

# Native Ad Markup Bidding (Beta)

## Native Ad Markup Bidding (Beta)

Native Ad Markup Bidding enables your bidder to submit native ad markup via `adm` in the OpenRTB bid response. Instead of registering every native creative asset you have with AppNexus, you are required to register only a single creative for each ad campaign or brand you represent; all native creative assets for that brand can be passed dynamically via ad markup in the bid response. This page explains how to use this feature.

**Note:** This feature does not currently support bidding with native video ad markup. Only non-video native ad markup is accepted.

### On This Page

- [Getting Started](#)
- [Register Native Creatives](#)
- [Creative Example](#)
- [Bid with Native Ad Markup](#)
- [Bid Response Supported Fields](#)
- [Bid Response Example](#)
- [FAQs](#)

## Getting Started

Your bidder must be enabled to use this feature. If you are not sure whether your bidder is enabled, please check with your AppNexus account representative.

Once enabled, there are two steps required to buy Native inventory via Ad Markup Bidding:

1. Register your creative. For each brand that you represent, you must register a single creative that represents that brand. Your creative must pass platform audit. All native creative assets associated with this brand within your bidder will serve through this creative. See [Register Native Creatives](#) below for details.
2. Bid with your creative assets on native inventory. See [Bid with Native Ad Markup](#) below for details.

## Register Native Creatives

For each brand that you represent, you must register a single creative that represents that brand. This is done using the [Creative Service](#). When registering a Native Creative, there are a few considerations to keep in mind:

- The creative must represent one of the actual native ads that you will dynamically pass on the bid response for this brand. The specific ad you choose to register does not matter, but the creative must contain at least one image asset and one data asset, consistent with an ad you would actually serve on native inventory.
- When registering a creative, only [AppNexus Macros](#) are supported. OpenRTB macros (such as `${AUCTION_PRICE}`) will not be expanded.
- The creative must be submitted for platform audit.
- You do not need to specify the `brand_id` field; this will be set by AppNexus during audit.
- Be sure to include impression and click trackers when registering your creative. The ad markup submitted in the bid response should use the same set of vendors (or fewer) that were registered with this creative
- Be sure to use native creative template 39461.

The following section provides an example of defining and registering a Native Creative.

## Creative Example

This example uses four data assets and two image assets, but you can choose to use a different combination depending on the assets you want to register. (Remember, you must have at least one asset of each type.) See the [Creative Service](#) for more details on native creative assets.

### Adding a native creative

```
$ cat native_creative.json
```

```
{
  "creative": {
    "description": "test description",
    "code": "test_code",
    "code2": "test_code2",
    "allow_audit": 1,
    "template": {
      "id": 39461
    },
    "native_attribute": {
      "link": {
        "url": "https://www.landingpage.com",
        "fallback_url": "http://fallback.com",
        "trackers": [{
          "url": "http://url.com",
          "url_secure": "https://secureurl.com"
        }]
      },
      "data_assets": [
        {
          "data_type": "title",
          "value": "Test Title"
        },
        {
          "data_type": "sponsored_by",
          "value": "Test Sponsor"
        },
        {
          "data_type": "description",
          "value": "Test description."
        },
        {
          "data_type": "call_to_action",
          "value": "Test CTA!"
        }
      ],
      "image_assets": [
        {
          "image_type": "main_image",
          "creative_asset_image": {
            "url": "http://url.com/main_img.jpg",
            "url_secure": "https://secureurl.com/main_img.jpg",
            "height": 1200,
            "width": 628
          }
        },
        {
          "image_type": "icon_image",
          "creative_asset_image": {
            "url": "http://url.com/icon_img.jpg",
            "url_secure": "https://secureurl.com/icon_img.jpg",
            "height": 150,
            "width": 150
          }
        }
      ],
      "image_trackers": [{
        "url": "http://url.com",

```

```

        "url_secure": "https://secureurl.com"
    }],
    "javascript_trackers": [{
        "url": "http://url.com",
        "url_secure": "https://secureurl.com"
    }]
}
}
}

$ curl -b cookies -c cookies -X POST -s @native_creative.json
'http://api.adnxs.com/creative/123'

{
  "response": {
    "status": "OK",
    "id": 12345,
    "count": 1,
    "start_element": 0,
    "num_elements": 100,
    "creative": {
      "id": 12345,
      "active": true,
      "member_id": 123,
      "description": "test description",
      "code": "test_code",
      "code2": "test_code2",
      "audit_status": "pending",
      "allow_audit": true,
      "ssl_status": "pending",
      "allow_ssl_audit": true,
      "template": {
        "id": 39461
      },
      "native_attribute": {
        ...
      },
      ...
    },
    "dbg": {
      ...
    }
  }
}

```

## Bid with Native Ad Markup

Note: This feature does not currently support bidding with native video ad markup. Only non-video native ad markup is accepted.

