ADMARU VIDEO PLAYER

Ads Player Configuration

Ad
maru

Player support Values/Parameters.

Support Video format – HTML 5 Videos (.mp4,.mov,WebM,.ogv),youtube,dailymotion,vimeo-oembed iframe

Support Video Stream – hls,mpeg dash,webrtc, rtmp with rtmp java script plugin,p2p stream

- **Responsive - works with any screen size**
- **Monetization - make money from your videos with any ads network –prebid,header bidding Up to VPAID 2 & Vast 4.2**
- **Streaming - support for hls.js, Shaka and dash.js streaming playback**
- **API - toggle playback, volume, seeking, and more through a standardized API**
- **Events - no messing around with Vimeo and YouTube APIs, all events are standardized across formats**

IMA Support Parameters:

Settings	Type	Description
adLabel	string	Replaces the "Advertisement" text in the ad label. Added for multilingual UI support.
adLabelNofN	string	Replaces the "of" text in the ad label (e.g. ... (1 of 2) ...). Added for multilingual UI support.
adTagUrl	string	A URL which returns a VAST, VMAP or ad rules response. This will override adsResponse.
adsRenderingSettings	object	JSON object with ads rendering settings as defined in the IMA SDK Docs(1).
adsResponse	string	The VAST, VMAP, or ad rules response to use in lieu of fetching one an ad tag. This is overridden if adTagUrl is set.

Settings	Type	Description
adsRequest	object	JSON object with ads request properties defined in the IMA SDK Docs(2). Properties set here that can also be provided elsewhere (e.g. adTagUrl) will override those other settings.
autoPlayAdBreaks	boolean	Whether or not to automatically play VMAP or ad rules ad breaks. Defaults to true.
deprecated adWillPlayMuted	boolean	Notifies the SDK whether the player intends to start ad while muted. Changing this setting will have no impact on ad playback. Defaults to false.
contribAdsSettings	object	Additional settings to be passed to the contrib-ads plugin(3) used by this IMA plugin.
debug	boolean	True to load the debug version of the plugin, false to load the non-debug version. Defaults to false.
disableAdControls	boolean	True to hide the ad controls(play/pause, volume, and fullscreen buttons) during ad playback. Defaults to false.
disableCustomPlaybackForIOS10Plus	boolean	Sets whether to disable custom playback on iOS 10+ browsers. If true, ads will play inline if the content video is inline. Defaults to false.
disableFlashAds	boolean	True to disable Flash ads - Flash ads will be considered an unsupported ad type. Defaults to false.
featureFlags	object	Sets IMA SDK feature flags.
forceNonLinearFullSlot	boolean	True to force non-linear AdSense ads to render as linear fullslot. If set, the content video will be paused and the non-linear text or image ad will be rendered as fullslot. The content video will resume once the ad has been skipped or closed.
id	string	**DEPRECATED** as of v.1.5.0, no longer used or required.
locale	string	Locale for ad localization. The supported locale codes can be found in Localizing for Language and Locale
nonLinearHeight	number	Desired height for non-linear ads. Defaults to 1/3 player height.

Settings	Type	Description
nonLinearWidth	number	Desired width of non-linear ads. Defaults to player width.
numRedirects	number	Maximum number of VAST redirects before the subsequent redirects will be denied and the ad load aborted. The number of redirects directly affects latency and thus user experience. This applies to all VAST wrapper ads.
omidVendorAccess	object	Sets and enables the Open Measurement SDK(4). Accepts an object with keys corresponding to OMID verification vendors(5). The value pair for each key should be the OMID access mode(6) associated with that vendor.
ppid	string	Sets the publisher provided ID
preventLateAdStart	boolean	Prevent ads from starting after the content has started if an adtimeout occurred (preroll, midroll, postroll). The default value is false
sessionId	string	Sets the session ID
showControlsForJSAds	boolean	Whether or not to show the control bar for VPAID JavaScript ads. Defaults to true.
showCountdown	boolean	Whether or not to show the ad countdown timer. Defaults to true.
vastLoadTimeout	number	Override for default VAST load timeout in milliseconds for a single wrapper. The default timeout is 5000ms.
vpaidAllowed	boolean	**DEPRECATED**, please use vpaidMode.
vpaidMode	VpaidMode(5)	VPAID Mode. Defaults to ENABLED. This setting,overrides vpaidAllowed.

