible values:<br>• `1`: Static Resource<br>• `2`: HTML Resource<br>• `3`: iframe Resource |
| *ext** | object | Audio extension object |

### Extension Object

Table 23: Audio extension Object Properties

| Value | Type | Description |
|---|---|---|
| *format* | *int* | Specifies in which field BidSwitch expects the creative URL to be placed in the bid response. This field exists due to how some Suppliers handle audio as audio and others handle it like video. Takes the following values.<br>• `1` for `seatbid.bid.ext.daast_url`<br>• `2` for `seatbid.bid.ext.vast_url` For Rubicon we'll always have format set to `2`.<br>• `3` for any, either VAST URL or DAAST URL. |

### Audio Object Example

```
{
"id":"1",
"bidfloor":0.03,
"audio":{
  "startdelay":0,
  "minduration":5,
  "maxduration":30,
  "maxextended":30,
  "minbitrate":300,
  "maxbitrate":1500,
  "api":[
    1,
    2
  ],
  "protocols":[
    9,
    10
  ],
```

(continues on next page)

(continued from previous page)


```
    "mimes":[
      "audio/aac",
      "audio/mp4",
      "audio/mpeg"
    ],
    "delivery":[
      2
    ],
    "btype": [
      4,
      21
    ],
    "battr": [
      13,
      14
    ],
    "companionad":[
      {
        "id":"1234567893-1",
        "w":300,
        "h":250,
        "pos":1,
        "battr":[
          13,
          14
        ]
      },
      {
        "id":"1234567893-2",
        "w":728,
        "h":90,
        "pos":1,
        "battr":[
          13,
          14
        ]
      }
    ],
    "companiontype":[
      1,
      2
    ]
  }
}
```

### 1.12.6 Native Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 24: Native Object

| Value | Type | Description |
|---|---|---|
| *request* | *object* | Contains the *Native Request Object* (page 64) object. |
| *battr** | *array of integers* | Blocked creative attributes as defined in OpenRTB., for example `[1, 3]` |
| *api** | *array of integers* | List of supported API frameworks for this impression as defined in OpenRTB, for example `[2,3,5]`. If an API is not explicitly listed, it is assumed not to be supported. |
| *ext** | *object* | Native Extension object, see *Native Extension Object* (page 62). |

#### Native Extension Object

Table 25: Native Request Extension Object

| Value | Type | Description |
|---|---|---|
| *triplelift** | *object* | Object for including TripleLift specific information. |
| *adchoicesurl_required** | *int* | (**Optional**) Set to `1` if the Supplier requires the Buyer to respond with an Ad Choices URL. This is a legal requirement for Japanese inventory. See *Native Ext Object* (page 151) section for response details. |

#### Native Request Extension Triplelift Object

Table 26: Native Request Extension TripleLift Object

| Value | Type | Description |
|---|---|---|
| *formats** | *array of integers* | Types supported by the particular placement according to the TripleLift classification. See the TripleLift Spec here https://github.com/triplelift/documentation/blob/master/ortb2.3.tl.md#231-triplelift-object |

### Native Object Example

```
{
  "native":{
    "ext":{
      "triplelift":{
        "formats":[
          1,
          2
        ]
      }
    },
    "request":{
      "ver":"1.2",
      "layout":1,
      "adunit":4,
      "assets":[
        {
          "id":1,
          "required":1,
          "title":{
            "len":25
          }
        },
        {
          "id":7,
          "video":{
            "protocols": [
              2,
              5
            ],
            "mimes":[
              "video/mpeg",
              "video/mp4"
            ],
            "minduration":2,
            "maxduration":2,
            "ext":{
              "playbackmethod":[
                1,
                2
              ]
            }
          },
          "required":1
        }
```

(continues on next page)

(continued from previous page)

```
      ]
    },
    "api":[
      3
    ],
    "battr":[
      13,
      14
    ]
  }
}
```

### 1.12.7 Native Request Object

**Note:** Fields marked with asterisk (*) are optional.

Table 27: Native Request Object

| Value | Type | Description |
|---|---|---|
| *ver** | *string* | Version of the Native Markup in use, for example, `"1"` |
| *layout** | *integer* | The Layout ID of the native ad unit as described in OpenRTB Native specification, for example, `3` |
| *adunit** | *integer* | The Ad unit ID of the native ad unit as described in OpenRTB Native specification. |
| *plcmttype** | *integer* | The design/format/layout of the ad unit being offered. See the *Native Placement Type* (page 67) for a list of supported placement types |
| *plcmtcnt** | *integer* | The number of identical placements in this Layout, for example, `1` |
| *seq** | *integer* | 0 for the first ad, 1 for the second ad, and so on. This is not the sequence number of the content in the stream. |
| *assets* | *array of objects* | An array of Asset Objects. Any bid must comply with this array of elements. See the *Native Asset Object* (page 65) section for more details. |
| *privacy** | *integer* | Set to `1` when the native ad supports buyer-specific privacy notices. Set to `0` (or leave absent) when the native ad does not support custom privacy links or if support is unknown. |
| *context** | *integer* | The context in which the ad appears. See *Context Type Description* (page 67) |
| *contextsubtype** | *integer* | A more detailed context in which the ad appears. See *Context SubType ID Description* (page 68) |

continues on next page

Table 27 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *eventtrackers** | *array of objects* | Specifies what type of event tracking is supported, see *Event Tracker Request Object* (page 68) |

### Native Asset Object

Table 28: Native Asset Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *integer* | Unique asset id, for example 2 |
| *required** | *integer* | Set to 1 if asset is required (exchange will not accept a bid without it), default is 0. |
| *title* ** | *object* | Native title object, see the *Native Asset Title Object* (page 65) for more details. |
| *img* ** | *object* | Native image object, see the *Native Asset Image Object* (page 65) for more details. |
| *video* ** | *object* | Native video object, see the *Native Asset Video Object* (page 66) for more details. |
| *data* ** | *object* | Native asset data object, see the *Native Asset Data Object* (page 67) section for more details. |

**Note:** (**) There may be exactly one of the fields marked with double asterisk in one asset object.

### Native Asset Title Object

Table 29: Native Asset Title Object

| Value | Type | Description |
|---|---|---|
| *len* | integer | Maximum length of the text in the title element, for example, 30 |

### Native Asset Image Object

The image asset object may contain the exact image size, the minimum image size, or both. If the exact image size (w/h) or the minimum image size (wmin/hmin) is specified in the bid request, then the image asset in the bid response should specify the values.

- The size of the image should be equal to or larger than the minimum specified
- The image asset in the bid response should contain the `w` and `h` fields, see the *Native Assets Image Object* (page 152).

Table 30: Native Asset Image Object

| Value | Type | Description |
|---|---|---|
| *type** | *integer* | Image asset type, for example `3`. Takes the following values:<br>• `1` Icon<br>• `2` Logo (Logo image for the brand/app)<br>• `3` Main (Large image preview for the ad) |
| *w** | *integer* | Width of the image in pixels, for example, `300` |
| *wmin** | *integer* | The minimum requested width of the image in pixels, for example, `100` |
| *h** | *integer* | Height of the image in pixels, for example, `250` |
| *hmin** | *integer* | The minimum requested height of the image in pixels, for example, `100` |
| *mimes** | *array of strings* | Whitelist of content MIME types supported, for example, `["image/gif"]` If blank, assume all types are allowed. |

### Native Asset Video Object

Table 31: Native Asset Video Object

| Value | Type | Description |
|---|---|---|
| *mimes* | *array of strings* | Content MIME types supported, for example, `["video/mpeg", "video/mp4"]` |
| *minduration* | *integer* | Minimum video ad duration in seconds, for example, `2` |
| *maxduration* | *integer* | Maximum video ad duration in seconds, for example `15` |
| *protocols* | *array of integers* | Accepted video bid response protocols as defined in OpenRTB, for example, `[2,5]` |
| *ext** | *object* | *Native Asset Video Object Extension* (page 66) |

### Native Asset Video Object Extension

Table 32: Native Asset Video Object Extension Properties

| Value | Type | Description |
|---|---|---|
| *playback-method** | *array of integers* | Allowed playback methods as defined in the OpenRTB, for example `[1, 2]`. If none are specified, it is assumed all are allowed. |

### Native Asset Data Object

Table 33: Native Asset Data Object

| Value | Type | Description |
|---|---|---|
| *type* | *integer* | Data asset type as described in OpenRTB Native specification, for example, `1` |
| *len** | *integer* | Maximum length of the text in the element's response, for example, `25` |

### Native Placement Type

Table 34: Native Placement Type Options

| Value | Description |
|---|---|
| 1 | In the feed of content, for example as an item inside the organic feed/grid/listing/carousel. |
| 2 | In the atomic unit of the content, i.e. in the article page or single image page |
| 3 | Outside the core content, for example in the ads section on the right rail, as a banner-style placement near the content, etc. |
| 4 | Recommendation widget, most commonly presented below the article content. |
| 500+ | To be defined by the exchange |

### Context Type Description

Table 35: Context Options

| Value | Description |
|---|---|
| 1 | Content-centric context such as newsfeed, article, image gallery, video gallery, or similar |
| 2 | Social-centric context such as social network feed, email, chat, or similar |
| 3 | Product context such as product listings, details, recommendations, reviews, or similar |
| 500+ | To be defined by the exchange |

### Context SubType ID Description

Table 36: Subtype Options

| Value | Description |
|---|---|
| 10 | General or mixed content |
| 11 | Primarily article content (which of course could include images, etc as part of the article) |
| 12 | Primarily video content |
| 13 | Primarily audio content |
| 14 | Primarily image content |
| 15 | User-generated content, e.g. forums, comments, etc |
| 20 | General social content such as a general social network |
| 21 | Primarily email content |
| 22 | Primarily chat/IM content |
| 30 | Content focused on selling products, whether digital or physical |
| 31 | Application store/marketplace |
| 32 | Product reviews site primarily (which may sell product secondarily) |
| 500+ | To be defined by the exchange |

### Event Tracker Request Object

Table 37: Event Tracker Request Object

| Value | Type | Description |
|---|---|---|
| *event* | *integer* | Type of event available for tracking. See the *Event Tracking Types* (page 69) |
| *methods* | *array of integers* | Array of the types of tracking available for the given event. See the *Event Tracking Methods* (page 69) table |
| *ext** | *object* | This object is a placeholder that may contain custom JSON agreed to by the parties to support flexibility beyond the standard defined in this specification |

### Event Tracking Types

Table 38: Event Tracking Types

| Value | Type | Description |
|---|---|---|
| 1 | *Impression* | Impression |
| 2 | *viewable-mrc50* | Visible impression using MRC definition at 50% in view for 1 second. |
| 3 | *viewable-mrc100* | Visible impression using MRC definition at 100% in view for 1 second, i.e. GroupM standard |
| 4 | *viewable-video50* | Visible impression for video using MRC definition at 50% in view for 2 seconds. |
| 500+ | *exchange specific* | |

### Event Tracking Methods

Table 39: Event Tracking Methods

| Value | Type | Description |
|---|---|---|
| 1 | *img* | Image-pixel tracking – The URL provided in the response will be inserted as a 1x1 pixel at the time of the event. |
| 2 | *js* | Javascript-based tracking – The URL provided in the response will be inserted as a js tag at the time of the event. |
| 500+ | *exchange specific* | Could include custom measurement companies such as Moat, DoubleVerify, IAS, etc – in this case additional elements will often be passed. |

### Native Request Object Example

```
{
   "native":{
      "request":{
         "plcmtcnt":1,
         "plcmttype":2,
         "privacy":1,
         "context":1,
         "contextsubtype":12,
         "eventtrackers":[
            {
               "event":1,
               "methods":[
                  1,
                  2
```

(continues on next page)

(continued from previous page)

```
            ]
         },
         {
            "event":2,
            "methods":[
               1
            ]
         }
      ],
      "assets":[
         {
            "id":1,
            "data":{
               "type":12
            },
            "required":1
         },
         {
            "title":{
               "len":50
            },
            "id":2,
            "required":1
         },
         {
            "id":3,
            "img":{
               "w":80,
               "h":80,
               "type":1
            },
            "required":1
         },
         {
            "id":4,
            "img":{
               "w":1200,
               "h":627,
               "type":3
            },
            "required":1
         },
         {
            "data":{
               "type":3
```

(continues on next page)

(continued from previous page)


```
                },
                "id":5,
                "required":0
            },
            {
                "id":6,
                "data":{
                    "len":100,
                    "type":2
                },
                "required":1
            },
            {
                "id":7,
                "video":{
                    "mimes":[
                        "video/mpeg",
                        "video/mp4"
                    ],
                    "minduration":2,
                    "protocols":[
                        2,
                        5
                    ],
                    "maxduration":2,
                    "ext":{
                        "playbackmethod":[
                            1,
                            2
                        ]
                    }
                },
                "required":1
            }
        ],
        "ver":"1.2"
    }
  }
}
```

### 1.12.8 Device Object

**Note:** Fields marked with an asterisk (*) are optional.

In the even that a user has opted-out under the GDPR or CCPA, the following data will be removed from the request, `device.ip` or `ipv6`, and `device.ifa`

Table 40: Device Object Properties

| Value | Type | Description |
|---|---|---|
| *geo* | *object* | Geo Object as derived from the device's location services, or supplied by the Supplier if the device IP is missing. For more information, see the *Geo Object* (page 76) section. |
| *ip* | *string* | Specifies the IPv4 address closest to the device.<br>• **Semi-Required** One of IP or IPv6 required<br>• Required for almost all requests, any containing invalid IP addresses will be discarded based on IP Validation e.g. `87.224.77.0`<br>• **Note:** This field is only optional for Connected TV inventory |
| *ipv6* | *string* | **Semi-Required** One of IP or IPv6 required. IP address in IPv6, for example `fe80:0:0:0:200:f8ff:fe21:67cf` |
| *ua** | string | Browser or application user agent string, for example, `"Mozilla/5.0 (Windows NT 6.3; WOW64; rv:35.0)Gecko/ 20100101Firefox/35.0"` |
| *language** | *string* | Alpha-2/ISO 639-1 code of browser language, for example, `en` |
| *carrier** | *string* | Carrier or ISP derived from the IP address, for example, `WIFI` |
| *connection-type** | *integer* | Connection type as defined in OpenRTB, for example, `2` |
| *didsha1** | *string* | Hardware device ID (e.g., IMEI); hashed via SHA1, for example, `CCF6DC12B98AEB2346AFE1BEE7860DF01FDE158B`<br>Deprecated since version 5.3. |
| didmd5* | string | Hardware device ID (e.g., IMEI); hashed via MD5. `93D05D4D69DEE2BC6645D9F0A0C1938C`<br>Deprecated since version 5.3. |
| *dpidsha1** | *string* | Platform device ID (e.g., Android ID); hashed via SHA1, for example, `CCF6DC12B98AEB2346AFE1BEE7860DF01FDE158B` |
| *dpidmd5** | string | Platform device ID (e.g., Android ID); hashed via MD5, for example, `93D05D4D69DEE2BC6645D9F0A0C1938C` |

continues on next page

Table 40 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *ifa*** | *string* | The ID for Advertisers (IFA) in clear text (i.e. not hashed), for example Apple's IDFA or Android's Advertising ID. **Note:** This field is required for in-app requests.<br>The Apple IDFA is usually uppercase, and the Android Advertiser ID is usually lowercase. For example:<br>• Android `"035911ea-467d-4056-903b-65cf44f5633b"`<br>• iOS `"30255BCE-4CDA-4F62-91DC-4758FDFF8512"`<br>• iOS 14 `"00000000-0000-0000-0000-000000000000"`, with Apple's deprecation of the IDFA as part of iOS 14, this field should be passed using all zeros, rather than removed or an empty string, as all zeros are the expected format.<br>**Note:** This fields can use UUIDv4 or UUIDv5 formats. If passing synthetic IFAs, you should use either the v4 or v5 implementation of RFC 4122[22] to generate them. There's a number of online tools for checking the version if you need to troubleshoot UUID issues, e.g. UUID / GUID Validator[23] |
| *make** | *string* | Device make, for example, `"Apple"` |
| *mccmnc** | *string* | Mobile carrier as the concatenated MCC-MNC code (e.g., `"310-005"` identifies Verizon Wireless CDMA in the USA). Refer to https://en.wikipedia.org/wiki/Mobile_country_code for further examples. **Note:** that the dash between the MCC and MNC parts is required to remove parsing ambiguity |
| *model** | *string* | Device mode, for example, `"iPhone"` |
| *os** | *string* | Device operating system, for example, `"iOS"` |
| *osv** | *string* | Device operating system version, for example, `"3.1.2"` |
| *h** | integer | Physical height of the screen in pixels, for example, `750` |
| *w** | *integer* | Physical width of the screen in pixels, for example, `1334` |
| *pxratio** | *float* | The ratio of physical pixels to device independent pixels, for example, `1.0` |
| *dnt** | *integer* | Do not track.<br>• `0`: do not track is set to false<br>• `1`: do not track is set to true in the browser, for example, `0` |
| *lmt** | *integer* | Limit Ad Tracking. Signal commercially endorsed (e.g., iOS, recommended Android):<br>• `0`: tracking is unrestricted,<br>• `1`: tracking must be limited per commercial guidelines, for example, `0` |
| *devicetype** | *integer* | Device type as defined by OpenRTB, for example, `4` |
| *js** | *integer* | 1 if the device supports JavaScript; otherwise 0. |
| *flashver** | *string* | Flash version detected, for example, `10.1` |

continues on next page

Table 40 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *ext** | *object* | Device extension object |