Once you have registered a Native Creative and it has passed platform audit, you can begin bidding with that creative using the [OpenRTB protocol](#). Note that AppNexus supports v1.1 of the [OpenRTB Dynamic Native Ads API](#). Some considerations:

- Either the `crid` or `adid` field must be included in the bid response to identify the creative that was registered. The `crid` value must match the creative's `code`; `adid` must match the creative's `id`.
- Native creative assets must be passed via the `seatbid.bid.adm.native` object. Be sure to include image assets, data assets, impression trackers, and click trackers. These assets will serve instead of the creative asset you initially registered.

The creative assets you bid with must belong to the same brand as the creative asset you initially registered. AppNexus periodically scans and reviews ad markup creative content to ensure this content is consistent with the registered creative. If there is a discrepancy, your creative can be rejected by platform audit. Frequent creative audit rejections due to rotating brands can result in revoked access to the Ad Markup Bidding with Native feature.

- If you do not pass native creative assets in your bid response, or the assets are malformed, the registered creative content will serve by default.

## Bid Response Supported Fields

- The following fields are ignored in the bid response: `adomain` and `ver`.
- Brand is set during platform audit of the registered creative, so `adomain` is not needed.

## Bid Object

Field	Type	Description
<code>adm</code>	string	Conveys ad markup in case the bid wins. The <code>native</code> object (described below) should be passed in as a JSON-encoded string.

## Native Object

Field	Type	Description
<code>assets</code>	array of objects	(Required) List of the native ad's assets. See <a href="#">Asset Object</a> below.
<code>link</code>	object	(Required) The default destination link for the native ad. Each individual asset can have its own link object, which applies if that asset is clicked. If an individual asset link does not have a link object, the parent link object is used. See <a href="#">Link Object</a> below.
<code>imptrackers</code>	array of strings	<p>Array of impression-tracking URLs expected to return a 1x1 image or HTTP 204 (No Content) response. This is typically passed only when using third-party trackers.</p> <p>The following OpenRTB macros are supported in this field:</p> <ul style="list-style-type: none"> <li>• <code>#{AUCTION_ID}</code> - AppNexus <code>auction_id_64</code></li> <li>• <code>#{AUCTION_BID_ID}</code> - ID of the bid specified in the <code>bidid</code> field in the bid response</li> <li>• <code>#{AUCTION_IMP_ID}</code> - ID of the impression, from the <code>impid</code> field in the <code>bid</code> object of the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_SEAT_ID}</code> - ID of the winning seat, from the <code>seat</code> field in the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_AD_ID}</code> - ID of the buyer's creative, from the <code>adid</code> field in the <code>bid</code> object of the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_PRICE}</code> - Clearing price of the impression in the currency specified in the <code>cur</code> field in the bid response</li> <li>• <code>#{AUCTION_CURRENCY}</code> - Currency of the clearing price, as specified in the <code>cur</code> field in the bid response</li> </ul>
<code>jstracker</code>	string	<p>Optional JavaScript impression tracker. This should be wrapped in <code>&lt;script&gt;</code> tags.</p> <p>The following OpenRTB macros are supported in this field:</p> <ul style="list-style-type: none"> <li>• <code>#{AUCTION_ID}</code> - AppNexus <code>auction_id_64</code></li> <li>• <code>#{AUCTION_BID_ID}</code> - ID of the bid specified in the <code>bidid</code> field in the bid response</li> <li>• <code>#{AUCTION_IMP_ID}</code> - ID of the impression, from the <code>impid</code> field in the <code>bid</code> object of the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_SEAT_ID}</code> - ID of the winning seat, from the <code>seat</code> field in the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_AD_ID}</code> - ID of the buyer's creative, from the <code>adid</code> field in the <code>bid</code> object of the <code>seatbid</code> object</li> <li>• <code>#{AUCTION_CURRENCY}</code> - Currency of the clearing price, as specified in the <code>cur</code> field in the bid response</li> </ul>
<code>privacy</code>	string	If support was indicated in the request, URL of a page informing the user about the buyer's targeting activity.
<code>ext</code>	object	Used for identifying AppNexus-specific extensions to the OpenRTB bid response.

## Native Ext Object

AppNexus supports a single object in the native ext object to support AppNexus-specific extensions:

Field	Type	Description
appnexus	object	Specifies the AppNexus-specific extensions to the OpenRTB bid response.

## Native Ext AppNexus Object

AppNexus supports the following fields in the `appnexus` extension object:

Field	Type	Description
third_party_imptrackers	array of strings	Array of impression-tracking URLs expected to return a 1x1 image or HTTP 204 (No Content) response. The following OpenRTB macros will be expanded as empty strings: <ul style="list-style-type: none"><li>• <code>\${AUCTION_PRICE}</code> - Clearing price of the impression in the currency specified in the <code>cur</code> field in the bid response</li><li>• <code>\${AUCTION_CURRENCY}</code> - Currency of the clearing price, as specified in the <code>cur</code> field in the bid response</li></ul> All other OpenRTB macros will expand normally.

## Asset Object

AppNexus supports the following fields to define one or more native `asset` objects to be included as a JSON-encoded string as part of the `native` object in the `adm` field of the `bid` object.