(1) AdsRenderingSettings
(2) AdsRequest
(3) contrib-ads plugin
(4) Open Measurement SDK guide
(5) OmidVerificationVendor

(6) OmidAccessMode
(7) ImaSdkSettings.setVpaidMode

IMA Plugin Ad Events

The IMA Plugin fires events that can be listened for. Ad lifecycle events can be listened for by following our Advanced Example. Other events are emited from the videojs player. Please see the below example to set up listeners for these events.

```
this.player = videojs('content_video');

this.player.on('ads-manager', function(response){
  var adsManager = response.adsManager;
  // Your code in response to the `ads-manager` event.
})
```

Below are the events added by the videojs-ima plugin to the videojs player.

Event	Event String	Payload
Ad Started	'ads-ad-started'	none
Ads Manager	'ads-manager'	google.ima.AdsManager
Ads Loader	'ads-loader'	google.ima.AdsLoader
Ads Request	'ads-request'	google.ima.AdsRequest

Disable automatic ad break playback

In some circumstances you may want to prevent the SDK from playing ad breaks until you're ready for them. In this scenario, you can disable automatic playback of ad breaks in favor of letting the SDK know when you're ready for an ad break to play. To do so:

1. Set `autoPlayAdBreaks` to false in the initial options.
2. Provide an ad break ready listener via `setAdBreakReadyListener`.
3. Call `player.ima.playAdBreak()` in your ad break ready listener when you're ready to play the ads.

Logo Configuration

Property	Attributes	Type	Default value	Description
image	**Required**	**String**		**The URL to the logo image.**
url	Optional	String		A url to be linked to from the logo. If the user clicks the logo the link will open in a new window.
position	Optional	String	"top-right"	The location to place the logo (top-left, top-right, bottom-left, or bottom-right).
offsetH	Optional	Number	0	Horizontal offset (px) from the edge of the video.
offsetV	Optional	Number	0	Vertical offset (px) from the edge of the video.
width	Optional	Number		The width of the logo image (px). If not specified, it will be the width of the original image.
height	Optional	Number		The height of the logo image (px). If not specified, it will be the height of the original image.
padding	Optional	Number	5	Padding around the logo image (px).
fadeDelay	Optional	Number, Null	5000	Time until fade-out begins (msec). If `null` is specified, automatic fade-out is not performed.
hideOnReady	Optional	Boolean	false	Do not show the logo image when the player is ready.
opacity	Optional	Boolean	1	The opacity of the logo (from `[0, 1]`). If not specified, it will default to 1.

Methods

You can also manually show / hide the logo image at any time.

```
// To show the logo image on the player's play event:
player.on('play', () => {
  player.logo().show();
});
```

Method	Description
show()	Show the logo image
hide()	Hide the logo image

Prebid Out Stream Render Sample Code:

Demo 1:

Place this code in the page header.

```
<script>
    var pbjs = pbjs || {};
    pbjs.que = pbjs.que || [];

    function callANRenderer(bid, ad) {
      const adResponse = {
          ad: {
              video: {
                  content: ad,
                  player_width: 640,
                  player_height: 480,
              }
          }
      }

      bid.renderer.push(() => {
          window.ANOutstreamVideo.renderAd({
              targetId: bid.adUnitCode,
              adResponse,
          });
      });
    }

    function render(bid) {
        let ad = bid.ad || bid.vastXml;

        if (ad) {
            callANRenderer(bid, ad)
        } else {
            if (bid.vastUrl) {
              (async () => {
                  ad = await fetch(resp).then(resp => resp.text());

                  if (typeof ad === 'string') {
                      callANRenderer(bid, ad);
                  } else {
                      console.log('Invalid VAST');
                  }
              })();
            } else {
                console.log('Invalid ad');
            }
        }
    }

    const adUnits = [{
        code: 'admaru',
        mediaTypes: {
            video: {
                context: 'outstream',
                playerSize: [640, 480],
                mimes: ['video/mp4'],
                protocols: [1, 2, 3, 4, 5, 6, 7, 8],
                playbackmethod: [2],
                skip: 1,
                renderer: {
                    render,
                    url: " outstreamrender.js"
                },
            }
        },
        bids: [{
            bidder: 'appnexus',
            params: {
                placementId: 13232385
            }
        }]
    }];

    pbjs.que.push(function() {
        pbjs.addAdUnits(adUnits);
        pbjs.requestBids({
            timeout: 1000,
            bidsBackHandler: function(bids) {
                const highestCpmBids = pbjs.getHighestCpmBids('admaru');
                pbjs.renderAd(document, highestCpmBids[0].adId);
            }
        });
    });
</script>
```

Place this code in the page body.

```
<div id='admaru'>
  <p>Test Outstream Video Ad Demo</p>
</div>
```

```
playback_method:


    'unknown': 0,
    'auto_play_sound_on': 1,
    'auto_play_sound_off': 2,
    'click_to_play': 3,
    'mouse_over': 4,
    'auto_play_sound_unknown': 5
```

context:

```
    'unknown': 0,
    'pre_roll': 1,
    'mid_roll': 2,
    'post_roll': 3,
    'outstream': 4,
    'in-banner': 5
```

Demo 2:

```
<script>
  var pbjs = pbjs || {};
  pbjs.que = pbjs.que || [];

  function render(bid) {
    const videoContent = document.getElementById('contentElement');
    const adContainer = document.getElementById('adContainer');
    const adDisplayContainer = new google.ima.AdDisplayContainer(adContainer,
videoContent);

    const adsRequest = new google.ima.AdsRequest();
    adsRequest.adTagUrl = '';
    adsRequest.adsResponse = bid.vastXml;

    adsLoader = new google.ima.AdsLoader(adDisplayContainer);
    adsLoader.requestAds(adsRequest);


adsLoader.addEventListener(google.ima.AdsManagerLoadedEvent.Type.ADS_MANAGER_LOADED,
(adsManagerLoadedEvent) => {
      const adsRenderingSettings = new google.ima.AdsRenderingSettings();
      adsRenderingSettings.restoreCustomPlaybackStateOnAdBreakComplete = true;
      /* videoContent should be set to the video DOM element. */
      const adsManager = adsManagerLoadedEvent.getAdsManager(videoContent,
adsRenderingSettings);

      /* Ad event listeners */
      adsManager.addEventListener(google.ima.AdEvent.Type.LOADED, () =>
videoContent.play());
      adsManager.addEventListener(google.ima.AdEvent.Type.CONTENT_PAUSE_REQUESTED, ()
=> videoContent.pause());
      adsManager.addEventListener(google.ima.AdEvent.Type.CONTENT_RESUME_REQUESTED,
() => videoContent.play());

      /* Play the ad */
      videoContent.load();
      adDisplayContainer.initialize();
      adsManager.init(640, 360, google.ima.ViewMode.NORMAL);
      adsManager.start();
    }, false);
  }

  /* Prebid video ad unit */
  const videoAdUnit = {
    code: 'adContainer',
    renderer: {
      render,
      url: 'https://imasdk.googleapis.com/js/sdkloader/ima3.js'
    },
    mediaTypes: {
      video: {
        context: 'outstream',
        playerSize: [640, 480],
        mimes: ['video/mp4'],
        protocols: [1, 2, 3, 4, 5, 6, 7, 8],
        playbackmethod: [2],
        skip: 1,
      }
    },
    bids: [{
      bidder: 'appnexus',
      params: {
        placementId: 13232385
      }
    }]
  };

  pbjs.que.push(() => {
    pbjs.addAdUnits(videoAdUnit);

    pbjs.requestBids({
      bidsBackHandler: (bids) => {
        const highestCpmBids = pbjs.getHighestCpmBids('adContainer');
        pbjs.renderAd(document, highestCpmBids[0].adId);
      }
    });
  });
</script>
```

