

Getting started with Google Ad Manager's Dynamic Ad Insertion

Best practices for delivering a seamless ad experience across live, linear, and on-demand video content

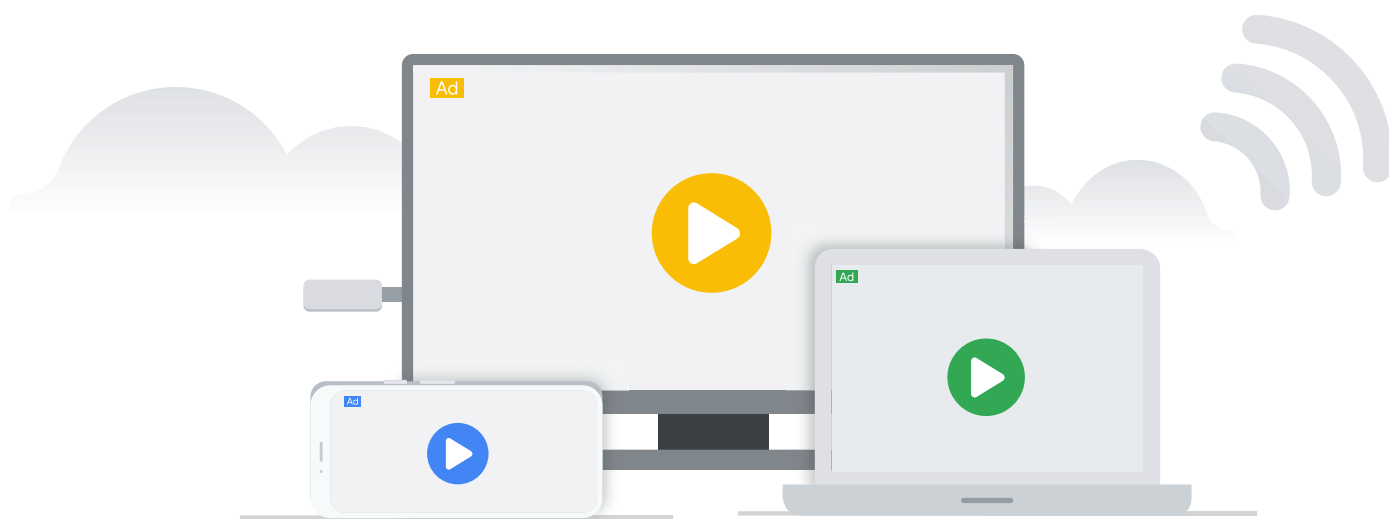


Table of Contents

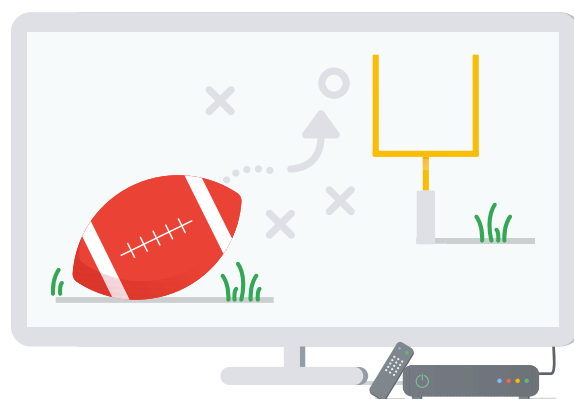
Executive Summary	3
Part 1: Preparing to Implement DAI	6
Fundamental technologies and prerequisites	6
Implementing adaptive streaming	8
HTTP live streaming (HLS)	8
HLS video on-demand	11
HLS live/linear	11
Dynamic adaptive streaming over HTTP (DASH)	12
DASH video on-demand	13
DASH live/linear	14
Part 2: Implementing DAI	16
Generating ad tags in DAI	16
Best practices and recommendations for ad tags	16
Platform support	17
Optimal player development	18
Part 3: Best practices for managing live events with DAI	19
1. Plan for live events	20
2. Build for failure	22
3. Leverage data and standards	24
Part 4: Troubleshooting common challenges with DAI	27
Locate a DAI session ID or debug key	27
Stream activity monitor (SAM)	28
Live stream monitor (LSM)	29
Get started with DAI	32

Executive Summary

Television continues to offer the ultimate viewing experience—the big screen, enhanced sound, and high-quality picture that people have come to love and expect. But the way we watch TV has changed. Today, [80% of US households](#) use a connected TV device, and subscribers have an [average of four](#) paid video streaming services. This new over-the-top (OTT) technology lets people stream TV content anytime and anywhere, whether that's via connected TV on the big screen in the living room, on a mobile phone or tablet when on the go, or from their laptop in another room.