[22] https://tools.ietf.org/html/rfc4122
[23] https://www.freecodeformat.com/validate-uuid-guid.php

### Device Ext

Table 41: Device Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *atts** | *int* | (iOS Only) An integer passed to represent the app's app tracking authorization status, can contain the following values.<br>• `0` = not determined<br>• `1` = restricted<br>• `2` = denied<br>• `3` = authorized |
| *truncated_ip** | *integer* | Indicates whether the IP address in the `device.ip` field is truncated. `1` = truncated `0` = not truncated. It is assumed not truncated and required if the IP address is truncated. If not declared the bid request might be classified as invalid traffic, by Google. This is a field only for Google, see the Display & Video 360 OpenRTB Specification[24] |
| *ifa_type** | *string* | Indicates the origin of the `device.ifa` field, whether it was provided from the device itself or generated by a publisher or Supplier in the supply chain. Takes the following values from the Guidelines for Identifier for Advertising (IFA) on CTV/OTT platforms[25]<br>• `"aaid"` Android TV<br>• `"rida"` Roku<br>• `"afai"` Amazon Fire<br>• `"idfa"` Apple tvOS<br>• `"msai"` Xbox/Microsoft<br>• `"dpid"` Generic device platform ID<br>• `"ppid"` Publisher provided ID<br>• `"sspid"` SSP provided ID<br>• `"sessionid"` Short-lived session ID (frequency capping only) |
| *idv** | *str* | Passes the ID for Vendor (IDFV[26]). A persistent unique identifier for each app on a device that identifies the device to the app's vendor. The value of this property is the same for apps that come from the same vendor running on the same device. A different value is returned for apps on the same device that come from different vendors, and for apps on different devices regardless of vendor, e.g. `"1F277D46-12BF-482A-9085-B4F811DD6E4D"`<br>---<br>**Note:** Initially this field was passed as `idfv` but was officially changed after version 1 of the IAB specification. You should expect to see this is both forms: `idfv` or `idv`.<br>--- |

### 1.12.9 Geo Object

The information provided in the Geo Object is based on MaxMind database[27], except latitude and longitude values.

---

**Note:** Fields marked with an asterisk (*) are optional.

In the even that a user has opted-out under the GDPR or CCPA, the location data will be anonimised by reducing `geo.lat` and `geo.lon` 3 decimal places.

---

Table 42: Geo Object Properties

| Value | Type | Description |
|---|---|---|
| *accuracy** | int | Estimated location accuracy in meters. |
| *lat** | *float* | Latitude from -90 to 90. South is negative, for example, `52.35` |
| *lon** | *float* | Longitude from -180 to 180. West is negative, for example, `4.9167` |
| *type** | *integer* | Source of location data as defined by OpenRTB, for example, `1` |
| *country** | *string* | Country using ISO-3166-1[28] Alpha-2, for example `NL` |
| *region** | *string* | Region using ISO-3166-2 region codes, for example, `NY` |
| *city** | *string* | City name as provided by MaxMind, for example, `Alkmaar` |
| *metro** | string | Google metro code; similar to but not exactly Nielsen DMAs. |
| *zip** | *string* | Zip/postal code, for example, `"90210"` |
| *utcoffset** | *integer* | Local time as the number +/- of minutes from UTC, for example, `-240` |

#### Geo Object Example

```
{
"geo": {
      "country": "US",
      "region": "NY",
      "city": "White Plains",
      "zip": "10601",
      "utcoffset": -240
}}
```

[24] https://developers.google.com/display-video/ortb-spec#supported-extension-for-device-object
[25] https://iabtechlab.com/standards/guidelines-identifier-advertising-over-the-top-platforms/
[26] https://developer.apple.com/documentation/uikit/uidevice/1620059-identifierforvendor
[27] https://www.maxmind.com/en/geoip2-services-and-databases
[28] https://www.iso.org/iso-3166-country-codes.html

### 1.12.10 User Object

**Note:** Fields marked with asterisk (*) are optional.

Table 43: User Object Properties

| Value | Type | Description |
|---|---|---|
| *id** | *string* | Unique BidSwitch ID of this user, for example, `"252eb154-b3e5-473f-bad8-9b6d7f8646e5"`. For in-app traffic the lowercase IDFA, or Android ID is used. For example, `"38f72eaf-5d6f-4143-824f-deaf753d7239"`. The User ID can be a maximum of `50` characters. |
| *buyeruid** | *string* | The Buyer user ID as mapped by BidSwitch for the DSP. The User ID can be a maximum of `50` characters. |
| *keywords** | *string* | Comma separated list of keywords, interests, or intent, for example, `Cars, sports, vacation` |
| *yob** | *integer* | Year of birth as a 4-digit integer, for example, `1977` |
| *gender** | *string* | Specifies the user gender, for example, `"F"`<br>• `"M"` = Male<br>• `"F"` = Female<br>• `"O"` = Known to be other, or omitted if unknown |
| data* | *array of objects* | Additional data. Each data object represents a different data source, for more information, see the *Data Object* (page 106) section. |
| *ext** | *object* | For more information, see the *User Ext Object Properties* (page 77) section. |

#### User Ext Object Properties

Table 44: User Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *ug** | integer | User group set up for the target user during cookie syncing, for example `1`. This takes the following values:<br>• `0`: unmatched user<br>• `1`: matched user<br>• `2 - 99`: custom user group. |
| *cookie_age** | *integer* | Number of days since the user cookie has been dropped, for example, `10` |

continues on next page

Table 44 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *floc** | *object* | Cohort ID that is common to a large cohort of users with similar browsing habits. Currently the cohort ID is simulated by the exchange (as opposed to by the browser). When simulated cohort ID is provided, traditional pseudonymous cookie-based user identifiers or device advertising identifiers would not be populated. Experimental feature; may be subject to change. See https://github.com/WICG/floc for more background on FLoC.<br>**Note:** In the event FLOC is being used there should be no `user.id` or `user.buyerid` present. |
| *digitrust** | *object* | Deprecated since version 5.3: passing this value moved to the `eids` object<br>Container object for the DigiTrust ID and related attributes, see the *DigiTrust Object* (page 81) object for more details. |
| *eids** | *array of objects* | Contains the Extended identifiers object, see the *Extended Identifiers* (page 81) section for details |
| *google_consent** | *array of integers* | Set of IDs corresponding to providers for whom the publisher has provided user consent using Google vendor list. A mapping of provider ID to provider name is posted at https://storage.googleapis.com/adx-rtb-dictionaries/providers.csv<br>Deprecated since version 5.3.<br>Replaced by the `consented_providers_settings` object, with the IDs passed in its `consented_providers` array. |
| *impdepth** | *int* | The count of impressions for a specific placement type in a given app session. The impression depth is reset once the session ends, e.g `2` |
| *sessionduration** | *int* | The total duration of time a user has spent so far in a specific app session expressed in seconds. For example, a user has been playing Word Game for 45 seconds, e.g. `45` |
| *consented_providers_settings** | *object* | Passes a set of IDs corresponding to providers for whom the publisher has provided user consent using Google vendor list. See the *Consented Provider Settings* (page 80) |

continues on next page

Table 44 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *consent** | *string* | Passes the Transparency & Consent Framework string. The current valid version of this string is v2.0. You can read more about the the information encoded within the consent string here: Transparency and Consent String with Global Vendor & CMP List Formats[29]<br>The Buyer should use the information in the consent string to ascertain which vendors and for which purposes the user gave consent.<br>• If the user has not given consent, then do not respond with an ad which utilises user information and neither access nor store information on the user's device e.g. cookies, IDFA, fingerprints<br>• If the user has given consent, then identify all vendors to whom the user has given consent and for which purposes<br>– Buyers should also only use and store user data if the user has given consent to the buyer and only for the purposes for which the user has given consent<br>– Buyers should only allow third-party direct or redirect links to those who have received consent from the user<br>– https://vendor-list.consensu.org/v2/vendor-list.json<br>• If no consent is given, you cannot use personal data and may not have the right to use cookies. Each party is responsible for determining what that means for their business. If user consent explicitly states that it is not given, then do not respond with an ad which utilises user information and neither access nor store information on the user's device e.g. cookies, IDFA, fingerprints<br>For more information see the following links:<br>• https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework<br>• https://gdpr-info.eu/<br>• http://gdpr-demo.labs.quantcast.com/user-examples/cookie-workshop.html |

[29] https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework/blob/master/TCFv2/IAB%20Tech%20Lab%20-%20Consent%20string%20and%20vendor%20list%20formats%20v2.md

### Consented Provider Settings

Table 45: Consented Providers

| Value | Type | Description |
|---|---|---|
| *consented_providers* | *array of integers* | Set of IDs corresponding to providers for whom the publisher has provided user consent using Google vendor list. A mapping of provider ID to provider name is posted at https://storage.googleapis.com/adx-rtb-dictionaries/providers.csv |

### FloC Object

Table 46: FloC Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | The value of a cohort ID – a string identifier that is common to a large cohort of users with similar browsing habits. |
| *type* | *integer* | Identifies the method of generating the cohort ID, see this Google whitepaper for more details: Evaluation of Cohort Algorithms for the FloC API[30]<br>• FLOC_TYPE_UNKNOWN = 0; Default value that should not be used.<br>• SIMULATED_AFFINITY_CLUSTERING_CENTROID_VERTICAL = 2; FLoC simulated using affinity hierarchical clustering with centroids and feature extraction based on Topic categories as described in the whitepaper.<br>• SIMULATED_SIMHASH_SORTING_LSH_DOMAIN_ONE_HOT = 3; FLoC simulated using SortingLSH clustering algorithm and Domain One-hot encoding feature extraction as described in the whitepaper. FLoC simulated using a k Random Centers locality-sensitive hash function as described in github.com/google/ads-privacy/blob/master/proposals/FLoC/k-random-centers.md with Domain TF-IDF feature extraction as described in the whitepaper.<br>• KCENTER_DOM_FILTERED_TFDIF = 4; |

```
{
  "user":{
    "ext":{
      "floc":{
        "id":"str123",
```

(continues on next page)

[30] https://github.com/google/ads-privacy/blob/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf

(continued from previous page)

```
        "type":3
      }
    }
  }
}
```

### DigiTrust Object

Table 47: Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | User ID as provided by DigiTrust. |

### Extended Identifiers

Supports the official Open RTB community extension for passing multiple third party user identifiers, see the official Extended Identifiers[31] spec for more details.

This object passes any additional User IDs a Supplier, Consent Management Platform (CMP), Data Management Platform (DMP), or BidSwitch may have for a user. There is a number of identity solutions that provide IDs for users that are not based on 3rd party data, and many of these solution providers extend their first-party data as offerings that can be used to offset the decline of 3rd party cookies.

BidSwitch and many of our partners support these offerings and have build systems to collate, map, and make these IDs available to Buyers, who can use them to improve the robustness of targeted advertising. When the appropriate consent is in place (e.g CCPA, GDPR) BidSwitch **passes all and any** correctly set ID in this field. The following IDs are the most commonly sent through BidSwitch, but if you are looking for a particular ID in requests, you should check the `source` value to identify it. You can find a more extensive list of possible IDs on the Prebid User ID Module page[32]

- The TradeDesk ID known as the Unified ID 2.0 and passed as `"uidapi.com"`, is based on a Single-Sign On (SSO) email validation service for publisher site access.
- The TradeDesk ID known as the Unified ID 1.0 and passed as `"adserver.org"`.You can read more about that here https://www.thetradedesk.com/us/about-us/industry-initiatives/unified-id-solution-2-0/unified-id-solution-1-0
- The LiveIntent ID, passed as `"liveintent.com"`, is built and authenticated daily through consent-based email, you can read more about here https://www.liveintent.com/identity-solutions/
- The ID5 ID, passed as `"id5-sync.com"`, enables publishers to create and distribute a shared first-party identifier, you can read more about that here https://www.id5.io/

[31] https://github.com/InteractiveAdvertisingBureau/openrtb/tree/master/extensions/2.x_official_extensions
[32] https://docs.prebid.org/dev-docs/modules/userId.html

- The prebid managed SharedID, passed using `"sharedid.org"`, see https://prebid.org/product-suite/sharedid/ for more details
- Federated Learning of Cohorts FloC[33] IDs may be passed in this field when coming from an Supplier using prebid, with the source given as `"chrome.com"`.
- The Intimate Merger ID[34] is commonly used in the Japanese marketplace, with the source passed as `intimatemerger.com`.
- The LiveRamp ID, passed as `"liveramp.com"`, associates anonymous device IDs and other online customer IDs from publishers, platforms, or data providers with an *IdentityLink* (IDL) single person-based identifier. You can read more about it here https://liveramp.com/our-platform/identity-resolution/

---

**Note:**

- The LiveRamp ID is encrypted and only enabled for certain Buyers that have the business contracts in place to decrypt and use this ID, contact support@bidswitch.com if you are a LiveRamp partner.
- The LiveIntent ID is only available to certain Buyers to whom LiveIntent wish to grant usage rights.

---

Table 48: eids Object Properties

| Value | Type | Description |
|---|---|---|
| *source* | *string* | (**Required**) Source or technology provider responsible for the set of included IDs. Expressed as a top-level domain. BidSwitch includes the following IDs when available.<br>• The Trade Desk backed Unified ID passed using `"adserver.org"` for v1.0 and as `"uidapi.com"` for v2.0.<br>• id5 passed using `"id5-sync.com"`.<br>• The LiveIntent ID passed using `"liveintent.com"`.<br>• IDL passed using `"liveramp.com"`.<br>• SharedID passed using `"sharedid.org"`.<br>• FloC IDs may be passed using `"chrome.com"` if coming from prebid, but see the *FloC Object* (page 80) also. |
| *uids* | *array of objects* | (**Required**) Passes the User IDs matched from the given provider. |

[33] https://github.com/google/ads-privacy/tree/master/proposals/FLoC
[34] https://dmp.intimatemerger.com/service/materials/

### uids Object

Table 49: uids Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | (**Required**) The User ID with this provider. |
| *atype* | *int* | (**Optional**) The type of user agent the match is from.<br>• 1 An ID which is tied to a specific web browser or device (cookie-based, probabilistic, or other).<br>• 2 In-app impressions, which will typically contain a type of device ID (or rather, the privacy-compliant versions of device IDs).<br>• 3 A person-based ID, i.e., that is the same across devices.<br>• 500+ Vendor-specific codes. |

### User Object Example

```
{
  "user":{
    "id":"45asdf987656789adfad4678rew656789",
    "buyeruid":"1234567890",
    "keywords":"sports, entertainment",
    "yob":1976,
    "gender":"F",
    "ext":{
      "ug":1,
      "cookie_age":15,
      "consent":"Y29uc2VudCBkYXRh",
      "consented_providers_settings":{
        "consented_providers":[
          1791
        ]
      },
      "eids":[
        {
          "source":"adserver.org",
          "uids":[
            {
              "id":"zzz123",
              "atype":1
            },
            {
              "id":"DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
              "atype":2
```

(continues on next page)

(continued from previous page)

```
            }
          ]
        },
        {
          "source":"liveintent.com",
          "uids":[
            {
              "id":"IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
              "atype":3
            }
          ]
        },
        {
          "source":"liveramp.com",
          "uids":[
            {
              "id":"0db20294a3908612bc7e732c2022095391074cf3"
            }
          ]
        }
      ]
    }
  }
}
```

### 1.12.11 Site Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 50: Site Object Properties

| Value | Type | Description |
|---|---|---|
| *publisher* | *object* | Publisher object, for more information, see the *Publisher Object* (page 99) section. |
| *id* | *string* | An exchange specific identifier comprised of the Supplier's exchange name, and the target site ID appended following an underscore. For example, if a Supplier such as Rubicon sends a bid request for site ID `"123"`, the field becomes `"rubicon_123"` |
| *name** | *string* | Site name (may be masked by publisher request), for example, `"Test Site"` |
| *domain** | *string* | Domain of the site, used for advertiser side blocking. `"testsite.com"` |

continues on next page

Table 50 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *content** | *object* | Passes the content object, see the *Content Object* (page 88) section for details |
| *cat** | *array of strings* | Array of IAB content categories for the site. `["IAB1", "IAB2-3"]` |
| *page** | *string* | URL of the page where the impression will be shown. `"http://testsite.com/main.asp"` |
| *ref** | *string* | Referrer URL that caused navigation to the current page, for example, `"http://testsite.com/main.asp"` |
| *privacypolicy** | *integer* | Indicates if the site has a privacy policy.<br>• `0` = No<br>• `1` = Yes. |
| *mobile** | *integer* | Mobile-optimized signal.<br>• `0` = No<br>• `1` = Yes. |
| *ext** | *object* | Site extension object |