Option	**Values**	**Default**	**Description**
Video Player Setup			
audioOnMouseover (deprecated)	• true • false • integer	true	This setting is no longer supported or tested because of current browser limitations on sounds. Play audio when the user's mouse is over the video, pause/unmute audio when the mouse is not over the video. If an integer is specified, the audio will not start until that many milliseconds of continuous mouse hovering over the video.
autoInitialSize	true / false	false	If true, the player width will match the page container. If false, the player width set on the placement will be used.
alignment	center / left / right	center	Alignment of the player if its size is smaller than the container.
playOnMouseover	true / false	false	When this setting is true, while the user's mouse is over the video it will play. If the mouse leaves the video it will pause and resume playing when the mouse returns over the video. When this setting is false, the video plays regardless of the user's mouse.
playVideoVisibleThreshold	0-100	50	The video will not start playing until the video container's visibility is above this threshold. If the video container's visibility drops below this threshold the video follows the behavior specified in `nonViewableBehavior`.
nonViewableBehavior	• "pause" • "mute"	"mute"	• "pause"- Video will pause when below `playVideoViewableThreshold`, and resume playback when above `playVideoViewableThreshold` • "mute" - Video will mute but not pause when below `playVideoViewableThreshold` and unmute when

Option	Values	Default	Description
			above`playVideoViewableThreshold` ○ When scrolled back above the viewable threshold, the video should not unmute if it has been muted or set to 0 volume explicitly by the user, we should simply restore the pervious volume state that existed when the user scrolled out of viewability.
expandTime	integer (milliseconds)	1000	The amount of time in milliseconds it takes to expand the ad and to collapse the ad.
playerTechnology	array of strings	["html5", "flash"]	VPAID media files with corresponding technology are given priority for display. Otherwise, other media files with their corresponding technology are displayed.
aspectRatio	• "auto" • "16:9" • "4:3"	If not defined, and fixedSizePlayer is false, 16:9	The aspect ratio of the video portion of the player. Does not take into account any control bars which do not overlay the video. "auto" uses the aspect ratio defined in the video metadata.

maxHeight	true / false / height in pixels	false	Optionally, enable and set a maximum height for the player as shown in the following example. The player will adjust to maintain the aspect ratio.
			`"maxHeight": {`
			`        "enabled": true,`
			`        "height": 480`
			`}`
			This setting was designed to optimize vertical video presentations and should only be used with a vertical aspect ratio. For more information, including recommended values, see Vertical Video Formats.

fixedSizePlayer	true / false	false	If true, final rendered player height will be the height of the entire player, including any top or bottom control bars.

Option	Values	Default	Description
			If false, final rendered player height will be the height of the video area, and any extra control bars (top bar, bottom control bar) will increase the total height of the player.

Player's Chrome Settings

Option	Values	Default	Description
adText	string	Ad	Ad indicator text shown in the top-right hand corner of the outstream ad unit. If `skipLocation` is also top-right, the `skipText`, and subsequently `skipButtonText`, will be shown to the left of adText in the top-right corner (see skippable object below).
disableCollapse	true/false	false	If true, the outstream ad will not collapse at the end of the video. The last frame that was showing at the end of the video will show. No control bar, click to play button or countdown text will show. Ad Indicator and Skip button will remain.
replay	true / false	false	Enables a replay button on the card that plays the ad a second time when clicked.
controlBarPosition	string	over (except on Android)	Determines whether the position of the bottom control bar floats over the playback area or is fixed. Specify `over` to have the bottom control bar float over the playback area when the user interacts with the video Specify `below` to have the bottom control bar remain visible in a fixed position below the playback area during the duration of the ad. On Android, the control bar is always displayed in this position.
showMute	true / false	true	Show or hide the mute button in the control bar. If `audioOnMouseOver` is set to `false`, a user activates sound by disabling the audio mute button. If `showMute` is also `false` and the audio mute button is hidden, a user activates sound by dragging the volume slider.
showVolume	true / false	true	Show or hide the volume slider on the control bar.
showProgressBar	true / false	false	Show or hide the progress bar on the control bar.