The growth of OTT has also created a heightened importance in streaming quality. People want the same broadcast-quality, reliable experience they've grown accustomed to with traditional TV. They don't want buffering, slow load times due to latency, or failed ads or content. According to Conviva, nearly [20% of viewers](#) abandon content when an ad has a 5-second delay. Streaming quality is important, and it's continuing to improve with faster internet speeds, new video distribution mechanisms, and reliable dynamic ad insertion technology.

Google Ad Manager is helping TV partners deliver a broadcast-quality, seamless, and personalized ad experience across every screen with [Dynamic Ad Insertion \(DAI\)](#). This server-side ad insertion (SSAI) technology makes it possible for partners to insert ads into live, linear, and on-demand TV programming, wherever viewers are watching. It minimizes buffering and latency, and it ensures ads are rendered in the appropriate format, so that viewers have the broadcast-quality experience they expect. With a direct integration with Ad Manager 360, DAI offers TV partners added scale, precision, and flexibility, so they can earn more revenue for their video inventory.



How does DAI work?

At a high level, DAI transcodes every video ad in advance to meet each device's format and delivery requirements while unifying ads and content into one real-time stream. Across live, linear, and on-demand content, DAI supports mobile apps (iOS, Android), web (HTML5), and [connected TV platforms](#).

Delivering seamless, personalized ads is especially challenging and critical during live events, where stream failure has high-stake consequences. DAI reliably supports millions of concurrent viewers with advanced prefetching ad decisioning, Google infrastructure and serving capacity, and a sophisticated stream failure detection and mitigation system.

What are the benefits of DAI?

With DAI, programmers and distributors can be confident that ads will deliver seamlessly into their video content. This has become even more important as video viewership grows. In 2019, the number of video ad impressions served using Ad Manager's DAI technology increased more than 140% year-over-year¹.

Highlights of Google Ad Manager's DAI solution include:

- **Full advanced TV monetization capabilities with Ad Manager 360 integration**

DAI is directly integrated into Google Ad Manager's ad server, which enables all advanced TV monetization capabilities to work out-of-the-box. This includes:

- **Support for Ad Manager advanced TV features, including Smarter Ad Breaks, inventory sharing, and TV forecasting.**
- **Scalable programmatic performance across all transaction types with full programmatic signals coverage and Google proprietary demand access, including TrueView and Bumpers via the Dynamic Ad Insertion Interactive Media Ads Software Development Kit (DAI IMA SDK).**
- **Integrated viewability measurement with Active View and Open Measurement.**

- **Live scale to ensure stability and maximize fill rates**

DAI can scale with the millions of viewers who are watching live TV. For example, Google Ad Manager's Dynamic Ad insertion solution has delivered relevant ads to more than 1 million concurrent livestream viewers during multiple live events occurring at the same time².

- **Integration with industry standards**

Our standards-driven approach helps publishers, advertisers, and technology providers follow similar advertising guidelines. As part of this, DAI is the first server-side ad insertion (SSAI) solution to be accredited by the Media Rating Council for its client-initiated impression measurement of video ads on live and on-demand content.

¹ Google Ad Manager Internal Data, Global, Q2'2018 - Q2'2019

² Google Ad Manager Internal Data, Global, August 2019 - August 2020

DAI also offers reliable, accurate Interactive Advertising Bureau (IAB) Video Ad Serving Template (VAST) compliant reporting and tracking with client-side verification. DAI also offers support for SCTE-35 and both HTTP Live Streaming (HLS) and Dynamic Adaptive Streaming over HTTP (DASH) standards.

- **Advanced troubleshooting capabilities**

Partners can easily troubleshoot and debug issues with robust debugging tools that include livestream monitoring and mitigation support.

- **Insightful reporting**

Ad Manager's advanced video metrics let publishers report on a number of video-specific metrics involving VAST, live stream ad breaks, positioning of ad breaks, and more. These metrics help partners analyze their streams and understand their performance.

- **Creative transcoding, deliver, and rotation protection**

Ad Manager transcodes and delivers the creatives so that the right creative format will always deliver to the right device. We also detect and protect from creative rotations to ensure that the right ad is delivered.