### Site Ext

Table 51: Site Extension Object Properties

| Value | Type | Description |
|---|---|---|
| *amp* | *bool* | Indicates if the site is AMP (Accelerated Mobile Pages) optimised, where `1` = Yes and `0` = No |
| *inventorypartnerdomain* | *string* | A pointer to the domain of the partner (of the site/app owner) with ownership of some portion of ad inventory on the site/app. The partner's `ads.txt` or `app-ads.txt` file will be hosted here. This directive was added in the (app-)ads.txt v1.0.3 specification[35] update. |

### Site Object Example

```
{
  "site":{
    "id":"abc35123",
    "name":"Site ABCD",
    "domain":"siteabcd.com",
    "cat":[
      "IAB2-1",
      "IAB2-2"
```

(continues on next page)

[35] https://iabtechlab.com/ads-txt/

(continued from previous page)

```
    ],
    "page":"http://siteabcd.com/page.htm",
    "ref":"http://referringsite.com/referringpage.htm",
    "privacypolicy":1,
    "ext":{
      "inventorypartnerdomain":"examplepartnerdomain.com"
    },
    "publisher":{
      "id":"abc2345",
      "name":"Publisher A"
    }
  }
}
```

### 1.12.12 App Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 52: App Object Properties

| Value | Type | Description |
|---|---|---|
| *publisher* | *object* | Publisher object, for more information, see the *Publisher Object* (page 99) section. |
| *id* | *string* | The application ID prefixed with the exchange name followed by an underscore. For example, if Rubicon sends a bid request for App ID "123", the field becomes `"rubicon_123"` |
| *name** | *string* | Application name, for example, `"Test App"` |
| *domain** | *string* | The domain of the app, for example, `"mygame.example.com"` |
| *cat** | *array of strings* | Array of IAB content categories for the publisher site, for example, `["IAB1", "IAB2- 3"]` |
| *content** | *object* | Passes the content object, see the *Content Object* (page 88) section for details |
| *bundle** | *string* | Application bundle or package name, for example, `"com.example.mygame"` |
| *paid** | *integer* | Specifies if the App is a free or paid version.<br>• `0` = The app is free,<br>• `1` = The app is a paid version. |
| *storeurl** | *string* | App store's URL for the mobile application, for example `"http://media-apps.cc/android"` |
| *storeid** | *string* | The ID of the app in an app store (e.g., Apple iTunes, Google Play). |
| *ver** | *string* | Application version, for example `"1.1"` |
| *privacypolicy** | *integer* | Indicates if the app has a privacy policy.<br>• `0` = No<br>• `1` = Yes. |

### Ext Object

Table 53: App Extension Object Properties

| Value | Type | Description |
|---|---|---|
| *inventorypartnerdomain* | *string* | A pointer to the domain of the partner (of the site/app owner) with ownership of some portion of ad inventory on the site/app. The partner's `ads.txt` or `app-ads.txt` file will be hosted here. This directive was added in the (app-)ads.txt v1.0.3 specification[36] update. |

[36] https://iabtechlab.com/ads-txt/

### 1.12.13 Content Object

**Note:** Due to how many Suppliers and Buyers adopted the content object after oRTB 3.0 was released, there can be some variations in how field names are really used as two options exist for some in the content object, e.g. `userrating` in 2.5 is `urating` in 3.0.

Here is the complete Adcom Content Object[37] Specification.

Table 54: Content Object Properties

| Value | Type | Description |
|---|---|---|
| *id** | *string* | ID uniquely identifying the content. |
| *episode** | *integer* | Episode number. |
| *title** | *string* | Content title e.g., `Lost Kingdom - SE02 EP03` |
| *season** | *string* | Content season e.g., `"Season 3"` |
| *artist** | *string* | Artist credited with the content |
| *genre** | *string* | Genre that best describes the content, e.g., rock, pop, etc. You may need to discuss with your trading partners about aligning around a particular taxonomy. There's a few taxonomies for defining the genre, the IAB Content taxonomy being the most standard one at the moment.<br>• Google have their own which you can download from their Display & Video 360 OpenRTB Specification[38]<br>• The IAB CONTENT TAXONOMY[39]<br>• An audio taxonomy is also being worked on by various entities, this page will be updated when a standard is finalised |
| *album** | *string* | Album to which the content belongs; typically for audio. |
| *isrc** | *string* | International Standard Recording Code conforming to ISO-3901. |
| *url** | *string* | A single URL of the content, for buy-side contextualization or review. |
| *cat** | *array of strings* | Array of content categories using IDs from the IAB taxonomy. |
| *prodq** | *integer* | Production quality: `1` Professionally Produced, `2` Prosumer `3` User Generated (UGC).<br>**Note:** This may also be passed as *videoquality* |

continues on next page

[37] https://github.com/InteractiveAdvertisingBureau/AdCOM/blob/master/AdCOM%20v1.0%20FINAL.md#object_content

Table 54 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *context** | *integer* | Type of content<br>• `1` Video (i.e., video file or stream such as Internet TV broadcasts)<br>• `2` Game (i.e., an interactive software game)<br>• `3` Music (i.e., audio file or stream such as Internet radio broadcasts)<br>• `4` Application (i.e., an interactive software application)<br>• `5` Text (i.e., primarily textual document such as a web page, eBook, or news article)<br>• `6` Other (i.e., none of the other categories applies)<br>• `7` Unknown |
| *rating** | *string* | Content rating (e.g., MPAA).<br>**Note:** this may also be passed as *contentrating* |
| *urating** | *string* | User rating of the content (e.g., number of stars, likes, etc.).<br>**Note:** this may also be passed as *urating* |
| *mrating** | *integer* | Media rating per IQG guidelines. Refers to the following list:<br>• `1` All Audiences<br>• `2` Everyone Over Age 12<br>• `3` Mature Audiences<br>• **Note:** this may also be passed as *qagmediarating* |
| *keywords** | *string* | Comma separated list of keywords describing the content. |
| *live** | *integer* | Indication of live content, where `0` = not live, `1` = live (e.g., stream, live blog).<br>**Note:** This may also be passed as *livestream* due to the difference between Open RTB 2.5 and 3.0. |
| *srcrel** | *integer* | Source relationship, where `0` = indirect, `1` = direct.<br>**Note:** This may also be passed as *sourcerelationship* due to the difference between Open RTB 2.5 and 3.0. |
| *len** | *integer* | Length of content in seconds; typically for video or audio. |
| *lang** | *string* | Content language using ISO-639-1-alpha-2.<br>**Note:** This may also be passed a *language* due to the difference between Open RTB 2.5 and 3.0. |
| *embed** | *integer* | Indicator of whether or not the content is embedded off-site from the the site or app described in those objects (e.g., an embedded video player), where 0 = no, 1 = yes.<br>**Note:** This may also be passed a *embedabble* due to the difference between Open RTB 2.5 and 3.0. |
| *producer** | *object* | Details about the content producer. Refer to Object: Producer. |
| *data** | *array of objects* | Additional user data. Each Data object represents a different data source, see the *Data Object* (page 106) |

[38] https://developers.google.com/display-video/ortb-spec#content-object
[39] https://iabtechlab.com/standards/content-taxonomy/

Listing 1: Content Object Example


```
{
  "content":{
    "cat":[
      "IAB-1",
      "IAB-2"
    ],
    "rating":"MPAA",
    "context":1,
    "episode":1,
    "genre":"comedy",
    "id":"472759950",
    "lang":"en",
    "len":15,
    "live":1,
    "prodq":1,
    "season":"Season 3",
    "title":"Law & Order Criminal Intent S03 - Ep18 Ill-Bred HD Watch",
    "url":"https://forums.watchuseek.com/showthread.php",
    "urating":"4.1"
  }
}
```


### 1.12.14 TV Object

Table 55: TV Object Properties

| Value | Type | Description |
|---|---|---|
| *publisher* | *object* | Publisher object, for more information, see the *Publisher Object* (page 99) section. |


```
{
  "ext":{
    "ssp":"abc123",
    "tv":{
      "publisher":{
        "id":"abc123_321"
      }
    }
  }
}
```

### 1.12.15 DOOH Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 56: DOOH Object Properties

| Value | Type | Description |
|---|---|---|
| *publisher* | *object* | Publisher object, for more information see the *Publisher Object* (page 99) section. |
| *audience** | *float* | Expected number of people reached by the ad opportunity, e.g. `10.5` |
| *impmultiply** | *float* | The *impmultiply* field is designed to be used when calculating the billable media cost by the Buyer and on the invoice. It should not be used to multiply the bid price in the bid response. For example, if the Buyer wins 3000 bids at a clearing price of $1.50 CPM and `impmultiply=4` each, then the invoiced amount is $18 (1.50 / 1000 * 4 * 3000).<br>The default value is `1` |

**Note:** Buyers should not use either the `audience` or `impmultiply` values when calculating their bid response. They should only return a bid based on CPM values.

Buyers should also ensure that their Suppliers are basing any billing calculation on the same multiple. Instead of the `impmultiply` value, some Suppliers base billing on the VAST *Total Impressions Macro* (page 38)

```
{
  "dooh":{
    "publisher":{
      "id":"rubicon_234563",
      "cat":[
        "IAB1"
      ]
    },
    "audience":1.56,
    "impmultiply":1.34
  }
}
```

### 1.12.16 Metric Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 57: Metric Object Properties

| Value | Type | Description |
|---|---|---|
| *type* | *string* | The type of metric being presented. Currently BidSwitch only supports using `viewability` as the metric type |
| *value* | float | A decimal number representing the value of the metric being supplied<br>• `viewability` probability is in the range *0.0 – 1.0*. |
| *vendor** | string | Source of the value declared by the Supplier |

#### Metric JSON Example

```
{
  "metric":[
    {
      "type":"viewability",
      "value":0.85
    }
  ]
}
```

### 1.12.17 Private Marketplace Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 58: Private Marketplace Object Properties

| Value | Type | Description |
|---|---|---|
| *private_auction** | *integer* | A value of 1 indicates that only bids submitted inside `pmp.deals` will take part in the auction. A value of 0 indicates that bids without deal information may also be considered for serving. |
| *deals* | *array of objects* | Array of Deal objects, for more information, see the *Deal Object* (page 97) section. |

#### Private Marketplace Object Example

```
{
   "pmp":{
      "private_auction":1,
      "deals":[
         {
            "id":"deal-1",
            "wseat":[
               "58"
            ],
            "bidfloor":2.5,
            "at":1
         },
         {
            "id":"deal-2",
            "bidfloor":2,
            "at":2
         }
      ]
   }
}
```

### 1.12.18 Source Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 59: Source Object Properties

| Value | Type | Description |
|---|---|---|
| *fd** | *integer* | Indicates the entity responsible for the final impression sale decision, using the following values:<br>• `0` = The exchange behind BidSwitch<br>• `1` = An upstream source (usually header bidder)<br>For example a bid request containing `ext.ssp='rubicon'` and `source.fd=0` implies that the auction is run at Rubicon SSP. If the same request has `source.fd=1` then the auction is run at a header bidder upstream from Rubicon SSP.<br>**Note**: The BidSwitch platform never acts as the decision maker. |
| *tid* | string | (**Recommended**) Transaction ID that must be common across all participants in this bid request (e.g., potentially multiple exchanges). |
| *pchain** | *string* | Payment ID chain string containing embedded syntax described in the TAG Payment ID Protocol v1.0. |
| *ext** | *object* | Contains additional fields, see *Source Extension* (page 94) |

### Source Extension

Table 60: Source Extension Object Properties

| Value | Type | Description |
|---|---|---|
| *omidpn* | *string* | Identifier of the OM SDK integration, the IAB Open Measurement specification on Github[40] |
| *omidpv* | *string* | Version of the OM SDK integration. |
| *schain* | *object* | Contains the supplychain object, as fully described here on the IAB Github Page[41]:<br>The SupplyChain object is composed primarily of a set of nodes where each node represents a specific entity that participates in the selling of a bid request. The entire chain of nodes from beginning to end would represent all sellers who were paid for an individual bid request. |

[40] https://github.com/InteractiveAdvertisingBureau/AdCOM/blob/master/OpenRTB%20support%20for%20OMSDK.md

[41] https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/supplychainobject.md

### SupplyChain Object

**Important:**

- This field is a recent addition to the OpenRTB Specification, and part of verifying the information passed in it requires Suppliers to provides a valid `sellers.json` file. The details of which can be found here: IAB sellers.json Spec[42]
- BidSwitch will only appear as a node when trading is conducted using Media Mode.

Table 61: schain

| Value | Type | Description |
|---|---|---|
| *complete* | *int* | (**Required**) Flag indicating whether the chain contains all nodes leading back to the source of the inventory, where `0` = no, `1` = yes. |
| *nodes* | *array of objects* | (**Required**) Array of objects in the order of placing in the chain. The original source of the request is first and the final seller of the request last, see *Supply Chain Nodes* (page 96) |
| *ver* | *str* | (**Required**) Version of the supply chain specification in use. Currently `"1.0"` is the only option. |

[42] https://iabtechlab.com/wp-content/uploads/2019/04/Sellers.json-Public-Comment-April-11-2019.pdf

### Supply Chain Nodes

Table 62: supply chain node

| Value | Type | Description |
|---|---|---|
| *asi* | *string* | (**Required**) The canonical domain name of the SSP, Exchange, Header Wrapper, etc system that bidders connect to. This may be the operational domain of the system, if that is different than the parent corporate domain, to facilitate WHOIS and reverse IP lookups to establish clear ownership of the delegate system. This should be the same value as used to identify sellers in an ads.txt file if one exists. |
| *sid* | *string* | (**Required**) The identifier associated with the seller or reseller account within the advertising system. This must contain the same value used in transactions (i.e. OpenRTB bid requests) in the field specified by the SSP/exchange. Typically, in OpenRTB, this is publisher.id. For OpenDirect it is typically the publisher's organization ID.Should be limited to 64 characters in length. |
| *hp* | *int* | (**Required**) Indicates whether this node will be involved in the flow of payment for the inventory.<br>When set to `1`, the advertising system in the `asi` field pays the seller in the `sid` field, who is responsible for paying the previous node in the chain.<br>When set to `0`, this node is not involved in the flow of payment for the inventory.<br>For version 1.0 of SupplyChain, this property should always be `1`. It is explicitly required to be included as it is expected that future versions of the specification will introduce non-payment handling nodes. Implementers should ensure that they support this field and propagate it onwards when constructing SupplyChain objects in bid requests sent to a downstream advertising system. |
| *rid** | *string* | The OpenRTB RequestId of the request as issued by this seller. |
| *name** | *string* | The business name of the entity represented by this node. This value is optional and should NOT be included if it exists in the advertising system's sellers.txt file. |
| *domain** | *string* | The business domain name of the entity represented by this node. This value is optional and should NOT be included if it exists in the advertising system's sellers.json file. |

#### Example Source JSON

```
{
    "source":{
        "fd":1,
        "tid":"FEDCBA9876543210",
        "ext":{
            "schain":{
                "complete":0,
                "ver":"1.0",
                "nodes":[
                    {
                        "asi":"reseller.com",
                        "sid":"aaaaa",
                        "rid":"BidRequest4",
                        "hp":1
                    }
                ]
            }
        }
    }
}
```

### 1.12.19 Deal Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 63: Deal Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | Deal ID, for example, `"AA-1234"` |
| *wseat** | *array of strings* | Array of Buyer seats allowed to bid on this Direct Deal, for example, `["58", "99"]`. If present, the allowed seat IDs may be supplied using the BidSwitch or Supplier taxonomy.<br>• The BidSwitch taxonomy uses the Buyer ID as the single seat ID value.<br>• The seat in the Supplier taxonomy may represent the whole Buyer or some entity on the Buyer side (e.g. agency)<br>• A bid request may contain multiple seat IDs in the Supplier taxonomy.<br>• The bid response should contain the appropriate seat value corresponding to one of values of the `wseat` field, see the *Seat Bid Object* (page 140) section. |
| *bidfloor** | *float* | Deal price in CPM. If it is a fixed price deal (`deal.at = 3`) then this is the exact price of the deal, otherwise this is the bid floor of the deal, for example, `1.3` |
| *bidfloorcur** | *string* | Bid floor currency specified using ISO-4217[43] alpha codes, for example, `"USD"` |
| *at** | *integer* | Auction type.<br>• `1` for first price auction.<br>• `2` for second price auction.<br>• `3` for fixed price deal. |
| *ext** | *object* | See the *Deal Extension Object* (page 98) section. |