Option	Values	Default	Description
allowFullscreen	true / false	true	Show or hide the fullscreen button.
disableTopBar	true / false	false	If set to `true`, the top player chrome bar (with adText and skipText/skipButtonText) will not display. If set to `false` the top player chrome bar will display except in the following case - If empty text fields are specified, a chrome bar will still be displayed, but will be empty.

Disable Collapse Settings

These settings enable the user to control what actions will occur when the video playback has completed.

enabled	boolean or integer	true	If true, the outstream ad will not collapse at the end of the video and the last frame that was showing at the end of the video will be displayed. No control bar, click to play button or countdown text will display, and the Ad Indicator and Skip button will remain. If false, the replay and hideControlBar values won't matter because the video will collapse on completion. If an integer is passed, the player sets a delay in milliseconds equal to the passed integer before collapsing the ad.
`replay`	boolean	false	If true, a replay button displays once the video is completed, enabling the user to replay the video.
hideControlBar	boolean	false	If true, and the replay setting is false, the bottom control bar is hidden after video completion.

VPAID Handler Settings

These settings determine how the video player handles video documents that use VPAID.
Note: When a VPAID ad reaches its final frame, the Outstream player hides the control bar, and does not display a replay button. This behavior prevents any potential conflicts with interactive end screen elements supplied by VPAID. The player handles all VPAID ads this way, regardless of the configuration specified in Monetize or in page-level Outstream settings.

Option	Values	Default	Description
vpaidTimeout	integer (milliseconds)	5000	If a VPAID ad unit is showing, time to wait between a VPAID ad unit starting and when it actually starts before timing-out and collapsing the ad. We might encounter this if a VPAID ad

Option	Values	Default	Description
			unit attempts to run a secondary auction which is especially slow, or hits misconfigured servers. We will also likely encounter this in the case of a non-fully-compliant VPAID ad unit
terminateUnresponsiveVPAIDCreative	boolean	false	If a VPAID creative does not respond properly to a pause event, terminate it
delayExpandUntilVPAIDInit	true / false	true	If true, wait to expand video area for AdLoaded event to be returned and the correct position on the page before Outstream can be expanded in a VPAID case. If false, the outstream will expand as soon as it has a creative and is at the correct position on the page.
delayExpandUntilVPAIDImpression	true / false	false	Delay Outstream expansion for VPAID ads until `AdImpression` or `AdVideoStart` event has been received. Note that since we are starting the ad and delaying its visibility until Outstream expands, this might lead to a higher incidence of discrepancies in reporting.

Mobile Setting

This setting enables the player on mobile devices.

Option	Values	Default	Description
enableInlineVideoForIos	boolean	true	Enable inline video for iOS devices.

Not Shown in UI Settings

The following settings are for specialized configurations that aren't available through Monetize.

Option	Values	Default	Description
parentIframeGeometryEvent	string	n/a	For non-friendly iFrame implementation cases only.