Field	Type	Description
id	integer	(Required) The unique asset ID. Must match an asset ID in the request.
required	integer	Set to 1 if bidder requires asset to be displayed.
title	object	The title object, for title assets. See <a href="#">Title Object</a> below.
img	object	The image object, for image assets. See <a href="#">Image Object</a> below.
data	object	The data object, for data assets such as ratings, prices, and so on. See <a href="#">Data Object</a> below.
link	object	This object is not supported in our native implementation.

## Title Object

Used to define a title asset in a native object.

Field	Type	Description
text	string	(Required) The text for a title element.

## Image Object

Used to define an image asset in a native object. Used for all image elements of the native ad, such as icons, main image, and so on.

Field	Type	Description
url	string	(Required) The URL of the image asset.
w	integer	(Recommended) The width of the image, in pixels.
h	integer	(Recommended) The height of the image, in pixels.

ext	object	Used for identifying AppNexus-specific extensions to the OpenRTB bid response.
-----	--------	--

## Image Ext Object

AppNexus supports a single object in the native ext object to support AppNexus-specific extensions:

Field	Type	Description
appnexus	object	Specifies the AppNexus-specific extensions to the OpenRTB bid response.

## Image Ext AppNexus Object

AppNexus supports the following fields in the `appnexus` extension object:

Field	Type	Description
prevent_crop	boolean	Allows the buyer to indicate whether the image can be cropped or not. This can be applied to icon and main image. <ul style="list-style-type: none"> <li>If flag is set to 1, the image can not be cropped (fill)</li> <li>If flag is set to 0, the image can be cropped (fit)</li> <li>If flag is not passed in/default behavior: 0; images are assumed to allow modifications unless explicitly indicated otherwise</li> </ul>

## Data Object

Used to define a data asset in a native object. Used for all miscellaneous elements in a native ad, such as ratings, price, review count, downloads, and so on.

Field	Type	Description
label	string	An optional formatting string name of the data type.
value	string	The formatted string of data to be displayed (such as "5 stars" or "\$10").

## Link Object

Used to define the link for a native asset. When clicked, the user is taken to the location of the link. Can only be defined on the parent native object.

Field	Type	Description
url	string	(Required) The landing URL for the clickable link. <b>Macros cannot be used in this field.</b>
clicktrackers	Array of strings	Array of third-party tracking URLs to be fired when the link is clicked.
fallback	string	A fallback URL to be used if the URL is not supported by the device.

## Bid Response Example

```

{
  "seatbid": [
    {
      "bid": [
        {
          "nurl":
"http://rtb-fakeurl.com/lax/wintrk=CwE&wp=${AUCTION_PRICE}&curr=${AUCTION_CURRENCY}&aid=${AUCTION_AD_ID}",
          "adid": "12345",
          "crid": "test_code",
          "price": 2.50,
          "adm":
"{\"assets\": [{\"id\":1,\"img\":{\"url\":\"https://url.com/dynamic_main_img.jpg\"},\"w\":1200,\"h\":628}],\"id\":2,\"title\":{\"text\":\"Dynamic Title\"}}, {\"id\":3,\"data\":{\"value\":\"Dynamic Description\"}}, {\"id\":4,\"data\":{\"value\":\"Dynamic Sponsor\"}}],\"link\":{\"url\":\"https://www.landingpage.com\"},\"clicktrackers\":[\"https://thirdparty-clicktracker-1.com\",\"https://thirdparty-clicktracker-2.com\"],\"fallback\":[\"https://fallback-url.com\"],\"imptrackers\":[\"https://imptracker.com\"]}",
          "impid": "3226285750417000001",
          "id": "6ab34155-c960-1111-abcd-52b7321adbbb"
        }
      ],
      "seat": "123"
    }
  ],
  "id": "3",
  "cur": "USD"
}

```

## FAQs

### Why do I have to register a creative for each brand I represent?

AppNexus policy prohibits brand rotation on creatives. By registering a creative for each brand you work with, your creative will be able to pass AppNexus audit. This will maximize the native inventory on which it can serve. AppNexus ensures that each creative complies with our audit policies by performing an initial audit and then periodically scanning the creative content that your bidder dynamically passes in its bid responses. If the dynamic content served by your bidder differs substantially from the registered creative (i.e. images and text for a different brand) it will be reaudited and may be rejected.

### Will I be charged creative audit fees for periodically reaudited creative ad markup?

No. Creative audit fees will apply *only* during your creative's initial audit.

### What happens if my Native Creative passes initial audit but fails a subsequent reaudit?

Your creative will not be permitted to serve. The audit failure may be due to rotating brands. If you believe your creative has been failed incorrectly or have other questions please contact customer support and select the Category "Creative Audit". Please note that frequent creative audit rejections due to rotating brands may result in revoked access to the Ad Markup Bidding with Native feature.

### Where do I go for more help?

If you have additional questions, please contact your account representative or [customer support](#).