### Deal Extension Object

Table 64: Deal Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *buyer_wseat** | *array of strings* | Specifies the Advertisers/Agencies that should have access to this deal in the Buyer's system. You should use their seat ID with the Buyer, e.g. `["agency-123", "advertiser-456"]`.<br>**Note**: To obtain the correct Seat ID for an agency at a particular Buyer you will need to contact the agency. You may also find more information about this in the `buyers` field description of the dpa section. |

[43] https://www.iso.org/iso-4217-currency-codes.html

#### Deal Object Example

```
{
  "pmp":{
    "private_auction":1,
    "deals":[
      {
        "id":"deal-1",
        "wseat":[
          "58"
        ],
        "bidfloor":2.5,
        "at":1,
        "ext":{
          "type":3,
          "buyer_wseat":[
            "abc-123"
          ]
        }
      },
      {
        "id":"deal-2",
        "bidfloor":2,
        "at":2
      }
    ]
  }
}
```

### 1.12.20 Publisher Object

Table 65: Publisher Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | An exchange specific identifier comprised of the the Supplier exchange name, and the publisher ID appended following an underscore. For example, if a Supplier such as Rubicon sends the publisher ID `"25"`, then the value of the field becomes `rubicon_25`.<br>For Google AdX bid requests, the Google Seller network ID is used. |
| *name** | string | Publisher name, for example `"AAP"` |
| *cat** | array of string | Array of IAB content categories for the publisher. `["IAB1", "IAB2-3"]` |
| *domain** | string | The highest level domain of the publisher, e.g. `example.com` |

**Note:** Fields marked with an asterisk (*) are optional and may not be sent in each request.

#### Publisher Object Example

```
{
   "publisher":{
      "id":"SSPid_12345",
      "name":"Publisher A"
   }
}
```

### 1.12.21 Ext Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 66: Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *ssp* | *string* | The Supplier identification string, for example `"rubicon"`. |
| *media_src* | *string* | The Supplier identification string, for example `"rubicon"`.<br>• For standard bid traffic, this will contain the same value as `ext.ssp` |
| *ads_txt** | *object* | Contains the ads.txt information about the Supplier, see the *ads.txt Object* (page 102) and ads-txt sections for more details. |
| *google** | *object* | *Google Object* (page 103). This field is optional and is used only in Google AdX bid requests. |
| *google_query_id** | *string* | This represents a unique ID for the overall query. In the event that there are multiple call-outs for a query, all call-out requests for that query will contain the same `google_query_id`, see more here: https://developers.google.com/authorized-buyers/rtb/openrtb-guide#bidrequestext<br>It is highly recommended to include this ID if you are selling Exchange Bidding Dynamic Allocation (EDBA) sourced inventory through BidSwitch to Google DV360. |
| *gumgum** | *object* | *Gumgum Object* (page 103). This field is optional and is used only in Gumgum bid requests. |
| *rubicon** | *object* | *Rubicon Object* (page 104). This field is optional and is used only in Rubicon bid requests. |
| *adtruth** | *object* | *Adtruth Object* (page 103). This field is optional and is present if the corresponding data is provided by the Supplier. |

continues on next page

Table 66 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *tv** | *object* | *TV Object* (page 90) describing the programmatic TV. One of these objects should be present in the request: `site`, `app`, `ext.tv`, or `ext.dooh`. |
| *dooh** | *object* | *DOOH Object* (page 91) describing the Digital Out Of Home inventory. One of these objects should be present in the request: `site`, `app`, `ext.tv`, or `ext.dooh`. |
| *clktrkrq** | *integer* | 1 for bid requests that require click tracking macro in the bid response, 0 otherwise. |
| *s2s_nurl** | integer | Deprecated since version 5.2: replaced by `imp.ext.s2s_nurl`, see *Impression Ext* (page 46)<br>`1` indicates a bid request that only uses a server-to-server win notification for won auctions. This has the following implications for the `${AUCTION_PRICE}` macro:<br>• The `adm` field cannot contain the `${AUCTION_PRICE}`. For v5.1 and v5.2 the `nurl` field should contain the macro.<br>• As of v5.2, the `burl` field should contain the `${AUCTION_PRICE}`<br>• BidSwitch expects that `nurl` calls for Bid Requests with `s2s_nurl` set to `1` should return a HTTP status `200`, `204`, or `30x`, see the *Server-to-Server (s2s) Calls* (page 40) section for more information.<br>See the *5.x Updating Overview* (page 24) and *burl Field Overview* (page 30) sections for more information. |
| *is_secure** | integer | 0 for non-secure pages; 1 for secure pages. Creatives for secure pages should be SSL-compliant.<br>Deprecated since version 2.6: Use `imp.secure` instead, see *Impression Object* (page 44). |
| *wt** | *float* | The SmartSwitch bid weight. This field shows the total number of bid requests with the same SmartSwitch Score that this Bid Request represents.<br>This number can be used to calculate the amount of traffic of this type that would be received if SmartSwitch were turned off. Any bid request that you receive with this weight (X) is highly likely to be sent X times without SmartSwitch filtering. See the ss-bid-weight section for more details.<br>When this field value equals 1, there are no unsent bid requests with a matching score. |
| *tgroup** | *array of integers* | An array of Targeting Group IDs which map to the IDs in the myBidSwitch UI. Each ID present indicates that its targeting settings matched this request. |

### ads.txt Object

Table 67: Ads.txt Object Properties

| Value | Type | Description |
|---|---|---|
| *status* | *int* | Indicates what information the ads.txt file contained regarding this Suppliers selling relationship with the publisher:<br>• 1 = direct<br>• 2 = reseller<br>• 3 = unauthorized |
| *pub_id* | *string* | Exchange-specific publisher ID, e.g "abc-123". Note: Do not use any of the following symbols in the publisher ID, as doing so may cause issues:<br>`, # % $ @ * & ? ! ` ~ " ' / \ \| ( ) { } [ ]+ = ^` |
| *auth_id** | *string* | Passes the TAG ID if present in the ads.txt file, e.g. `8765jfhfg09j` |
| *supplier_domain** | *string* | Passes the Supplier's domain listed in the publisher's ads.txt file, e.g. `"openx.com"` |

```
{
  "ext":{
    "ssp":"ssp-name",
    "media_src":"ssp-name",
    "tgroup":[
      123,
      456,
      321,
      765
    ],
    "ads_txt":{
      "status":2,
      "auth_id":"1kjgh7653",
      "pub_id":"537120563",
      "supplier_domain":"openx.com"
    }
  }
}
```

### Google Object

Table 68: Google Object Properties

| Value | Type | Description |
|---|---|---|
| *detected_vertical** | *array of objects* | List of detected vertical values as defined by Google, see *Detected_vertical Object* (page 103). |

### Detected_vertical Object

Table 69: Detected Vertical Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | integer | The vertical id as defined by AdX docs, for example, `1014` |
| *weight* | float | Weight for this vertical, in the (0.0, 1.0] range. More relevant verticals have higher weights, for example, `1.0` |

### Adtruth Object

Table 70: Adtruth Object Properties

| Value | Type | Description |
|---|---|---|
| *tdl_millis** | int64 | Time Difference Linking (TDL), which allows for differentiation between similarly configured devices that might share a DeviceInsight identifier, for example, `19534993` |
| RECIPE VERSION_x | string | The specific AdTruth device identification recipe version and the corresponding DeviceInsight identifier. There might be multiple AdTruth recipe versions, for example, `"WEB_APP_BRIDGE_4_0":"4FD87B97751E9C305FD2AF1AA2D3"` |

### Gumgum Object

Table 71: Gumgum Object Properties

| Value | Type | Description |
|---|---|---|
| *cat* | *array of string* | Site or application category in Gumgum taxonomy, for example, `["GGE23-1", "GGE22"]`. The full taxonomy can be found in this Google doc[44] |

[44] https://docs.google.com/spreadsheets/d/1qt3thvCHjpDzdpybpwNySkB8LNmEjvCyQrJKH6MYQio/export?format=csv&id=1qt3thvCHjpDzdpybpwNySkB8LNmEjvCyQrJKH6MYQio&gid=0

#### Rubicon Object

Table 72: Rubicon Object Properties

| Value | Type | Description |
|---|---|---|
| *ast* | *int* | Rubicon Project Auction SubType, e.g. `401`.<br>• `201` Modified First Price<br>• `401` Modified Second Price |

### 1.12.22 Regs Object

**Note:** Fields marked with asterisk (*) are optional.

Table 73: Regs Object Properties

| Value | Type | Description |
|---|---|---|
| *coppa** | *integer* | Flag indicating whether or not this request falls under the COPPA regulations established by the USA FTC.<br>• `0` = No.<br>• `1` = Yes. |
| *ext** | *object* | See the *Regs Ext Object* (page 105) |

### Regs Ext Object

Table 74: Regs Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *gdpr** | *integer* | Indicates whether the request falls under GDPR regulations:<br>• `0` = No<br>• `1` = Yes<br>• Under OpenRTB conventions for optional attributes, omission indicates **Unknown**<br>If consent is given, you should check if the `user.ext.consent` field is present to ascertain what form of consent was given, see the *User Ext Object Properties* (page 77) section |
| *us_privacy** | string | Passes the user privacy status for requests which fall under CCPA[45] regulations. The string uses 4 characters, e.g. `"1YN-"`, passed in the following order.<br>1. **Version Number** The IAB CCPA Specification version that applies to this string, passed as an integer. Currently only `1` is available.<br>2. **Explicit Notice** (`N` = No, `Y` = Yes, `-` = Not Applicable) Indicates whether explicit notice has been provided to the user as required by 1798.115 (d) of the CCPA and whether they have had the opportunity to opt-out of the sale of their data pursuant to 1798.120 and 1798.135 of the CCPA.<br>3. **Opted-Out** (`N` = No, `Y` = Yes, `-` = Not Applicable) Indicates whether the user has opted-out of the sale of their personal information pursuant to 1798.120 and 1798.135.<br>4. **LSPA** (`N` = No, `Y` = Yes, `-` = Not Applicable) Indicates whether the publisher is a signatory to the IAB Limited Service Provider Agreement (LSPA) and that the publisher declares the transaction should be treated as a "Covered Opt Out Transaction" or a "Non Opt Out Transaction" as defined in the agreement. |

```
{
    "regs":{
        "ext":{
            "gdpr":1,
            "us_privacy":"1YN-"
        }
    }
}
```

[45] https://iabtechlab.com/standards/ccpa/

### 1.12.23 Data Object

**Note:** Fields marked with asterisk (*) are optional.

Table 75: Data Object Properties

| Value | Type | Description |
|---|---|---|
| *id* * | *string* | Exchange-specific ID for the data provider, for example `"BSW001"` |
| *name* | *string* | Exchange-specific name for the data provider, for example `"domain-origin"` |
| *segment* | *array of objects* | Array of Segment objects that contain the actual data values, see *Segment Object* (page 106). |

#### Segment Object

Table 76: Segment Object Properties

| Value | Type | Description |
|---|---|---|
| *id** | string | ID of the data segment specific to the data provider, for example, `"Seg123"` |
| *name* | string | Name of the data segment specific to the data provider, for example, `"status"` |
| *value** | *string* | String representation of the data segment value, for example, `"verified"` |

```
{
   "data":[
      {
         "name":"domain-origin",
         "segment":[
            {
               "name":"status",
               "value":"verified"
            },
            {
               "name":"domain",
               "value":"abcd.com"
            }
         ]
      }
   ]
}
```

### 1.12.24 Bid Request JSON Examples

#### Banner Ad Example

```
{
    "site": {
        "id": "gumgum_www.answers.com_ed2265d8",
        "ref": "http://ad32.answers.com/click.php?source=fb&param4=fb-us-de-red&
↪param3=www.answers.com%2Farticle%2F31029589%2Finsanely-useful-life-hacks-to-make-
↪everything-easier&param1=tattoo&param2=67660042&param5=10153631993521186&
↪param6=6049542139960&adt=4342",
        "publisher": {
            "name": "www.answers.com",
            "id": "gumgum_946353442_12535"
        },
        "name": "www.answers.com",
        "cat": [
            "IAB24"
        ],
        "domain": "answers.com",
        "page": "http://www.answers.com/article/31029589/insanely-useful-life-
↪hacks-to-make-everything-easier?paramt=null&param4=fb-us-de-red&param1=tattoo&
↪param2=67660042&s=8"
    },
    "wseat": [
        "165",
        "16"
    ],
    "source": {
        "fd": 0
    },
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890",
        "keywords": "sports, entertainment",
        "yob": 1976,
        "gender": "F",
        "ext": {
            "ug": 1,
            "cookie_age": 15,
            "consent": "Y29uc2VudCBkYXRh",
            "consented_providers_settings": {
                "consented_providers": [
                    1791
                ]
            },
```

(continues on next page)

(continued from previous page)


```
            "eids": [
                {
                    "source": "adserver.org",
                    "uids": [
                        {
                            "id": "zzz",
                            "atype": 1
                        },
                        {
                            "id": "DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
                            "atype": 2
                        }
                    ]
                },
                {
                    "source": "liveintent.com",
                    "uids": [
                        {
                            "id": "IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
                            "atype": 3
                        }
                    ]
                },
                {
                    "source": "liveramp.com",
                    "uids": [
                        {
                            "id": "0db20294a3908612bc7e732c2022095391074cf3"
                        }
                    ]
                }
            ]
        }
    },
    "device": {
        "pxratio": 0,
        "language": "en",
        "mccmnc": "310-005",
        "w": 1920,
        "geo": {
            "country": "US",
            "lon": -80.237,
            "city": "West Palm Beach",
            "lat": 26.638,
            "zip": "33414",
```


(continues on next page)

(continued from previous page)


```
            "region": "FL",
            "type": 2
        },
        "os": "Windows",
        "devicetype": 2,
        "h": 1080,
        "ip": "73.139.39.18",
        "js": 1,
        "ua": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/
↪47.0",
        "dnt": 0
    },
    "tmax": 75,
    "cur": [
        "USD"
    ],
    "imp": [
        {
            "bidfloor": 3.213,
            "metric": [
                {
                    "type": "viewability",
                    "value": 0.85
                }
            ],
            "id": "1",
            "banner": {
                "pos": 1,
                "h": 600,
                "battr": [
                    1,
                    3,
                    5,
                    6,
                    8,
                    9,
                    10,
                    14,
                    15,
                    16
                ],
                "w": 160,
                "format": [
                    {
                        "h": 300,
```


(continues on next page)

(continued from previous page)

```
                        "w": 300
                    },
                    {
                        "h": 350,
                        "w": 300
                    }
                ],
                "btype": [
                    1
                ]
            },
            "exp": 300,
            "tagid": "gumgum_25108",
            "bidfloorcur": "USD",
            "ext": {},
            "secure": 0,
            "instl": 0
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "regs": {
        "ext": {
            "gdpr": 1
        }
    },
    "ext": {
        "wt": 1,
        "clktrkrq": 0,
        "is_secure": 0,
        "ssp": "gumgum",
        "media_src": "gumgum",
        "tgroup": [
            123,
            456,
            321,
            765
        ],
        "ads_txt": {
```

(continues on next page)

(continued from previous page)

```
            "status": 2,
            "auth_id": "1kjgh7653",
            "pub_id": "537120563",
            "supplier_domain": "example.com"
        }
    },
    "at": 2
}
```

### In App Example

```
{
    "regs": {
        "coppa": 0
    },
    "id": "4ecfe2ab-c275-48fb-8c0b-c7103579eaa0",
    "source": {
        "fd": 1,
        "ext": {
            "schain": {
                "complete": 0,
                "ver": "1.0",
                "nodes": [
                    {
                        "asi": "reseller.com",
                        "sid": "aaaaa",
                        "rid": "BidRequest4",
                        "hp": 1
                    }
                ]
            }
        }
    },
    "app": {
        "id": "adaptv_",
        "publisher": {
            "name": "",
            "id": "adaptv_11690"
        },
        "storeurl": "https://play.google.com/store/apps/details?id=com.zynga.looney
↪",
        "bundle": "com.zynga.looney",
        "cat": [
            "IAB1"
```

(continues on next page)