Option	Values	Default	Description
parentIframeGeometryCb	function callback	n/a	For non-friendly iFrame implementation cases only.
vpaidEnvironmentVars	JSON Object	n/a	Sends JSON object containing vpaidEnvironmentVars to VPAID creative as "envinronmentVars" parameter of VPAID initAd() call.
cbNotification	function callback	n/a	Callback function to be called on specific ad unit. Function foodprint is: adCallback(eventType, eventName, adId, eventData) where eventType is one of VPAID, VAST, AdUnit and eventName will be in one of the following eventType categories: **VPAID**: AdLoaded, AdStarted, AdStopped, AdSkipped, AdSkippableStateChange, AdSizeChange, AdLinearChange, AdDurationChange, AdExpandedChange, AdRemainingTimeChange, AdVolumeChange, AdImpression, AdVideoStart, AdVideoFirstQuartile, AdVideoMidpoint, AdVideoThirdQuartile, AdVideoComplete, onAdVClickThru, AdInteraction, AdUserAcceptInvitation, AdUserMinimize, AdUserClose, AdPaused, AdPlaying, AdLog, AdError **VAST**: creativeView, start, midpoint, firstQuartile, thirdQuartile, complete, mute, unmute, pause, rewind, resume, fullscreen, expand, collapse, close, exitFullscreen, skip, progress, acceptInvitation, acceptInvitationLinear, closeLinear, impression, error **AdUnit**: collapseStart, collapseEnd, expandStart, expandEnd, adComplete, renderedPlayerSize, waterfall (see eventData below) adId will be the id for the ad which is defined by AST (target element ID) and passed to the renderer upon renderAd() call

Option	Values	Default	Description
			eventData = additional data passed in with the event For waterfall event, this will be an Object containing the following fields: status (stepStart, stepFail, stepSuccess, timeout), step, totalAvailableSteps, stepLimit, maxTime, timeRemaining, failureReason (if status = stepFail)

Waterfall Handling Settings with Mediation

The following settings apply only when Video Mediation is configured.

With mediation, waterfall handling of video ads increases your likelihood of serving a viable creative within the timeout period. With the waterfall process, you can attempt to load a series of ads, stopping when a suitable ad is found. You can set the timeout period as well as the maximum number of mediation ad sources that will be checked when trying to find a suitable ad. Once you have enabled mediation on the placement via Monetize, you can set waterfall options to further fine-tune your timeout preferences.

Option	Values	Default	Description
waterfallTimeout	integer	15000	Maximum amount of time in milliseconds that waterfall processing of mediated ad sources is allowed to run before it stops attempting to deliver a viable ad.
waterfallSteps	• -1 • 0 • positive integer	-1	Maximum number of waterfall attempts that are allowed to try and display the ad. Mediation processing produces a list of ad sources to try to get a viable ad to render. This list may include mediated bids as well as an RTB ad (the ad that won the auction). The publisher (or renderer) can control the number of waterfall mediation ad sources to try in an attempt to show an ad. The possible values for this option are interpreted as follows: • -1 = all the ad sources in the waterfall list will be processed until a viable ad is found or the waterfallTimeout has expired • 0 = only one ad sources will be used to produce an ad - use the first ad on the list

Option	Values	Default	Description
			• positive integer = the maximum number of non-rtb ad sources to try until a viable ad is found o If the number of ad sources is less than the value of this property then all of the ad sources will be attempted o If none of the non-rtb ad sources yields a viable ad source and the waterfall timeout has not expired, then the rtb ad should be attempted. o If you anticipate that the number of attempts to render an ad will exceed 6, then it is recommended that you do set this value to some number greater than 6.

Player Skinning Settings

The following settings are nested under the `playerSkin` node, and determine the look and feel of the outstream player.

Option	Values	Default	Description
topBarColor	string	"black"	Background color for the top chrome that contains ad indicator and skipping text *the value of all player skin colors can be any css-ready string, including "black", "#FFFFFF", "rgb(255,255,255)" and "rgba(255,255,255,1)"
topBarTextColor	string	"#CCCCCC"	Text color for the top chrome that contains ad indicator and skipping text.
dividerColor	string	"#555555"	Color of the divider between the top chrome and the video.
dividerHeight	positive integer	1	Height of the divider between the top chrome and the video.
controlBarColor	string	"#CCCCCC"	Background color for the control bar that contains play button, elapsed time, and volume controls.
bigPlayButtonColor	string	"#3B4249"	Color of the play button that overlays the video when paused.
controlBarControlsColor	string	"#CCCCCC"	Control of the play button, elapsed time text, and volume controls in the control bar.

Option	Values	Default	Description
controlBarDividerColor	string	"#555555"	Color of the divider between the control bar and the video on iOS. Has no effect on desktop browser or android html5 player.