- **Integration with leading technology vendors**

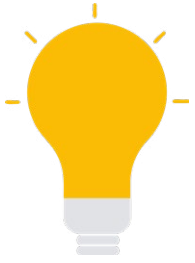
DAI is integrated with numerous encoders, video players, and content delivery networks, giving you flexibility to keep working with your existing products and services.

- **Global support**

We offer world-class solution architecture, product onboarding consulting services, live event support services, and 24/7 troubleshooting through a global team of video experts.

Interested in learning more? This guide will help you learn how to use Ad Manager's DAI solution to seamlessly optimize your ad experience for all content types, troubleshoot your DAI setup, and deliver the best ad experience across screens to grow revenue.





Part I: Preparing to implement DAI

Fundamental technologies and prerequisites

There are several steps you should take before you incorporate DAI into your video business. You might already be using some of them for your traditional video advertising, while others are unique to Ad Manager DAI and require additional implementation.

These systems should already be in place:

- **Content delivery network (CDN)**
The origin server for all your digital content.
- **Content management system (CMS)**
Maintains metadata for your digital content and provides a media-rich site summary (MRSS) feed output. A CMS is optional for live/linear but required for video on-demand.

You should also implement these technologies:

- **[For Live/Linear content only] Injector**
This technology takes the SCTE-104 information and cuts the content, using cue-out and cue-in to make room for ad breaks. This allows for dynamic ad insertion in between content, so that the content transitions into ads seamlessly, and after playback of ads it transitions back to the content. If there are no monetization opportunities, the injector continues to feed content to the encoder/transcoder.
- **Encoder/transcoder**
Although DAI will automatically encode the ads, the partner will still be responsible for encoding the content. This technology has two functions for partner content:
 - The contribution encoder takes the analog-formatted content and transforms it into digital format. Once that's complete, the distribution encoder takes the new content and converts it from a single digital format to multiple formats, or variants.

- The distribution encoder also creates and sends SCTE-35 data with the variants. SCTE-35 data maps out when there are ad breaks and how many ads should appear within the particular piece of content.

- **[Optional] Digital rights management (DRM)**

You'll only need this technology if a partner needs to encrypt digital content to comply with syndication or business rules; for example, if content can only be streamed in certain regions or if users have to authenticate in order to view it. You can protect your content with Token auth and other services where DRM isn't mandatory, but DRM provides encryption keys to the packager during the packaging step and validates that key when the content is being requested by the end user. If both keys are approved, the device is authorized to play the content.

- **Packager**

Once the distributing encoder creates the variants and SCTE-35 data, the packager takes each variant stream and packages it all together in an adaptive streaming format such as HLS or DASH with SCTE 35 metadata and DRM (if applicable). The packager then places the transport content streams into the CDN.

Additional prerequisites

Ad Manager supports many options for delivering video ads to a wide variety of inventory, but DAI has additional prerequisites which ensure that only premium quality ads are matched with your premium content.

- **All video ads must be able to be transcoded**

Transcoding means taking one video format and converting it into another format. A transcoded ad can also be formatted into different variants to change its compression ratio or quality.

- **All video ads must be VAST-compliant**

The IAB's Video Ad Serving Template (VAST) is a template format designed to standardize communication between video players and servers in order to facilitate traffic across numerous publishing platforms. All video ads used with DAI must be VAST-compliant. [Google Ad Manager supports up to VAST version 4.2](#), including versions 3 and 2 as well.

- **Set up unique ad units for DAI inventory**

When you create your DAI network in Google Ad Manager, make sure you set up DAI-specific ad units. This will reduce trafficking errors while letting you track and report DAI campaign delivery and prevent noncompliant ads from being targeted to DAI inventory.

- **Ingest video ad content into the DAI CDN**

Before you deploy your DAI campaign for live, linear, or VOD, video ad content must be ingested or uploaded into the Ad Manager DAI CDN, where it will be transcoded into the required HLS or DASH variants. If ingestion and transcoding aren't done prior to launch, they'll occur on the first monetizable impression. You should set up tests for [DAI campaigns](#) to ensure all video ads are eligible, transcoded, and ready to serve to your live campaign.