(continued from previous page)


```
        ],
        "name": "looney tunes dash!"
    },
    "wseat": [
        "68"
    ],
    "user": {
        "id": "dfc68ac9-9530-44b0-bdea-44cf153d7cea",
        "ext": {
            "ug": 1
        }
    },
    "device": {
        "dpidsha1": "0d7e6f65e1db717f0ed298bd268cc6415fa72124",
        "language": "en",
        "mccmnc": "310-005",
        "geo": {
            "country": "US",
            "lon": -78.83,
            "city": "Hamburg",
            "lat": 42.71,
            "zip": "14075",
            "region": "NY",
            "type": 2
        },
        "ifa": "dfc68ac9-9530-44b0-bdea-44cf153d7cea",
        "lmt": 0,
        "os": "android",
        "devicetype": 4,
        "ip": "72.88.84.159",
        "ua": "Mozilla/5.0 (Linux; Android 6.0.1; SM-G920P Build/MMB29K; wv)␣
↪AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/51.0.2704.81 Mobile␣
↪Safari/537.36",
        "dpidmd5": "c65cf7c5bffe94c62cf20c5e465d92f7"
    },
    "tmax": 120,
    "cur": [
        "USD"
    ],
    "imp": [
        {
            "bidfloor": 0.02268,
            "metric": [
                {
                    "type": "viewability",
```


(continues on next page)

(continued from previous page)


```
                    "value": 0.85
                }
            ],
            "id": "1",
            "instl": 0,
            "exp": 300,
            "bidfloorcur": "USD",
            "ext": {},
            "secure": 0,
            "video": {
                "protocols": [
                    2,
                    5
                ],
                "placement": 2,
                "playbackend": 1,
                "minduration": 5,
                "playbackmethod": [
                    1
                ],
                "maxduration": 30,
                "startdelay": 0,
                "linearity": 1,
                "mimes": [
                    "video/mp4"
                ]
            }
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "ext": {
        "is_secure": 0,
        "wt": 1,
        "clktrkrq": 0,
        "ssp": "adaptv",
        "media_src": "adaptv"
    },
    "at": 2
}
```

### In App iOS 14

**Note:** See the highlighted lines that denote the iOS 14 fields, see the *Protocol Release Notes* (page 2) for more details.

```
{
    "regs": {
        "ext": {
            "gdpr": 0,
            "us_privacy": "1YN-"
        }
    },
    "id": "4ecfe2ab-c275-48fb-8c0b-c7103579eaa0",
    "source": {
        "fd": 1,
        "ext": {
            "schain": {
                "complete": 0,
                "ver": "1.0",
                "nodes": [
                    {
                        "asi": "reseller.com",
                        "sid": "aaaaa",
                        "rid": "BidRequest4",
                        "hp": 1
                    }
                ]
            }
        }
    },
    "app": {
        "id": "adaptv_",
        "publisher": {
            "name": "Fyber",
            "id": "pubnative_1007055"
        },
        "storeurl": "https://apps.apple.com/us/app/wordscapes/id1207472156",
        "bundle": "1207472156",
        "cat": [
            "IAB1"
        ],
        "name": "Wordscapes"
    },
    "wseat": [
        "68"
```

(continues on next page)

(continued from previous page)


```
    ],
    "user": {
        "ext": {
            "impdepth": 2,
            "sessionduration": 45
        }
    },
    "device": {
        "dpidsha1": "0d7e6f65e1db717f0ed298bd268cc6415fa72124",
        "language": "en",
        "mccmnc": "310-005",
        "geo": {
            "country": "US",
            "lon": -78.83,
            "city": "Hamburg",
            "lat": 42.71,
            "zip": "14075",
            "region": "NY",
            "type": 2
        },
        "ifa": "00000000-0000-0000-0000-000000000000",
        "ext": {
            "idfv": "1F277D46-12BF-482A-9085-B4F811DD6E4D"
        },
        "lmt": 1,
        "os": "ios",
        "devicetype": 4,
        "ip": "72.88.84.159",
        "ua": "Mozilla/5.0 (Linux; Android 6.0.1; SM-G920P Build/MMB29K; wv)␣
↪AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/51.0.2704.81 Mobile␣
↪Safari/537.36"
    },
    "tmax": 120,
    "cur": [
        "USD"
    ],
    "imp": [
        {
            "bidfloor": 0.02268,
            "metric": [
                {
                    "type": "viewability",
                    "value": 0.85
                }
            ],
```


(continues on next page)

(continued from previous page)

```
            "id": "1",
            "secure": 0,
            "video": {
                "protocols": [
                    2,
                    5
                ],
                "placement": 2,
                "playbackend": 1,
                "minduration": 5,
                "playbackmethod": [
                    1
                ],
                "maxduration": 30,
                "startdelay": 0,
                "linearity": 1,
                "mimes": [
                    "video/mp4"
                ]
    },
            "instl": 0,
            "exp": 300,
            "bidfloorcur": "USD",
            "ext": {
                "skadn": {
                    "version": "2.0",
                    "sourceapp": "90987654321",
                    "skadnetids": [
                        "abc123.skadnetwork",
                        "def234.skadnetwork"
                    ]
                }
            }
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "ext": {
        "wt": 1,
```

(continues on next page)

(continued from previous page)


```
        "clktrkrq": 0,
        "is_secure": 0,
        "ssp": "gumgum",
        "media_src": "gumgum",
        "tgroup": [
            123,
            456,
            321,
            765
        ],
        "ads_txt": {
            "status": 2,
            "auth_id": "1kjgh7653",
            "pub_id": "537120563",
            "supplier_domain": "example.com"
        }
    },
    "at": 2
}
```


### Native Ad


```
{
    "id": "129ca6dd-5403-4476-a4a6-555d6a538bc4",
    "app": {
        "id": "pubnative_1009429",
        "publisher": {
            "name": "",
            "id": "pubnative_1005292"
        },
        "storeurl": "https://play.google.com/store/apps/details?id=com.leo.
↪appmaster",
        "bundle": "com.leo.appmaster",
        "cat": [
            "IAB3"
        ],
        "name": "PG_lock_pic"
    },
    "wseat": [
        "167"
    ],
    "source": {
        "fd": 0
    },
```


(continues on next page)

(continued from previous page)

```
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890",
        "keywords": "sports, entertainment",
        "yob": 1976,
        "gender": "F",
        "ext": {
            "ug": 1,
            "cookie_age": 15,
            "consent": "Y29uc2VudCBkYXRh",
            "consented_providers_settings": {
                "consented_providers": [
                    1791
                ]
            },
            "eids": [
                {
                    "source": "adserver.org",
                    "uids": [
                        {
                            "id": "zzz",
                            "atype": 1
                        },
                        {
                            "id": "DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
                            "atype": 2
                        }
                    ]
                },
                {
                    "source": "liveintent.com",
                    "uids": [
                        {
                            "id": "IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
                            "atype": 3
                        }
                    ]
                },
                {
                    "source": "liveramp.com",
                    "uids": [
                        {
                            "id": "0db20294a3908612bc7e732c2022095391074cf3"
                        }
                    ]
```

(continues on next page)

(continued from previous page)


```
                }
            ]
        }
    },
    "device": {
        "connectiontype": 3,
        "model": "Micromax A096",
        "mccmnc": "310-005",
        "language": "en",
        "geo": {
            "country": "IN",
            "lon": 85.1167,
            "city": "Patna",
            "lat": 25.6,
            "zip": "800002",
            "region": "34",
            "type": 2
        },
        "ifa": "793ff4b0-d077-4002-aeb6-b8ea64dd4b2b",
        "osv": "5.0.2",
        "os": "Android",
        "carrier": "Airtel",
        "devicetype": 1,
        "ip": "223.176.12.242",
        "ua": "Dalvik/2.1.0 (Linux; U; Android 5.0.2; Micromax A096 Build/LRX21M)",
        "dnt": 1
    },
    "tmax": 80,
    "cur": [
        "USD"
    ],
    "imp": [
        {
            "bidfloor": 0.324,
            "id": "1",
            "native": {
                "request": {
                    "plcmtcnt": 1,
                    "plcmttype": 2,
                    "privacy": 1,
                    "context": 1,
                    "contextsubtype": 12,
                    "assets": [
                        {
                            "id": 1,
```


(continues on next page)

(continued from previous page)

```
                "data": {
                    "type": 12
                },
                "required": 1
            },
            {
                "title": {
                    "len": 50
                },
                "id": 2,
                "required": 1
            },
            {
                "id": 3,
                "img": {
                    "w": 80,
                    "h": 80,
                    "type": 1
                },
                "required": 1
            },
            {
                "id": 4,
                "img": {
                    "w": 1200,
                    "h": 627,
                    "type": 3
                },
                "required": 1
            },
            {
                "data": {
                    "type": 3
                },
                "id": 5,
                "required": 0
            },
            {
                "id": 6,
                "data": {
                    "len": 100,
                    "type": 2
                },
                "required": 1
            }
```

(continues on next page)

(continued from previous page)


```
                    ],
                    "ver": "1.2"
                }
            },
            "exp": 1800,
            "bidfloorcur": "USD",
            "ext": {},
            "instl": 0
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "ext": {
        "wt": 112,
        "clktrkrq": 0,
        "is_secure": 0,
        "ssp": "pubnative",
        "tgroup": [
            123,
            456,
            321,
            765
        ],
        "ads_txt": {
            "status": 2,
            "auth_id": "1kjgh7653",
            "pub_id": "537120563",
            "supplier_domain": "example.com"
        }
    },
    "at": 2
}
```

### Private Deal Example


```
{
    "id": "500da108-85f8-44af-ac98-d7adcf9e0daf",
    "site": {
        "id": "spotx_SpotX_www.businessinsider.com",
        "publisher": {
            "name": "",
            "id": "spotx_84548"
        },
        "domain": "businessinsider.com",
        "page": "https://www.businessinsider.com/japanese-potato-changed-olivia-
↪munns-appearance-2016-2"
    },
    "wseat": [
        "145"
    ],
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890",
        "keywords": "sports, entertainment",
        "yob": 1976,
        "gender": "F",
        "ext": {
            "ug": 1,
            "cookie_age": 15,
            "consent": "Y29uc2VudCBkYXRh",
            "consented_providers_settings": {
                "consented_providers": [
                    1791
                ]
            },
            "eids": [
                {
                    "source": "adserver.org",
                    "uids": [
                        {
                            "id": "zzz",
                            "atype": 1
                        },
                        {
                            "id": "DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
                            "atype": 2
                        }
                    ]
                },
```


(continues on next page)

(continued from previous page)

```
                {
                    "source": "liveintent.com",
                    "uids": [
                        {
                            "id": "IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
                            "atype": 3
                        }
                    ]
                },
                {
                    "source": "liveramp.com",
                    "uids": [
                        {
                            "id": "0db20294a3908612bc7e732c2022095391074cf3"
                        }
                    ]
                }
            ]
        }
    },
    "device": {
        "model": "Chrome - Windows",
        "mccmnc": "310-005",
        "language": "en",
        "geo": {
            "country": "US",
            "city": "Jackson Heights",
            "region": "NY",
            "zip": "11372"
        },
        "make": "Google",
        "osv": "NT 6.1",
        "os": "Windows 7",
        "devicetype": 2,
        "ip": "98.14.26.33",
        "ua": "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)␣
↪Chrome/43.0.2342.0 Safari/537.36",
        "dnt": 0
    },
    "tmax": 415,
    "cur": [
        "USD"
    ],
    "imp": [
        {
```

(continues on next page)

(continued from previous page)


```
            "bidfloor": 0.011232,
            "metric": [
                {
                    "type": "viewability",
                    "value": 0.85
                }
            ],
            "id": "1",
            "pmp": {
                "private_auction": 1,
                "deals": [
                    {
                        "bidfloor": 10.8,
                        "id": "14a44.a7993.d401",
                        "bidfloorcur": "USD",
                        "at": 2
                    }
                ]
            },
            "instl": 0,
            "exp": 300,
            "tagid": "spotx_http://search.spotxchange.com/vast/2.00/149110?VPAID=1&
↪content_type=game&vmaxd=30&player_width=640&player_height=480&content_page_
↪url=http%3A%2F%2Fwww.businessinsider.com%2Fjapanese-potato-changed-olivia-munns-
↪appearance-2016-2&cb=1468903661330&adtid=T6GDD32DDC8UD5OOS5I6QROV4O&viewability=0
↪",
            "bidfloorcur": "USD",
            "secure": 0,
            "video": {
                "protocols": [
                    2,
                    5
                ],
                "companiontype": [
                    1,
                    2,
                    3
                ],
                "placement": 2,
                "playbackend": 1,
                "minduration": 1,
                "maxduration": 60,
                "startdelay": 0,
                "api": [
                    1
```


(continues on next page)

(continued from previous page)


```
                ],
                "linearity": 1,
                "h": 480,
                "delivery": [
                    2
                ],
                "ext": {
                    "player_type": 3
                },
                "mimes": [
                    "video/x-flv",
                    "video/mp4",
                    "application/x-shockwave-flash"
                ],
                "w": 640,
                "companionad": [
                    {
                        "h": 250,
                        "id": "1",
                        "w": 300,
                        "btype": [
                            4,
                            21
                        ]
                    }
                ]
            }
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "ext": {
        "is_secure": 0,
        "wt": 1,
        "clktrkrq": 0,
        "ssp": "spotx",
        "media_src": "spotx"
    },
    "at": 2
}
```

### Video Example


```
{
    "regs": {
        "coppa": 0,
        "ext": {
            "gdpr": 1
        }
    },
    "id": "75c0238c-3b52-4b87-957a-817f83e853f1",
    "site": {
        "id": "adaptv_",
        "publisher": {
            "name": "",
            "id": "adaptv_4182"
        },
        "cat": [
            "IAB1"
        ],
        "page": "http://kissasian.com"
    },
    "wseat": [
        "126"
    ],
    "source": {
        "fd": 1,
        "ext": {
            "schain": {
                "complete": 0,
                "ver": "1.0",
                "nodes": [
                    {
                        "asi": "reseller.com",
                        "sid": "aaaaa",
                        "rid": "BidRequest4",
                        "hp": 1
                    }
                ]
            }
        }
    },
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890",
        "keywords": "sports, entertainment",
        "yob": 1976,
```


(continues on next page)

(continued from previous page)


```
        "gender": "F",
        "ext": {
            "ug": 1,
            "cookie_age": 15,
            "consent": "Y29uc2VudCBkYXRh",
            "consented_providers_settings": {
                "consented_providers": [
                    1791
                ]
            },
            "eids": [
                {
                    "source": "adserver.org",
                    "uids": [
                        {
                            "id": "zzz",
                            "atype": 1
                        },
                        {
                            "id": "DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
                            "atype": 2
                        }
                    ]
                },
                {
                    "source": "liveintent.com",
                    "uids": [
                        {
                            "id": "IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
                            "atype": 3
                        }
                    ]
                },
                {
                    "source": "liveramp.com",
                    "uids": [
                        {
                            "id": "0db20294a3908612bc7e732c2022095391074cf3"
                        }
                    ]
                }
            ]
        }
    },
    "device": {
```


(continues on next page)

(continued from previous page)


```
        "language": "en",
        "mccmnc": "310-005",
        "geo": {
            "country": "US",
            "lon": -75.15,
            "city": "Philadelphia",
            "lat": 39.94,
            "zip": "19147",
            "region": "PA",
            "type": 2
        },
        "lmt": 0,
        "os": "Other",
        "devicetype": 6,
        "ip": "73.141.79.240",
        "ua": "Mozilla/5.0 (PlayStation 4 3.55) AppleWebKit/537.78 (KHTML, like␣
↪Gecko)"
    },
    "tmax": 120,
    "cur": [
        "USD"
    ],
    "imp": [
        {
            "bidfloor": 0.02268,
            "id": "1",
            "instl": 0,
            "exp": 300,
            "bidfloorcur": "USD",
            "ext": {},
            "secure": 0,
            "video": {
                "protocols": [
                    2,
                    5
                ],
                "placement": 2,
                "skip": 1,
                "playbackend": 1,
                "minduration": 5,
                "playbackmethod": [
                    3
                ],
                "maxduration": 60,
                "startdelay": 0,
```


(continues on next page)

(continued from previous page)

```
                "linearity": 1,
                "mimes": [
                    "video/mp4"
                ]
            }
        }
    ],
    "bcat": [
        "IAB25-3",
        "BSW1",
        "BSW2",
        "BSW10",
        "BSW4",
        "IAB26"
    ],
    "ext": {
        "wt": 1,
        "clktrkrq": 0,
        "is_secure": 0,
        "ssp": "adaptv",
        "media_src": "adaptv",
        "s2s_nurl": 1
    },
    "at": 2
}
```

### Audio Ad Example

```
{
    "id": "1234534625253",
    "wseat": [
        "58"
    ],
    "source": {
        "fd": 1,
        "ext": {
            "schain": {
                "complete": 0,
                "ver": "1.0",
                "nodes": [
                    {
                        "asi": "reseller.com",
                        "sid": "aaaaa",
                        "rid": "BidRequest4",
```

(continues on next page)