Sidestream Settings

The following settings are configured using the `sidestream` node, and determine whether and how videos are played in sidestream mode.

Option	Values	Default	Description
enabled	true / false	false	If `true`, sidestream behavior is enabled.
startInSideStream	true / false	false	When sidestream is enabled, setting this option to `true` specifies that the creative will begin playing in sidestream mode, instead of in the outstream player. `startInSidestream=true` is not recommended for mediated placements. Warning: Setting both `startInSidestream` and `returnToPage` to the same value can result in ad display problems and is not recommended.
returnToPage	true / false	true	When sidestream is enabled, setting this option to `true` specifies that when the user scrolls down after sidestream is initiated, the sidestream player closes and the ad pauses until the user scrolls up again, then resumes playing in the outstream player. Setting this option to "false" ensures that the video plays to completion in sidestream mode, and collapses the original outstream player. Warning: Setting both `startInSidestream` and `returnToPage` to the same value can result in ad display problems and is not recommended.
position	"top-left" "top-right" "bottom-left" "bottom-right"	"bottom-right"	Specifies corner of the publisher page which the player will move to when side-stream functionality is activated.

Option	Values	Default	Description
xOffset	integer	0	Specifies the distance from the page edge on the x-axis, in pixels which the player will be shown when side-stream functionality is activated.
yOffset	integer	0	Specifies the distance from the page edge on the y-axis, in pixels which the player will be shown when side-stream functionality is activated.
width	integer	<blank>	Specifies the width of the player in sidestream mode, if not set, the player will the the same width as it is in the non-sidestream mode.
height	integer	<blank>	Specifies the height of the player in sidestream mode, if not set, the player will the the same height as it is in the non-sidestream mode.
maxPageOverlay	integer		Specifies the maximum percentage of the browser window that sidestream will cover. If the size of the player will be larger than this percentage, the player will not go into sidestream mode, and will behave as described in the "nonViewablePlayer" option.

End Card Settings

The following settings are configured using the `endCard` node. They determine whether an end card is displayed when the player remains open after video play is complete. For an end card to be displayed, first tell the player not to collapse after play by setting `disableCollapse.enabled` to `true`. Otherwise, the player will close when the play is complete and the end card settings will be ignored.

Option	Values	Default	Description
enabled	true / false	false	If `true`, an end card is displayed at the end of video play, while the video player remains open. However, if `disableCollapse.enabled` is set to `false`, the player will close and end card settings will be ignored.
color	string	none	When an end card is enabled, draws a colored layer on top of the final frame of the video. Use a CSS color value. For example, enter `#000000` for black.
clickable	true / false	true	If set to `true`, the ad area remains clickable while the end card is shown. Clicking in the ad area opens the clickthrough URL from the ad that was just displayed, as if the ad had been clicked.
imageURL	Complete URL		The URL of the image (including https:) to be shown on the end card if no companion ad is specified or available. If you specify an image, an image layer is drawn on top of the color layer (if `color` is specified) or on top of the video frame.

Option	Values	Default	Description
			The image is centered in the ad area and drawn at the native resolution of the source image unless you specify the size of the image using the `imageWidth` and `imageHeight` options. Note: Typically, an end card image is used as the fallback, with a companion ad defined using companionID. The image is never displayed if a valid companion ad is available.
imageWidth	string	none	If `imageUrl` has been defined, the width of the rendered image. You can use an exact pixel width (e.g. "320px"), or specify a percentage-based width (e.g. "100%") so that the image scales when the ad area is resized. If imageWidth is not specified, the image is drawn at its native width.
imageHeight	string	none	If imageUrl has been defined, the height of the rendered image. You can use an exact pixel width (e.g. "200px"), or specify a percentage-based width (e.g. "100%") so that the image scales when the ad area is resized. If imageWidth is not specified, the image is drawn at its native width.
showCompanion	true/false	true	Specifies whether or not to show a companion ad (if available) on the end card. For ads that don't use companions, the image defined by imageUrl will be displayed on the end card. For ads that contain multiple companions, the end card shows the companion ad that most closely matches the aspect ratio and size of the ad unit area.