Implementing adaptive streaming

Once you've checked off the key prerequisites, your next step is to implement adaptive streaming to ensure an optimal viewing experience. Ad Manager DAI can stitch ads into live linear streams, live event streams, and on-demand content in two adaptive bitrate video streaming standards:

- HTTP Live Streaming (HLS)
- Dynamic Adaptive Streaming over HTTP (DASH)

Deciding which standard to use will depend on your business; both work with Ad Manager's DAI, and while DASH is newer than HLS, both standards are dynamic, adaptive, and support 4K streaming. Let's take a look at how DAI can be implemented for both streaming standards.

HTTP live streaming (HLS)

HTTP Live Streaming, or HLS, is an adaptive bitrate streaming format that uses a manifest file, or m3u8 file, to define a playlist of segments. This manifest is generated by a packager or an encoder. This packager, or encoder, also chops the video into small video chunks called "segments" and uses SCTE-35 messages in the MPEG transport streaming standard to know when an ad break opportunity is available. The packager then uses these SCTE-35 messages to insert a tag into the manifest, which notifies Ad Manager DAI of an upcoming ad break.

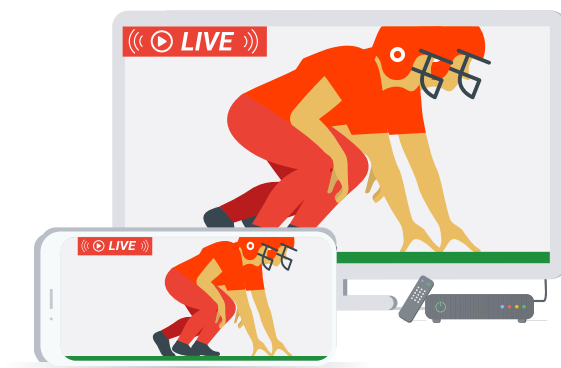
Here's an example of a playlist that's ready for DAI:

Master Manifest

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-STREAM-INF:BANDWIDTH=694272,CODECS="avc1.77.41,mp4a.40.2",RESOLUTION=768x432
chunklist_b694272.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=202752,CODECS="avc1.66.31,mp4a.40.2",RESOLUTION=480x270
Chunklist_b202752.m3u8
```

Variant Manifest

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:26
#EXTINF:10.000,
http://media.example.com/wifi/segment26.ts
#EXTINF:10.000,
http://media.example.com/wifi/segment27.ts
#EXTINF:6.000,
```




```
http://media.example.com/wifi/segment28.ts
#EXT-X-CUE-OUT:DURATION=30
#EXTINF:4.000,
http://media.example.com/wifi/segment29.ts
#EXTINF:10.000,
http://media.example.com/wifi/segment30.ts
#EXTINF:10.000,
http://media.example.com/wifi/segment31.ts
#EXTINF:6.000,
http://media.example.com/wifi/segment32.ts
#EXT-X-CUE-IN
#EXTINF:4.000,
http://media.example.com/wifi/segment33.ts
```

HLS best practices

Follow these practices to optimize your HLS stream for Ad Manager DAI:

HLS playlists and rendition groups

Deliver playlists using GZIP content-encoding (recommended). Variants in master playlists and segments can use either absolute or relative URLs. All ad assets are transcoded to include I-FRAME playlists, including slates. Ad Manager DAI doesn't generate I-FRAME playlists from the content or generate new Rendition Groups based on video/audio combinations.

- **For live/linear**

I-FRAME playlists must be generated and conditioned by the encoder/packager. Rendition Groups are delivered as they're generated from the Live encoder.

- **For video on-demand (VOD):**

I-FRAME playlists, if required, must be supplied in the master manifest at ingest. Ad Manager will only raise an error if the provided I-FRAME playlist file can't be accessed or parsed. Rendition Groups are delivered as they are provided in the master manifest.

HLS codecs and resolution

HLS master playlists for live/linear streams and VOD assets must include the optional **CODECS** and **RESOLUTION** attributes, which are used to find ads that match these encoding settings. If these values aren't provided, ads are matched based on bandwidth alone, which could result in resolution or codecs mismatches between ad and content, poor user experiences, and playback issues across devices. The **CODECS** attribute must contain both the audio and video codecs.

DAI supports the **AAC**, **AC-3** and **E-AC-3** audio codecs.

HLS segment duration

HLS media playlists must have media segment durations declared as decimal-floating-point numbers in order to help determine the precise time that ad breaks will start and end. The less accurate the duration values, the worse the system's determination of when ad breaks will occur in the resulting stream. Also as a recommendation, the `EXT-X-TARGETDURATION` needs to be as close as possible to the average segment duration in order to avoid playback issues.