(continued from previous page)

```
                    "hp": 1
                }
            ]
        }
    }
},
"imp": [
    {
        "id": "1",
        "secure": 1,
        "audio": {
            "startdelay": 0,
            "minduration": 5,
            "maxduration": 30,
            "maxextended": 30,
            "minbitrate": 300,
            "maxbitrate": 1500,
            "api": [
                1,
                2
            ],
            "protocols": [
                9,
                10
            ],
            "mimes": [
                "audio/aac",
                "audio/mp4",
                "audio/mpeg"
            ],
            "delivery": [
                2
            ],
            "battr": [
                13,
                14
            ],
            "companionad": [
                {
                    "id": "1234567893-1",
                    "w": 300,
                    "h": 250,
                    "pos": 1,
                    "btype": [
                        4,
```

(continues on next page)

(continued from previous page)

```
                        21
                    ],
                    "battr": [
                        13,
                        14
                    ],
                    "expdir": [
                        2,
                        4
                    ]
                },
                {
                    "id": "1234567893-2",
                    "w": 728,
                    "h": 90,
                    "pos": 1,
                    "battr": [
                        13,
                        14
                    ]
                }
            ],
            "companiontype": [
                1,
                2
            ]
        }
    }
],
"site": {
    "id": "google_234563",
    "domain": "siteabcd.com",
    "page": "https://siteabcd.com/page.htm",
    "ref": "http://google.com/?q=siteabcd",
    "publisher": {
        "id": "google_25"
    }
},
"device": {
    "ip": "64.124.253.1",
    "mccmnc": "310-005",
    "geo": {
        "country": "US",
        "region": "NY",
        "city": "White Plains",
```

(continues on next page)

(continued from previous page)

```
            "zip": "10601"
        },
        "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",
        "language": "en"
    },
    "ext": {
        "wt": 1,
        "clktrkrq": 0,
        "is_secure": 0,
        "ssp": "gumgum",
        "media_src": "gumgum",
        "tgroup": [
            123,
            456,
            321,
            765
        ],
        "ads_txt": {
            "status": 2,
            "auth_id": "1kjgh7653",
            "pub_id": "537120563",
            "supplier_domain": "example.com"
        }
    },
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "buyeruid": "1234567890",
        "keywords": "sports, entertainment",
        "yob": 1976,
        "gender": "F",
        "tmax": 120,
        "ext": {
            "ug": 1,
            "cookie_age": 15,
            "consent": "Y29uc2VudCBkYXRh",
            "consented_providers_settings": {
                "consented_providers": [
                    1791
                ]
            },
            "eids": [
                {
                    "source": "adserver.org",
                    "uids": [
```

(continues on next page)

(continued from previous page)


```
                        {
                            "id": "zzz",
                            "atype": 1
                        },
                        {
                            "id": "DB700403-9A24-4A4B-A8D5-8A0B4BE777D2",
                            "atype": 2
                        }
                    ]
                },
                {
                    "source": "liveintent.com",
                    "uids": [
                        {
                            "id": "IPl4zj44RhezVyNE83bYfogYRN6W8LaCy3USy8dPQ==",
                            "atype": 3
                        }
                    ]
                },
                {
                    "source": "liveramp.com",
                    "uids": [
                        {
                            "id": "0db20294a3908612bc7e732c2022095391074cf3"
                        }
                    ]
                }
            ]
        }
    },
    "regs": {
        "ext": {
            "gdpr": 0,
            "us_privacy": "1YN-"
        }
    },
    "cur": [
        "USD"
    ]
}
```

### TV Ad Example


```
{
    "id": "1234534625253",
    "wseat": [
        "58"
    ],
    "imp": [
        {
            "id": "1",
            "bidfloor": 15,
            "exp": 360,
            "video": {
                "mimes": [
                    "video/x-flv",
                    "video/mp4",
                    "application/x-shockwave-flash",
                    "application/javascript"
                ],
                "placement": 2,
                "skip": 0,
                "playbackend": 1,
                "minduration": 30,
                "maxduration": 30,
                "protocols": [
                    3,
                    6
                ]
            }
        }
    ],
    "source": {
        "fd": 0
    },
    "device": {
        "geo": {
            "country": "US",
            "region": "TX"
        },
        "devicetype": 6
    },
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "keywords": "A_18_24,COLLEGE,HHI_75_100,M_18_24,HH_SINGLE",
        "data": [
            {
```


(continues on next page)

(continued from previous page)

```
                "name": "program_types",
                "segment": [
                    {
                        "name": "HOCKEY_DEC",
                        "value": "8"
                    },
                    {
                        "name": "BASEBALL_DEC",
                        "value": "9"
                    },
                    {
                        "name": "FOOTBALL_DEC",
                        "value": "8.1"
                    }
                ]
            }
        ]
    },
    "tmax": 150,
    "cur": [
        "USD",
        "EUR"
    ],
    "ext": {
        "ssp": "dish",
        "media_src": "dish",
        "tv": {
            "publisher": {
                "id": "dish_"
            }
        }
    }
}
```

### DOOH Ad Example

```
{
    "id": "1234534625253",
    "wseat": [
        "58"
    ],
    "imp": [
        {
            "id": "06d690d1-bac0-43fa-9257-f6d62d984231",
```

(continues on next page)

(continued from previous page)

```
            "bidfloor": 15,
            "exp": 360,
            "video": {
                "mimes": [
                    "video/x-flv",
                    "video/mp4"
                ],
                "placement": 2,
                "playbackend": 1,
                "skip": 0,
                "exp": 300,
                "minduration": 30,
                "maxduration": 30,
                "protocols": [
                    3,
                    6,
                    7
                ],
                "w": 640,
                "h": 480
            },
            "pmp": {
                "private_auction": 1,
                "deals": [
                    {
                        "id": "deal-1",
                        "wseat": [
                            "58"
                        ],
                        "bidfloor": 2.5,
                        "bidfloorcur": "USD",
                        "at": 3
                    }
                ]
            }
        }
    ],
    "device": {
        "ifa": "035911ea467d03b65cf44f5633b",
        "mccmnc": "310-005",
        "ip": "64.124.253.1",
        "geo": {
            "lat": 35.012344,
            "lon": -115.12345,
            "country": "US",
```

(continues on next page)

(continued from previous page)

```
            "region": "NY",
            "city": "White Plains",
            "zip": "10601"
        },
        "devicetype": 6
    },
    "user": {
        "id": "45asdf987656789adfad4678rew656789",
        "data": [
            {
                "id": "6",
                "name": "Data Provider 1",
                "segment": [
                    {
                        "name": "M",
                        "value": "15"
                    },
                    {
                        "name": "F",
                        "value": "9"
                    },
                    {
                        "name": "M65+",
                        "value": "8.1"
                    },
                    {
                        "name": "TOTAL",
                        "value": "19.5"
                    }
                ]
            },
            {
                "name": "demographic",
                "segment": [
                    {
                        "id": "12341318394918",
                        "name": "auto intenders"
                    }
                ]
            }
        ]
    },
    "tmax": 150,
    "cur": [
        "USD",
```

(continues on next page)

(continued from previous page)


```
        "EUR"
    ],
    "source": {
        "fd": 0
    },
    "ext": {
        "wt": 1,
        "clktrkrq": 0,
        "dooh": {
            "publisher": {
                "id": "rubicon_234563",
                "cat": [
                    "IAB1"
                ]
            },
            "audience": 19.5,
            "impmultiply": 1
        },
        "is_secure": 0,
        "ssp": "rubicon",
        "tgroup": [
            123,
            456,
            321,
            765
        ],
        "ads_txt": {
            "status": 2,
            "auth_id": "1kjgh7653",
            "pub_id": "537120563",
            "supplier_domain": "example.com"
        }
    }
}
```


## 1.13 Bid Response v5.3

This is the top level object that is returned by the Buyer. Each bid response sent from the Buyer to BidSwitch should contain the following fields.

**Note:** Fields marked with an asterisk (*) are optional.

Bid responses for in-app inventory match banner or video response formats.

BidSwitch supports a maximum of 2 bids per ad slot.

Table 77: Bid Response Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | Specifies the ID of the bid request to which this is a response to, for example, `"d7d1e107-fe7c-4a57-9592-d1d41fa702d9"` |
| *seatbid* | *array of objects* | An array of Seat Bid objects, see the *Seat Bid Object* (page 140) section. The length of the array can be either 1+ (for yes-bid) or 0 (for no-bid). |
| *cur** | *string* | Sets the bidding currency using ISO-4217[46] alphabetic codes. If not provided USD is assumed, `"USD"` |
| *ext* | *object* | The bid response extension object. This field may be omitted in No Bid Responses, see the *Bid Response Ext Object* (page 139) section. |
| *nbr** | *integer* | Return a No-bid Reason to BidSwitch. See the valid reasons in the dsp-nbr section |

### 1.13.1 Bid Response Ext Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 78: Bid Response Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *protocol* | *string* | The latest BidSwitch protocol version this bid response is compliant with, for example `"5.3"` |
| *true_price_opt_out** | *boolean* | Prevents True Price from processing the response. If you have True Price enabled for trading, this allows you to bypass it for certain responses or for comparative testing of its efficiency. The following are the valid options:<br>• `1` prevents True Price from processing the response.<br>• `0` or absent means True Price applied if enabled on your account. |

```
{
  "id": "1234567890",
  "ext": {
    "protocol": "5.3",
    "true_price_opt_out": 1
```

(continues on next page)

[46] https://www.iso.org/iso-4217-currency-codes.html

(continued from previous page)

```
  }
}
```

### 1.13.2 Seat Bid Object

Formally there can be multiple bids within the Seat Bid object for two reasons.

- There can be several slots in each bid request.
- There can be more than one bid for a single slot.

BidSwitch allows no more than two bids for a single ad slot. Bids belonging to the same seat must be in the same `seatbid.bid` array, i.e. all `seatbid.seat` values must be unique per response.

Table 79: Seat Bid Object Properties

| Value | Type | Description |
|---|---|---|
| *bid* | *array of objects* | Array of Bid Objects, see *Bid Object* (page 140). The maximum number of bid objects per single bid request ad slot is two. |
| *seat** | string | The seat ID of the bidder on whose behalf this bid is made. The value should match one of the values supplied in the `wseat` field of the bid request. For example, `"34"`.<br>• This field is always required in responses, unless you have your account setup for single-seat responses, see the ag-seat section for more details about that.<br>• Some Suppliers expect their Buyers to respond with their Seat ID rather than BidSwitch's ID, see the ssp-spec-id for a list o those Suppliers. |

**Note:** Fields marked with asterisk (*) are optional.

### 1.13.3 Bid Object

**Note:** (*) Fields marked with an asterisk are usually optional, but may be required for some Suppliers, check for usage notes

Table 80: Bid Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *string* | A bidder generated ID for the bid object, used for tracking and debugging purposes, for example `3`. |
| *impid* | *string* | The ID of the impression object (`imp`) from the bid request to which this bid response applies, for example `"1"` |
| *price* | *float* | The bid price as a float value, expressed as CPM. All prices assumed to be in USD if the `cur` parameter is omitted, for example `1.23` |
| *adm** | *string* | Creative markup for banner ads.<br>• For protocol version 4.x this field should not contain the win price macro.<br>• From version 5.x, this field can contain the win price macro unless the bid request contains `ext.s2s_nurl` field value equal to `1`, see the *5.x Updating Overview* (page 24) section for more information.<br>• This field is required for banner ads, and is ignored for video or native bid responses.<br>• The *adm* field is supported from protocol 4.0, so bid responses containing the `adm` field but not containing the `ext.protocol` value of `4.0+` are deemed invalid.<br>• No more than one win price macro can be used in the `adm` field, otherwise BidSwitch records multiple impression events.<br>• When using an `<iframe>` to include markup and/or pixel trackers, you should close the `</iframe>` properly as not doing so may result in discrepancies on some browsers.<br>`<a href=\"http://adserver.com/click?adid=125&tracker=${CLICK_URL:URLENCODE}\"> <img src=\"http://image1.cdn.com/impid=102\"/></a>` |

continues on next page

Table 80 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *burl* | *string* | Specifies the billing notice URL called by BidSwitch using a server-to-server call when BidSwitch records a billable impression.<br>• Introduced with **v5.2** of the BidSwitch protocol, the `burl` is always a Server-to-Server(s2s) notification, see *Server-to-Server (s2s) Calls* (page 40) for more details.<br>• This field should contain the win price macro, see the *Macros* (page 35) section<br>• See the *Using the burl Field* (page 25) section for more details.<br>• BidSwitch expects that `burl` calls should return a HTTP status `200`, `204`, or `30x`, see the *Server-to-Server (s2s) Calls* (page 40) section for more information.<br>• The field is supported in protocol 5+ versions only.<br>• You can respond with a non-secure `burl` for secure bid requests:<br>`"burl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_PRICE}"` |

continues on next page

Table 80 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *nurl* | *string* | The win notice URL called if the bid wins.<br>• This field should NOT be used for submitting creative markup.<br>• The URL can contain the win price macro, see the *Macros* (page 35) and *5.x Updating Overview* (page 24) sections.<br>• This URL will be mostly called from the user's browser and should thus be SSL-compliant for requests with `imp.secure` set to `1`. But if this URL is to be called using a s2s call as specified in the Bid Request (`imp.ext.s2s_nurl` set to `1`) then is is recommended that it be non-SSL-compliant.<br>• For video responses, you should use the `bid.ext.vast_url` field to pass the VAST URL, see *Video Ext Object* (page 56).<br>• For v5.0 and v5.1, if the Bid Request set the `ext.s2s_nurl` field value to `1` this URL will be called by a s2s call.<br>• BidSwitch expects that `nurl` calls for Bid Requests with `ext.s2s_nurl` set to `1` should return a HTTP status `200`, `204`, or `30x`, see the *Server-to-Server (s2s) Calls* (page 40) section for more information.<br>• As of v5.2, if the bid request set the `ext.s2s_nurl` field value to `1`, only the `burl` field will be called. Therefore, use the `burl` field to pass the win price macro. See the *burl Field Overview* (page 30) section for more details.<br>`http://adserver.com/winnotice?impid=102&winprice=${AUCTION_PRICE}`<br>**Note**: This describes the behaviour in version `4.0+`, which changed since version 2.x. For more information about the 2.x behaviour, see the *nurl Response Difference* (page 22) section. |
| *iurl** | *string* | Sample image URL (without cache busting) for content checking, e.g. `"http://adserver.com/preview?impid=102"`<br>**REQUIRED**: for banner bid requests. |
| *language** | *string* | The Alpha-2 ISO 639-1[47] code for the creative's language, for example, `ja`. The nonstandard code `"xx"` may also be used if the creative has no linguistic content (e.g., a banner with just a company logo).<br>Use this field instead of the deprecated `bid.ext.language` field. |

continues on next page

Table 80 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *adid** | *string* | ID that references the ad to be served if the bid wins. Either the `adid` field or `crid` field should be present in the response, for example `"3021"`<br>**Notes:**<br>• Sometimes a Supplier's CA Service bans creatives that seen in both secure (https) and non-secure (http) forms. Therefore it is recommended to have separate IDs for secure and non-secure versions of the same creative, e.g. `cr_123` & `cr_123_ssl`<br>• When BidSwitch passes the creative ID to the Supplier it prepends the Buyer ID, using the the following syntax: `{DSP_ID}_{Creative_ID}`, for example `70_650029457` or `79_0RsSUxZety0O1`.<br>• When BidSwitch receives both `adid` and `crid` fields, the `adid` ID field is used to pass the creative ID to Suppliers |
| *adomain* | *array of strings* | Advertiser's primary or top-level domain for advertiser checking. This can be a list of domains if there is a rotating creative.<br>**:Note:**<br>• Some Suppliers allow only one domain. To those Suppliers BidSwitch only sends the first domain from the list, for example, `["advertiser.com"]`<br>• BidSwitch invalids non-ascii domains. If you need to pass a domain that uses unicode character such as in the Cyrillic or Japanese script, use punycode[48] |
| *bundle** | *string* | A platform-specific application identifier intended to be unique to the app and independent of the exchange. On Android, this should be a bundle or package name (e.g., com.foo.mygame) |
| *cid** | *string* | Campaign ID or similar that is used by the Buyer to track and organize their campaigns, for example, `102`<br>**REQUIRED** in responses for Rubicon, Nexage, Smaato and MoPub. |

continues on next page

Table 80 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *crid** | *string* | Creative ID to assist with ad quality checking. Either the `adid` field or `crid` field should be present in the response, for example "`3021`"<br>**Notes:**<br>• Sometimes a Supplier's CA Service bans creatives that are seen in both secure (https) and non-secure (http) forms. Therefore it is recommended to have separate IDs for secure and non-secure versions of the same creative, e.g. `cr_123` & `cr_123_ssl`<br>• When BidSwitch passes the creative ID to the Supplier it prepends the Buyer ID, using the the following syntax: `{DSP_ID}_{Creative_ID}`, for example `70_650029457` or `79_0RsSUxZety0O1`.<br>• When BidSwitch receives both `adid` and `crid` fields, the `adid` ID field is used to pass the creative ID to Suppliers |
| *attr** | *array of integers* | Creative attributes as defined in the OpenRTB protocol, for example, `[1,3]`. |
| *dealid** | *string* | Reference to the `deal.id` from the bid request, if this bid pertains to a private marketplace direct deal, for example, `"AA-1234"` |
| *h** | *integer* | The height of the creative in pixels when an alternative ad size is used, relevant for banner ads only. `250`. **Required:** For Pubmatic Banner ads. |
| *w** | *integer* | The width of the creative in pixels when an alternative ad size is used, relevant for banner ads only. `300`. **Required:** For Pubmatic Banner ads. |
| *cat** | *array of strings* | The IAB category of the creative.<br>**REQUIRED** in bid responses to MoPub, Smaato and YieldOne bids. If the Supplier only accepts one category in the bid response the first array element will be used, for example, `["IAB1"]`<br>**Note:** For backward compatibility, this field can also be a string when using the BidSwitch 4.0 protocol. |
| *ext** | object | This field may be required under certain circumstances, see *Bid Ext Object* (page 146). |