Code Sample

The following example shows a typical outstream video player setup.

```
rendererOptions: {
            "cbNotification": handlePublisherCb,
            "autoInitialSize": false,
            "alignment": "center",
            "playOnMouseover": false,
            "playVideoVisibleThreshold": 50,
            "expandTime": 1000,
            "adText": "Ad",
            "showMute": true,
            "showVolume": true,
            "showPlayToggle": true,
            "showBigPlayButton": true,
            "showProgressBar": false,
            "allowFullscreen": true,
            "vpaidTimeout": 5000,
            "terminateUnresponsiveVPAIDCreative": false,
            "vpaidEnvironmentVars": {},
            "enableInlineVideoForIos": true,
            "preloadInlineAudioForIos": false,
            "disableCollapse": {
                  "enabled": false,
                  "replay": false
              },
            "aspectRatio": "16:9",
            "nonViewableBehavior": "mute",
            "delayExpandUntilVPAIDInit": true,
            "delayExpandUntilVPAIDImpression": false,
            "delayStartUntilNotified": false,
            "controlBarPosition": "over",
            "disableTopBar": false,
            "waterfallTimeout": 15000,
            "waterfallSteps": -1,
            "skippable": {
                "enabled": true,
                "videoThreshold": 15,
                "videoOffset": 5,
                "skipLocation": "top-left",
                "skipText": "Video can be skipped in %%TIME%% seconds",
                "skipButtonText": "SKIP",
                "allowOverride": false
            },
            "playerSkin": {
                "topBarColor": "#5A1011",
                "topBarTextColor": "yellow",
                "dividerColor": "lightblue",
                "dividerHeight": 1,
                "controlBarColor": "blue",
                "bigPlayButtonColor": "red",
                "controlBarControlsColor": "yellow",
                    "controlBarDividerColor": "red",
                    "videoBackgroundColor": ""
                },
            "fixedSizePlayer": false,
            "sideStream": {
                "enabled": true,
                "startInSideStream": false,
                "returnToPage": true,
                "position": "top-right",
                    "xOffset": 0,
                    "yOffset": 0,
                    "maxPageOverlay": 50,
                    "width": 300,
                    "height": 250,
                },
            "topMostViewableCheck": {
                "enabled": true,
                "checkVerticalOnly": true,
                "rateOfBeacons": 10
                },
            "intervals": {
                "playAndPause": 250,
                "focusAndBlur": 500
                },
            "learnMore": {
                "enabled": false,
                "separator": "-",
                "text": "Learn More",
                "clickToPause": true
                },
            "endCard": {
                "enabled": true,
                "clickable": true,
                "color": "",
                "imageUrl": "",
                "imageWidth": "",
                "imageHeight": "",
                "showCompanion": true
               }
             }
```

Placement-Level Outstream Options

The options below can be set on the placement in Monetize or via the API.

Tip:

A best practice for setting up your inventory is to have a one-to-one mapping between an ad unit position on your website and a placement in Xandr. In other words, try to have only one placement per ad unit position. With one-to-one mapping, you can control your outstream ad unit's appearance and behavior from the placement settings and provide any necessary additional configuration from the page.

If you choose to map multiple ad unit positions to a single Run of Site placement, you control your outstream ad unit's appearance and behavior from the on-page settings. This decision allows you to use different settings for each ad unit position on your website.

Monetize Field	API Field	Values	Default	Definition
Skippable	supports_skippable	true / false	false	If true, skippability is enabled.
Player Width	width	integer	0	The default width of the player. This field is mandatory. If the width is zero, the autoInitialSize option will not automatically set the

Monetize Field	API Field	Values	Default	Definition
				width of the player on the page.
Playback Method	playback_method	auto_play_sound_on / auto_play_sound_off / click_to_play / mouseover / auto_play_sound_unknown		