HLS timestamp alignment

The Presentation Timestamp (PTS) across segments in variant media playlists must align. P-frame and B-frame refers to other frames to decode a frame. The H.264 codec also references non-adjacent frames for better compression. Because these frames depend on other frames for decoding and presentation, Decode Timestamp (DTS) gives sequence by which frames are decoded and PTS gives sequence by which frames are presented. Aligning PTS is important when you adapt to different bitrates during asset playback.

HLS media encryption

If you want to encrypt your media content, it must be encrypted with AES-128 or SAMPLEAES. AES-128 encrypts or scrambles the TS while SAMPLE-AES scrambles individual media (audio or video). Apple Fairplay stream uses SAMPLE-AES. Ad Manager DAI supports FairPlay DRM technology for HLS streams.

Below is an example:

A media playlist with #EXT-X-KEY

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:26
#EXT-X-KEY:METHOD=AES-128,URI="https://media.example.com/asset/key1.
php?id=<signature>",IV=<initialization vector>
#EXTINF:9.901,
http://media.example.com/wifi/segment26.ts
#EXTINF:9.901,
http://media.example.com/wifi/segment27.ts
```

HLS video on-demand encoding

Conditioning for mid-roll

If you [serve mid-roll ads in your content](#), the video segments must be conditioned or preconditioned. Mid-roll ad breaks for DAI might not fall on your stream's typical video segment boundaries. For example, your content may normally be divided into 10-second segments, whereas you need an ad break at 0:36 (36 seconds). In this case, you need to divide a 10-second video into 2 separate segments that are 6 seconds and 4 seconds, respectively, allowing an ad to be inserted at the correct time. The process of dividing a video segment in order to enable an ad break is called "[content conditioning](#)."

In order to support the insertion of ads at the desired ad break locations, your content must either be preconditioned or able to be conditioned by Google Ad Manager DAI.

GOP size

Media must be encoded with a keyframe at least every 2 seconds or GOP (group of pictures). Content is prepared for mid-roll ad break insertion at the closest keyframe. If there's no keyframe within one second of the cue point, insertion of ads beyond that second will lead to a poor user experience during playback. To prevent this, if a keyframe can't be found within one second of the indicated cue point, an ad break won't be inserted.

In order to offer a fade-to-black TV experience around the ad breaks, we recommend that you provide 1 to 2 seconds of creative blacks around the cue points for a seamless transition between content and ads.

Closed captions / subtitles

The following closed caption / subtitle formats are supported:

- Input formats: TTML (sidecar), WebVTT (sidecar), CEA 608/708
- Output formats: TTML (sidecar), WebVTT (sidecar), WebVTT - HLS

HLS live/linear encoding

HLS linear ad insertion

Create **OUT** and **IN** cue points in the playlist representing the ad insertion splice points for a linear broadcast or live feed:

```
#EXT-X-CUE-OUT:DURATION=<time>  
#EXT-X-CUE-IN
```

The **EXT-X-CUE-OUT** tag represents the splice start point and **EXT-X-CUE-IN** represents the splice end point. The **EXT-X-CUE-OUT** tag should be terminated by an **EXT-X-CUE-IN** tag indicating the end of the ad break. Ad Manager DAI replaces the entire section between the **EXT-X-CUE-OUT** tag and **EXT-X-CUE-IN** with video ad

content sourced from the media files ad server's VAST response.

The `DURATION=<time>` is required and is the approximate duration in seconds or floating point of the ad break (in other words, the difference in the time between the `EXT-X-CUE-IN` and `EXT-X-CUE-OUT`). Alternately, `EXT-X-CUE-OUT: <time>` can be substituted for `#EXT-X-CUE-OUT:DURATION=<time>`.

Dynamic adaptive streaming over HTTP (DASH)

Similar to HLS, Dynamic adaptive streaming over HTTP (DASH), also known as MPEG-DASH, is an adaptive bitrate video streaming standard that enables high-quality streaming of video content over the internet. DASH provides support for Digital Right Management (DRM) technology like Widevine, which enables you to deliver premium streaming content with protections against unauthorized access or theft.

Ad Manager supports DASH for both [video on-demand and live/linear streams](#), including the following features under the ISO standard:

- ISO-BMFF on-demand profile and ISO-BMFF Live profile
- Digital Rights Management (DRM) through MPEG-DASH Common Encryption (CENC)
- Support for Widevine and Playready
- Subtitle support for WebVTT and TTML (in-band and out-of-band in MPD)
- Closed caption support with "CEA608/708"

Learn more about these features in the [ISO DASH standard](#).



DASH video on-demand

Media presentation description (MPD) requirements

HTTPS must be used for the entirety of the DASH content referenced in the same MPD.

VOD/on-demand content MPD must be declared as `type="static"`.

- Each period in the MPD must specify a duration attribute.
- The ISO-BMFF on-demand profile must adhere to the ISO-BMFF on-demand profile of the [ISO DASH standard](#).

Media requirements

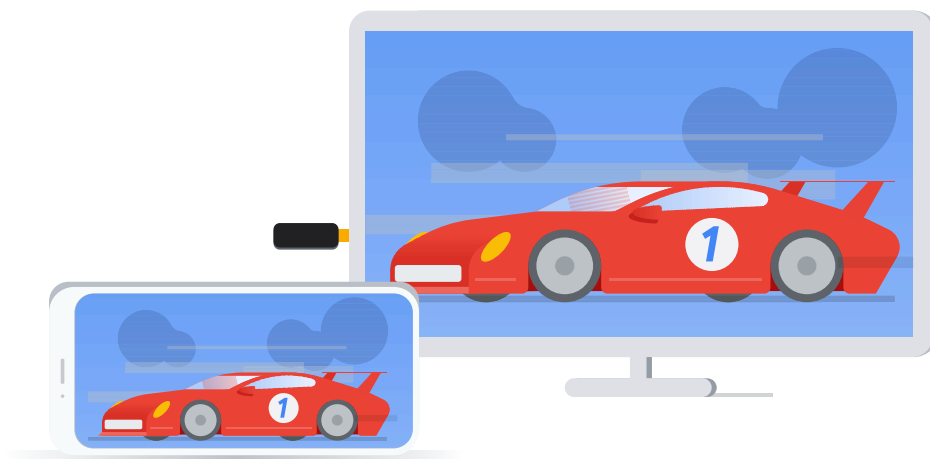
- The ISO-BMFF on-demand profile must adhere to the ISO DASH standard's segment format constraints.
- DASH content must follow all the rules specified in the [DASH-IF Interoperability Points](#).

Media encryption and DRM

If DRM is enabled, the MPD must contain the `<ContentProtection>` element and follow the syntax defined in the DASH spec. Ideally, the PSSH (Protection System Specific Header) box content should be present in the manifest.

Widevine example

```
<ContentProtection
  schemeIdUri="urn:uuid:EDEF8BA9-79D6-4ACE-A3C8-27DCD51D21ED">
  <cenc:pssh>BASE64_PSSH</cenc:pssh>
</ContentProtection>
```



DASH live/linear

Content guidelines

The content should be preconditioned based on the following:

- Media presentation must be broken into multiple periods, with each mid-roll **CUE-OUT** and **CUE-IN** falling on a period boundary.
- Content segments at the beginning and end of each period must be prepared so that every frame in the segment is included in the period.
- Cue points must align with a keyframe/IDR frame at the beginning of the period. This may require ending periods with a short segment when the cue point doesn't fall on a natural boundary. Learn more about cue points [here](#).

Ad break signaling

All content related to a single ad break must be included in exactly one period, which in turn must only contain the content segments to be replaced. The duration of the ad break specified in the **CUE-OUT** event should be exactly equal to the total duration of all the segments in the period.

- The ad break period must signal **CUE-OUT** based on the formats specified in these [samples](#).

MPD requirements

HTTPS must be used for the entirety of the DASH content referenced in the same MPD.

- Live/linear content MPD must be declared as `type="dynamic"`.
- Each period in the MPD must specify a valid start attribute.
- The ISO-BMFF Live profile must adhere to section 8.4 of the [ISO DASH standard](#).
- Addressing schemes are restricted to templates with number and time-based addressing.

MPD updates

- New periods can be added to the MPD at the end.
- Old periods from the top of the MPD can be removed.
- The last period in the MPD is the only one that should be updated.
 - Updates should only be for adding new segments to the last period.
 - Segments cannot be removed from the last period. [DASH-IF Interoperability Points](#).

Media requirements

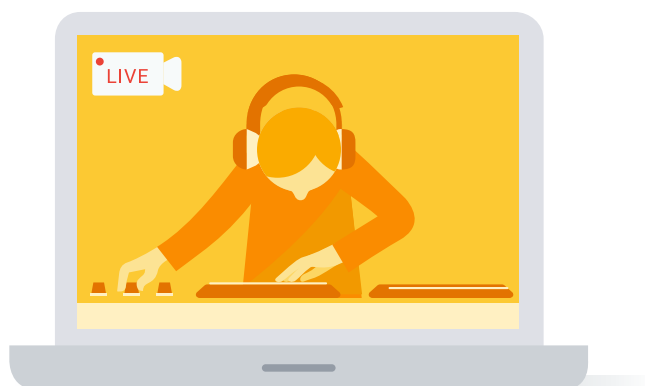
- The ISO-BMFF Live profile must adhere to section 8.4.3 (Segment format constraints) of the [ISO DASH standard](#).
- DASH content must follow all the rules specified in the Service Offering Requirements and Guidelines of the [DASH-IF Interoperability Points](#).

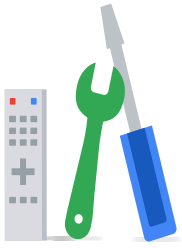
Media encryption and DRM

If DRM is enabled, the MPD must contain the <ContentProtection> element and follow the syntax defined in the DASH spec. Optimally, the PSSH (Protection System Specific Header) box content should be present in the manifest.

Widevine example