[47] https://www.iso.org/iso-639-language-codes.html
[48] https://www.punycoder.com/

### Bid Ext Object

Table 81: Bid Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *at1** | *int* | Indicates that the Buyer wishes their bid to be used in the Supplier 1st price auction before being passed to any upstream auction e.g. header bidding. This field currently only takes the following value:<br>• `1` indicates that the field should be included in the first-price auction before being passed further upstream<br>• This field is only valid with the following Supplier: Nexage (a.k.a Millennial Media in the myBidSwitch UI) |
| *asid** | *string* | Required only for Microad premium inventor responses. If you are using a 3rd party ad server you must specify which one, for example, `"Sizmek/Sizmek"`. See the *MicroAd 3PAS List* (page **??**) section for more information |
| *country** | *string* | Required only for Microad premium inventory responses and uses ISO 3166-1 Alpha-3 country codes, for example `JPN`. Specifies the target country of the Ad campaign. If you have multiple GEO targets, set the main one here. |
| *advertiser_name** | *string* | The name of the advertiser serving the creative, for example, `"Coca-Cola"`<br>• **REQUIRED** in bid responses to Ströer (AdScale), Centro, and BRX.<br>• Recommended in responses to YieldOne bids. |
| *agency_name** | *string* | The name of the agency representing the advertiser, for example, `"CCA"`<br>**REQUIRED** in bids responses to Ströer (AdScale) bids. |
| *agency_id** | *string* | ID of the agency representing the advertiser, for example, ‘‘`123`’’ |
| *lpdomain** | *array of strings* | The actual landing page URL of the creative. We highly recommend that you always fill this field, especially for mobile application ads, and for all Google responses.<br>**Required**: for Xandr (Appnexus) mobile ads<br>• `"adomain":["angrybirds.com"]`<br>• `"lpdomain":["https://itunes.apple.com/ru/app/angry-birds/id343200656?mt=8", "https://play.google.com/store/apps/details?id=com.rovio.angrybirds"]` |
| *data** | *array of object* | Returns arbitrary data to the Supplier, each object can take **data.name** and **data.value** to describe the data, see the *Data Response Object* (page 157) for more details. |

continues on next page

Table 81 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *language** | *string* | The Alpha-2 ISO 639-1[49] code for the creative's language, for example, `jp`.<br>Deprecated since version 5.2.: Use `seatbid.bid.language` instead. |
| *google** | *object* | Contains additional information for Google bids. This field is recommended. See the *Supplier Specific Fields* (page 157) section for more details. |
| *yieldone** | *object* | Contains additional information for YieldOne bids. This field is recommended. See the *Supplier Specific Fields* (page 157) section for more details. |
| *vast_url** | *string* | The URL pointing to the location of the VAST document for bid responses to video traffic, for example, `"http://adserver.com/vast?impid=102"`<br>• **Required** if the `video.ext.vast_url_rq` bid request field is set to `1`.<br>• If the `video.ext.vast_url_rq` bid request field is set to `0` or missing, you can include the VAST URL in the `nurl` field.<br>For more information see the *Video Ext Object* (page 56) section.<br>**Note:**<br>• The VAST URL should NOT contain a win price macro.<br>• The VAST document should NOT contain impression tracking URLs with win price macros. |
| *daast_url** | *string* | The URL pointing to the location of the DAAST document for the bid response, for example, `"http://adserver.com/daast?impid=102"`<br>**REQUIRED** for bid responses to audio traffic.<br>**Note:**<br>• The DAAST URL should NOT contain a win price macro.<br>• The DAAST document should NOT contain impression tracking URLs with win price macros. |
| *duration** | *integer* | Video ad duration in seconds, for example, `13`<br>**REQUIRED** in bid responses to BrightRoll Video (brx). |
| *native** | *object* | Contains the details of the native response, for more information, see *Native Response Object* (page 150). |
| *deal** | *string* | This is the ID of the deal between a publisher and a seat. It is used only if an exchange supports private auctions.<br>If the bid is associated with a direct deal then this field is required and its value should be equal to one of the elements in the `pmp.deals` field in the bid request object.<br>Deprecated since version 2.5: use `seatbid.bid.dealid` instead. |

continues on next page

Table 81 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *img_url** | *string* | The URL of the creative image. In order to receive the user cookie and win price, this URL should point to the Buyer handler and redirect to the actual creative location. The url may contain the win price macro, e.g. `${AUCTION_PRICE}`, but not the click macro.<br>If this field is present, the `nurl` field of the bid response will be ignored.<br>**Note**: This field is only valid in 2.x bid responses, see the *Deprecated 2.x Properties* (page 22) section for more details. |
| *click_url** | *string* | The creative click URL. Required if the `img_url` field is present.<br>**Note**: This field is only valid in 2.x bid responses, see the *Deprecated 2.x Properties* (page 22) section for more details. |
| *js_url** | *string* | A Javascript-based win notice URL.<br>• For in-app inventory, the ad markup should be returned using this URL.<br>• For website or video inventory this field may be used as a substitute for the `nurl` field.<br>• Ad markup should be in JavaScript format.<br>• The URL may contain macros, see the *Macros* (page 35) section for more details.<br>**Note**: This field is only valid in 2.x bid responses, see the *Deprecated 2.x Properties* (page 22) section for more details. |
| *skadn** | *object* | Apple Ad Network Object, this will be used to pass app data from iOS 14 and newer releases. See *SkAdNetwork Extension* (page 148) |

### SkAdNetwork Extension

Table 82: skadn Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *version** | *str* | Version of SKAdNetwork desired. Must be "2.0" or above. From SKAdNetwork v2.2 onwards, this should be used in the `fidelities` object. |
| *network** | *str* | Ad network identifier used in signature. Should match one of the items in the `imp.ext.skadnetids` array in the request |
| *campaign** | *str* | Campaign ID compatible with Apple's spec. As of 2.0, this should be an integer between 1 and 100, expressed as a string, e.g. `"45"` |

continues on next page

[49] https://www.iso.org/iso-639-language-codes.html

Table 82 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *fidelities** | *array of objects* | Supports multiple fidelity types introduced in SKAdNetwork v2.2, see the *SkAdNetwork Fidelities* (page 149) object for details.<br><br>**Note:** From SKAdNetwork v2.2 onwards, this object wraps some of the other fields in this table into it. As a result, `nonce`, `version`, `timestamp` and `signature` should be used in the `fidelities` object and considered deprecated in this object. |
| *itunesitem** | *str* | ID of advertiser's app in Apple's app store. Should match the `app.bundle` request field e.g `"880047117"` |
| *nonce** | *str* | An ID unique to each ad response (GUID/UUID) e.g. `"beeeb65e-b3de-02420004"`. From SKAdNetwork v2.2 onwards, this should be used in the `fidelities` object. |
| *sourceapp** | *str* | ID of publisher's app in Apple's app store, this should match the `imp.ext.skadn.sourceapp` value |
| *timestamp** | *str* | Unix time in millis string used at the time of signature. From SKAdNetwork v2.2 onwards, this should be used in the `fidelities` object. |
| *signature** | *str* | SKAdNetwork signature as specified by Apple e.g. `"MEQCIEQZRRyMyUXg=="`. From SKAdNetwork v2.2 onwards, this should be used in the `fidelities` object. |

### SkAdNetwork Fidelities

Table 83: Fidelities Object Properties

| Value | Type | Description |
|---|---|---|
| *version** | *str* | Version of SKAdNetwork desired. Must be "2.0" or above. |
| *nonce** | *str* | An ID unique to each ad response (GUID/UUID) e.g. `"beeeb65e-b3de-02420004"` |
| *timestamp** | *str* | Unix time in millis string used at the time of signature |
| *signature** | *str* | SKAdNetwork signature as specified by Apple e.g. `"MEQCIEQZRRyMyUXg=="` |

**Note:** Some of the fields are required by certain Suppliers. Responses to bid requests from these Suppliers without the required fields will be discarded.

#### Required Bid Response Fields Per Supplier

Table 84: Required Bid Response Fields Per Supplier

| Supplier | Required field(s) |
|---|---|
| Fyber | `seatbid.bid.bundle` |
| MicroAd (For Premium Inventory) | `seatbid.bid.cat`, `seatbid.bid.ext.country`, `seatbid.bid.ext.asid`. See also the *MicroAd 3PAS List* (page **??**) section. |
| Millennial Media (nexage) | `cid` |
| MoPub | `seatbid.bid.cid`, `seatbid.bid.cat` |
| PubMatic | `seatbid.bid.w`, `seatbid.bid.h` required for banner ads |
| Rubicon | `seatbid.bid.cid` |
| Smaato | `seatbid.bid.cid`, `seatbid.bid.cat` |
| Ströer (adscale) | `seatbid.bid.ext.advertiser_name`, `seatbid.bid.ext.agency_name` |
| TrustX | `seatbid.bid.burl`, `seatbid.bid.nurl` |
| Xandr (AppNexus) mobile ads | `seatbid.bid.ext.lpdomain` |
| YieldOne | `seatbid.bid.cat` |

### 1.13.4 Native Response Object

**Note:** Fields marked with an asterisk (*) are optional.

Table 85: Native Object Properties

| Value | Type | Description |
|---|---|---|
| *assets* | *array of objects* | List of native ad assets. |
| *link* | *object* | The *Native Link Object* (page 153). This is the default link object for the ad. Individual assets can also have a link object which applies if the asset is activated (clicked). If the asset has no link object, the parent link object applies. |
| *ext** | *object* | *Native Ext Object* (page 151) |
| *ver** | *string* | Version of the Native Markup version in use, for example, `"1"`. |
| *privacy** | *string* | If support was indicated in the request, return the URL of a page informing the user about the buyer's targeting activity. |
| *eventtrackers** | *array of objects* | Specifies what type of event tracking is supported, see *Event Tracker Response Object* (page 155) |

continues on next page

Table 85 – continued from previous page

| Value | Type | Description |
|---|---|---|
| *imptrackers* ** | *array of strings* | Deprecated since version 5.3: replaced by the `eventtrackers` field<br>An array of impression tracking URLs, expected to return a 1x1 image or 204 response, for example, `["http://adserver.com/native?impid=102"]`. This URL can contain the `win_price` macro. |
| *jstracker** | *string* | Deprecated since version 5.3: replaced by the `eventtrackers` field<br>Optional JavaScript impression tracker. This should be valid HTML with JavaScript already wrapped in `<script>` tags. It will be executed at impression time where it can be supported. **Note:** Currently the only Supplier that supports this field is TripleLift. |

**Note:** (**) imptrackers array should be used with following constraints:

For Rubicon, only one element is guaranteed to be triggered. Place the trackers in the order of importance, otherwise this array is optional and may not be present in each response.

### Native Ext Object

Table 86: Native Ext Object Properties

| Value | Type | Description |
|---|---|---|
| *viewtracker** | *string* | The view tracking URL that will be called when the ad is visible, if supported by the Supplier, for example, `"http://adserver.com/native_view?impid=102"` |
| *adchoiceurl** | *string* | A Buyer specific AdChoices URL that will replace default Supplier AdChoices URL, for example, `"https://adserver.com/privacy"`. |

### Native Assets Object

**Note:**

- (*) There may be exactly one of the fields marked with asterisk in one asset object.
- (**) The link object is optional and may not be present in each response.

Table 87: Native Asset Object Properties

| Value | Type | Description |
|---|---|---|
| *id* | *integer* | A unique asset ID, must match one of the asset IDs in the bid request, for example, `1`. |
| *required** | *integer* | Set to `1` if the asset is required (bidder requires it to be displayed), default is `0`, for example, `1`. |
| *title** | *object* | Title object for a title asset, see, *Native Assets Title Object* (page 152). |
| *img** | *object* | Image object for an image asset, see, *Native Assets Image Object* (page 152). |
| *video** | *object* | Video object for a video asset, see, *Native Asset Video Object* (page 153). |
| *data** | *object* | Data object for a data asset, see, *Native Asset Data Object* (page 153). |
| *link* ** | *object* | Link object for a call to action.<br>• The link object applies if the asset item is activated (clicked).<br>• If there is no link object on the asset, the parent link object on the bid response applies. See *Native Link Object* (page 153). |

### Native Assets Title Object

Table 88: Native Asset Title Object Properties

| Value | Type | Description |
|---|---|---|
| *text** | string | The text associated with the title element. `"Our product is the best!"` |

### Native Assets Image Object

Table 89: Native Asset Image Object Properties

| Value | Type | Description |
|---|---|---|
| *url* | *string* | (**Required**) URL of the image asset, for example, `"http://adserver.com/image?impid=102"`. |
| *h* | *integer* | (**Recommended**) Height of the image in pixels, for example, `250`. |
| *w* | *integer* | (**Recommended**) Width of the image in pixels, for example, `300`. |

**Important:** If the Supplier specifies the exact w/h or the wmin/hmin in the bid request, the bid response must contain the `w/h` values and these fields become required, see the relevant request field

for details *Native Asset Image Object* (page 65).

### Native Asset Video Object

Table 90: Native Asset Video Object Properties

| Value | Type | Description |
|---|---|---|
| *vasttag* | *string* | Vast XML, use the following example to format your VAST XML response. See the VAST Tag example below, *Video Asset vasttag Example* (page 155). |
| *ext** | *object* | *Native Asset Video Object Extension* (page 153) |

### Native Asset Video Object Extension

Table 91: Native Asset Video Object Extension Properties

| Value | Type | Description |
|---|---|---|
| *playback-method** | *integer* | Desired video playback method |

### Native Asset Data Object

Table 92: Native Asset Data Object Properties

| Value | Type | Description |
|---|---|---|
| *value* | *string* | The formatted string of data to be displayed. Can contain a formatted value such as ‘‘`5 stars`’’ or ‘‘`$10`’’ or ‘‘`3.4 stars out of 5`’’. |

### Native Link Object

Table 93: Native Link Object Properties

| Value | Type | Description |
|---|---|---|
| *url* | *string* | Landing URL of the clickable link, for example, `"http://advertiser.com/"` |
| *clicktrackers** | *array of strings* | Click tracker URLs to be activated when the URL is clicked, for example, `["http://adserver.com/click?impid=102"]` |

### Native Response Example


```
{
   "seatbid":[
      {
         "bid":[
            {
               "ext":{
                  "native":{
                     "ver":"1.2",
                     "eventtrackers":[
                        {
                           "event":1,
                           "method":2,
                           "url":"http://www.mytracker.com/tracker.js"
                        },
                        {
                           "event":2,
                           "method":1,
                           "url":"http://www.example.com/tracker.php"
                        }
                     ],
                     "privacy":"privacy-example.com",
                     "link":{
                        "url":"http://adserver.com/click?impid=102"
                     },
                     "assets":[
                        {
                           "id":1,
                           "video":{
                              "ext":{
                                 "playbackmethod":1
                              }
                           },
                           "required":1,
                           "title":{
                              "text":"A test Native Ad"
                           }
                        }
                     ]
                  }
               }
            }
         ]
      }
   ]
}
```

### Video Asset vasttag Example

```
<?xml version="1.0" encoding="UTF-8"?>
<VAST version="2.0">
   <Ad id="12345">
      <InLine>
         <AdSystem version="1.0">SpotXchange</AdSystem>
         <AdTitle><![CDATA[Sample VAST]]></AdTitle>
         <Impression>http://sample.com</Impression>
         <Description><![CDATA[A sample VAST feed]]></Description>
         <Creatives>
            <Creative sequence="1" id="1">
               <Linear>
                  <Duration>00:00:30</Duration>
                  <TrackingEvents />
                  <VideoClicks>
                     <ClickThrough><![CDATA[http://sample.com/openrt btest]]>
                     </ClickThrough>
                  </VideoClicks>
                  <MediaFiles>
                     <MediaFile delivery="progressive" bitrate="256"
                     width="640" height="480" type="video/mp4">
                     <![CDATA[http://sample.com/video.mp4]]>
                     </MediaFile>
                  </MediaFiles>
               </Linear>
            </Creative>
         </Creatives>
      </InLine>
   </Ad>
</VAST>
```

### Event Tracker Response Object

The event trackers response is an array of objects and specifies the types of events the bidder wishes to track and the URLs/information to track them. Buyers must only respond with methods indicated as available in the request.

---

**Note:** Most javascript trackers expect to be loaded at impression time, so it's not generally recommended for the Buyer to respond with javascript trackers on other events, but the appropriateness of this is up to each Buyer.

---

Table 94: Event Tracker Response Object

| Value | Type | Description |
|---|---|---|
| *event* | integer | Type of event to track, see the *Event Tracking Types* (page 69) table |
| *method* | integer | Type of method to track, see the *Event Tracking Methods* (page 69) table |
| *url** | string | The URL of the impage or js. Required for image or js, optional for custom. This value can contain the `win_price` macro. |
| *customdata** | object | containing `key:value` pairs - To be agreed individually with the exchange, an array of `key:value` objects for custom tracking, for example the account number of the DSP with a tracking company. IE {"accountnumber":"123"}. |
| *ext** | object | This object is a placeholder that may contain custom JSON agreed to by the parties to support flexibility beyond the standard defined in this specification |

### Event Tracker Response Example

```
{
   "eventtrackers":[
      {
         "event":1,
         "method":2,
         "url":"http://www.mytracker.com/tracker.js"
      },
      {
         "event":2,
         "method":1,
         "url":"http://www.example.com/tracker.php"
      }
   ]
}
```

### 1.13.5 Supplier Specific Fields

**Note:** Fields marked with an asterisk (*) are optional.

#### Google Object Properties

Table 95: Google Object Properties

| Value | Type | Description |
|---|---|---|
| *attribute** | *integer array* | List of creative attributes as defined by Google, for example, `[32, 22]` |
| *vendor_type** | *integer array* | List of creative vendor types as defined by Google, for example, `[42, 43]`<br>See the Google RTB Docs Download[50] page for this information. |

#### Yieldone Object Properties

Table 96: Yieldone Object Properties

| Value | Type | Description |
|---|---|---|
| *creative_type** | *string* | Creative type as defined by YieldOne, for example, `"HTML"` |
| *creative_category_id** | *integer* | Creative category as defined by YieldOne. The field is recommended for YieldOne bids, for example, `79` |

### 1.13.6 Data Response Object

Can be used to return arbitrary data to your Suppliers, if they support this field.

Table 97: Bid Response Data Extension Object

| Value | Type | Description |
|---|---|---|
| *name* | *string* | Used to specify the name of the entity to which the value refers, e.g. `"Scan Code"` |
| *value* | *string* | The value for the named data type being returned to the Supplier, e.g. `"1iuyyw-987762"` |

[50] https://developers.google.com/ad-exchange/rtb/data

Listing 2: Example Response with Data

```
{
 "seatbid":{
     "bid":{
         "ext":{
             "data":[
                 {
                     "name":"vdeal-id",
                     "value":"abc123"
                 },
                 {
                     "name":"dmp-data",
                     "value":"56752-86862"
                 }
             ]
         }
     }
 }
}
```

### 1.13.7 Bid Response JSON Examples

If the Buyer opts to not bid on the bid request, it should respond with a HTTP 204 response. BidSwitch also accepts bid responses with an empty array of seatbid objects as valid nobid responses.

#### Banner Bid Response

The following example shows an ad being served from the `adm` field, with the bid price for the impression being $9.43 CPM.

```
{
  "id": "1234567890",
  "ext": {
    "protocol": "5.3"
  },
  "seatbid": [
    {
      "bid": [
        {
          "id": "1",
          "impid": "102",
          "price": 9.43,
          "adid": "314",
          "cid": "42",
```

(continues on next page)

(continued from previous page)

```
            "cat": ["IAB12"],
            "language": "en",
            "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}",
            "adm": "<a href=\"http://adserver.com/click?adid=12345&tracker=${CLICK_
↪URL:URLENCODE}\"><img src=\"http://image1.cdn.com/impid=102\"/></a>",
            "nurl": "http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
            "iurl": "http://adserver.com/preview?crid=314",
            "adomain": [
              "advertiserdomain.com"
            ],
            "ext": {
              "advertiser_name": "Coca-Cola",
              "agency_name": "CC-advertising"
            }
          }
        ],
        "seat": "4"
      }
    ]
}
```

### Secure Banner Bid Response

The following example shows an ad being served from the `adm` field, with the bid price for the impression being $9.43 CPM, and suitable for serving in HTTPS environment.

```
{
  "id":"1234567890",
  "ext":{
     "protocol":"5.3"
  },
  "seatbid":[
     {
        "bid":[
           {
              "id":"1",
              "impid":"102",
              "price":9.43,
              "adid":"314",
              "cid":"42",
              "language": "en",
              "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
              "cat":[
```

(continues on next page)

(continued from previous page)

```
                "IAB12"
            ],
            "adm":"<a href=\"https://adserver.com/clickadid=12345&tracker=$
↪{CLICK_URL:URLENCODE}\"><img src=\"https://image1.cdn.com/impid=102\"/></a>",
            "nurl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
            "iurl":"http://adserver.com/preview?crid=314",
            "adomain":[
                "advertiserdomain.com"
            ],
            "ext":{
                "advertiser_name":"Coca-Cola",
                "agency_name":"CC-advertising"
            }
        }
      ],
      "seat":"8"
    }
  ]
}
```

### Response to iOS 14

**Note:** See the highlighted lines that denote the iOS 14 fields, see the *Protocol Release Notes* (page 2) for more details.

```
{
  "id": "1234567890",
  "ext": {
    "protocol": "5.3"
  },
  "seatbid": [
    {
      "bid": [
        {
          "id": "1",
          "impid": "102",
          "price": 9.43,
          "adid": "314",
          "cid": "42",
          "language": "en",
          "burl": "https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}",
          "cat": [
```

(continues on next page)

(continued from previous page)


```
            "IAB12"
          ],
          "adm": "<a href=\"https://adserver.com/clickadid=12345&tracker=${CLICK_
↪URL:URLENCODE}\"><img src=\"https://image1.cdn.com/impid=102\"/></a>",
          "nurl": "https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
          "iurl": "http://adserver.com/preview?crid=314",
          "adomain": [
            "advertiserdomain.com"
          ],
          "ext": {
            "advertiser_name": "Coca-Cola",
            "agency_name": "CC-advertising",
            "data": [
              {
                "name": "TUV",
                "value": "ABC123"
              }
            ],
            "skadn": {
              "network": "cDkw7geQsH.skadnetwork",
              "campaign": "45",
              "itunesitem": "880047117",
              "sourceapp": "123456789",
              "fidelities": [
                {
                  "timestamp": "1594406341",
                  "signature": "2RmFS7daRzSVZRVZ8RyMyUXg==",
                  "nonce": "473b1a16-b4ef-43ad-9591-fcf3aefa82a7",
                  "version": "2.2"
                }
              ]
            }
          }
        }
      ],
      "seat": "8"
    }
  ]
}
```

### Video Bid Response


```
{
   "cur":"USD",
   "ext":{
      "protocol":"5.3"
   },
   "id":"e9c3e120-ffcb-4300-9c98-644cb26f95df",
   "seatbid":[
      {
         "bid":[
            {
               "adid":"3",
               "nurl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
               "adomain":[
                  "nokia.com"
               ],
               "cid":"11",
               "language": "en",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
               "ext":{
                  "vast_url":"http://adserver.com/vast?impid=102",
                  "duration":15,
                  "advertiser_name":"Nokia"
               },
               "id":"1c3ff810-3623-4b04-8396-9e7ca071cb72",
               "impid":"1",
               "price":4.079077199308326
            }
         ],
         "seat":"1"
      }
   ]
}
```


### Secure Video Bid Response


```
{
  "cur":"USD",
  "ext":{
     "protocol":"5.3"
  },
  "id":"e9c3e120-ffcb-4300-9c98-644cb26f95df",
```


(continues on next page)

(continued from previous page)

```
  "seatbid":[
     {
        "bid":[
           {
              "adid":"3",
              "language": "en",
              "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
              "nurl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
              "adomain":[
                 "nokia.com"
              ],
              "cid":"11",
              "ext":{
                 "vast_url":"https://adserver.com/vast?impid=102",
                 "duration":15,
                 "advertiser_name":"Nokia"
              },
              "id":"1c3ff810-3623-4b04-8396-9e7ca071cb72",
              "impid":"1",
              "price":4.079077199308326
           }
        ],
        "seat":"1"
     }
  ]
}
```

### Secure Audio Bid Response

```
{
  "cur":"USD",
  "ext":{
     "protocol":"5.3"
  },
  "id":"e9c3e120-ffcb-4300-9c98-644cb26f95df",
  "seatbid":[
     {
        "bid":[
           {
              "adid":"3",
              "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
```

(continues on next page)

(continued from previous page)

```
                "nurl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
                "adomain":[
                   "nokia.com"
                ],
                "cid":"11",
                "language": "en",
                "ext":{
                   "daast_url":"https://adserver.com/daast?impid=102",
                   "duration":15,
                   "advertiser_name":"Nokia"
                },
                "id":"1c3ff810-3623-4b04-8396-9e7ca071cb72",
                "impid":"1",
                "price":4.079077199308326
             }
          ],
          "seat":"1"
       }
    ]
}
```

### Native Bid Response

The following example shows a native bid response

```
{
   "id":"1234567890",
   "ext":{
      "protocol":"5.3"
   },
   "seatbid":[
      {
         "bid":[
            {
               "id":"1",
               "impid":"102",
               "price":9.43,
               "adid":"314",
               "cid":"42",
               "language":"en",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
               "cat":[
                  "IAB12"
```

(continues on next page)

(continued from previous page)


```
                ],
                "nurl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
                "adomain":[
                   "advertiserdomain.com"
                ],
                "ext":{
                   "advertiser_name":"Coca-Cola",
                   "agency_name":"CC-advertising",
                   "native":{
                      "ver":"1",
                      "eventtrackers":[
                         {
                            "event":1,
                            "method":2,
                            "url":"http://www.mytracker.com/tracker.js"
                         },
                         {
                            "event":2,
                            "method":1,
                            "url":"http://www.example.com/tracker.php"
                         }
                      ],
                      "link":{
                         "url":"http://adserver.com/click?impid=102"
                      },
                      "assets":[
                         {
                            "id":1,
                            "required":1,
                            "title":{
                               "text":"A test Native Ad"
                            }
                         },
                         {

                         }
                      ]
                   }
                }
             }
          ],
          "seat":"58"
       }
   ]
}
```

### Secure Native Bid Response

The following example shows a native bid response with the title and image asset specified for serving in an HTTPS environment.

```
{
   "id":"1234567890",
   "ext":{
      "protocol":"5.3",
      "id":3,
      "required":1,
      "img":{
         "url":"https://adserver.com/image?crid=314"
      }
   },
   "seatbid":[
      {
         "bid":[
            {
               "id":"1",
               "impid":"102",
               "price":9.43,
               "adid":"314",
               "cid":"42",
               "cat":[
                  "IAB12"
               ],
               "nurl":"https://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
               "adomain":[
                  "advertiserdomain.com"
               ],
               "language":"en",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
               "ext":{
                  "advertiser_name":"Coca-Cola",
                  "agency_name":"CC-advertising",
                  "native":{
                     "ver":"1",
                     "link":{
                        "url":"https://adserver.com/click?impid=102"
                     },
                     "eventtrackers":[
                        {
                           "event":1,
                           "method":2,
```

(continues on next page)

(continued from previous page)

```
                        "url":"http://www.mytracker.com/tracker.js"
                     },
                     {
                        "event":2,
                        "method":1,
                        "url":"http://www.example.com/tracker.php"
                     }
                  ],
                  "assets":[
                     {
                        "id":1,
                        "required":1,
                        "title":{
                           "text":"A test Native Ad"
                        }
                     },
                     {

                     }
                  ]
               }
            }
         }
      ],
      "seat":"1"
   }
]
}
```

### Multi-bid Response

Please pay attention to the following conditions:

- The total size of your response should not exceed 50 kb
- There can be 2 `seatbid.bid` objects within 1 `seatbid` object
- A response containing two bids for the same impression should contain both `bid` objects within the same `seatbid.bid` array of objects

The following bid response example contains two bids, one for $9.43 CPM and one for $5.50 CPM. Both bids target the same ad slot, both would take part in the auction and if the first one is discarded due to publisher-side blocklist then the second one would be able to win the auction.

```
{
   "id":"1234567890",
   "ext":{
```

(continues on next page)

(continued from previous page)


```
      "protocol":"5.3"
   },
   "seatbid":[
      {
         "bid":[
            {
               "id":"1",
               "impid":"102",
               "price":9.43,
               "adid":"314",
               "cid":"42",
               "cat":[
                  "IAB12"
               ],
               "adm":"<a href=\"http://adserver.com/click?adid=12345&tracker=$
↪{CLICK_URL:URLENCODE}\"><img src=\"http://image1.cdn.com/impid=102\"/></a>",
               "nurl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
               "iurl":"http://adserver.com/preview?crid=314",
               "adomain":[
                  "advertiserdomain.com"
               ],
               "language":"en",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_
↪PRICE}",
               "ext":{
                  "advertiser_name":"Coca-Cola",
                  "agency_name":"CC-advertising"
               }
            },
            {
               "id":"2",
               "impid":"102",
               "price":5.5,
               "adid":"413",
               "cid":"43",
               "cat":[
                  "IAB12"
               ],
               "adm":"<a href=\"http://adserver.com/click?adid=12345&tracker=$
↪{CLICK_URL:URLENCODE}\"><img src=\"http://image1.cdn.com/impid=102\"/></a>",
               "nurl":"http://adserver.com/winnotice?impid=413&winprice=${AUCTION_
↪PRICE}",
               "iurl":"http://adserver.com/preview?crid=413",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_
↪PRICE}",
```


(continues on next page)

(continued from previous page)

```
                "adomain":[
                   "advertiserdomain.com"
                ],
                "ext":{
                   "advertiser_name":"Coca-Cola",
                   "agency_name":"CC-advertising"
                }
             }
          ],
          "seat":"1"
       }
    ]
 }
```

### Private Deal Bid Response

```
{
  "id":"1234567892",
  "ext":{
     "protocol":"5.3"
  },
  "cur":"USD",
  "seatbid":[
     {
        "bid":[
           {
              "id":"1114125-afaff2f-af2251",
              "impid":"1",
              "price":9.43,
              "adid":"314",
              "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
              "dealid":"deal-1",
              "adm":"<a href=\"http://adserver.com/click?adid=12345&tracker=$
↪{CLICK_URL:URLENCODE}\"> <img src=\"http://image1.cdn.com/impid=102\"/></a>",
              "nurl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
              "adomain":[
                 "advertiserdomain.com"
              ]
           }
        ],
        "seat":"58"
     }
```

(continues on next page)

(continued from previous page)

```
  ]
}
```

### Buyer No Bid Response

The following example shows a No Bid Response coming from a Buyer, which is an empty bid response.

**Note:** The preferred No Bid Response format is an empty HTTP 204 response.

```
{
  "id": "1234567891",
  "ext": {
    "protocol": "5.3"
  },
   "seatbid": []
}
```

### No Bid Reason

The following example shows a No Bid Reason response, which includes the reason code in the `nbr` field. See the dsp-nbr section for more information.

```
{
   "id":"1234567890",
   "seatbid":[

   ],
   "nbr":11
}
```

### TV/DOOH Bid Response

```
{
   "cur":"USD",
   "ext":{
      "protocol":"5.3"
   },
   "id":"e9c3e120-ffcb-4300-9c98-644cb26f95df",
   "seatbid":[
      {
```

(continues on next page)

(continued from previous page)


```
         "bid":[
            {
               "adid":"3",
               "burl":"https://adserver.com/imp?impid=102&winprice=${AUCTION_PRICE}
↪",
               "nurl":"http://adserver.com/winnotice?impid=102&winprice=${AUCTION_
↪PRICE}",
               "adomain":[
                  "nokia.com"
               ],
               "cid":"11",
               "language":"en",
               "ext":{
                  "vast_url":"http://adserver.com/vast?impid=102",
                  "duration":15,
                  "advertiser_name":"Nokia"
               },
               "id":"1c3ff810-3623-4b04-8396-9e7ca071cb72",
               "impid":"1",
               "price":4.079077199308326
            }
         ],
         "seat":"1"
      }
   ]
}
```