```
<ContentProtection  
  schemeIdUri="urn:uuid:EDEF8BA9-79D6-4ACE-A3C8-27DCD51D21ED">  
  <cenc:pssh>BASE64_PSSH</cenc:pssh>  
</ContentProtection>
```





Part 2: Implementing Dynamic Ad Insertion

Generating ad tags in DAI

Once you've set up video inventory for Dynamic Ad Insertion and your stream, you can [generate an ad tag](#) that works with your ads that are eligible to serve as DAI inventory. The tag generation process is generally the same as any other video ad tag. However, you do need to make some tag modifications in order to ensure compatibility with either a live/linear stream or for video on-demand content.

When you generate an ad tag, you can select an existing tag or create a new ad unit. We strongly recommend that you create unique ad units for DAI, which allows you to assign specific creatives to specific inventory and run reports exclusively on DAI delivery.

Best practices and recommendations for ad tags

Live Linear

- Must use `https://`
- Ad tags must use the URL and path: `https://pubads.g.doubleclick.net/gampad/live/ads`
- Must set a value for the `sz` and `iu` parameters
- Must set the `output` parameter to `xml_vast3`
- Must set these parameters to these values: `impl=s`, `env=vp`, `gdfp_req=1`, `unviewed_position_start=1`, `ad_rule=0`
- (Optional) You can control which creatives are eligible to serve based on a configuration set up in a [video creative profile](#) by adding the `pp` parameter and associated profile name.

Video on-demand

- Must use `https://`
- Ad tags must use the URL and path: `https://pubads.g.doubleclick.net/gampad/ads`
`https://support.google.com/admanager/answer/7125558`
- Must set a value for the `sz` and `iu` parameters
- Must set the output parameter to `xml_vmap1`
- Must set these parameters to these values: `impl=s`, `env=vp`, `gdfp_req=1`, `unviewed_position_start=1`, `ad_rule=1`
- Add the `cmsid` and `vid` parameters and assign them to their macros. It should look like this:
`&cmsid=%%CMS_ID%%&vid=%%VIDEO_ID%%.`
- (Optional) You can control which creatives are eligible to serve based on a configuration set up in a [video creative profile](#) by adding the `pp` parameter and associated profile name.

Learn more about any of these [video ad tag parameters](#), or see details about the [video content macros available for Dynamic Ad Insertion](#).

Platform support

Now let's take a look at support for Dynamic Ad Insertion platforms. Please refer [to this site](#), which provides an overview of support for apps. To Implement DAI, you have three options, depending on the platform:

- **Google Interactive Media Ads Software Development Kit (IMA SDK)**
IMA SDK is our recommended approach. This is the same SDK that enables players to make client-side video ad requests and parse the VAST responses, but we've added new methods that enable a player to request a VOD stream or live stream. The IMA SDK automatically allows your player to send Ad Manager signals that maximizes programmatic demand, ensures spam protection, and provides more accurate reporting and event tracking. [Learn more](#).
- **Google Dynamic Ad Insertion API (DAI API)**
DAI API allows the player to make multiple direct HTTP requests to the Ad Manager DAI system but is responsible for reporting metrics back to Ad Manager through a combination of client-side and server-side architecture. We recommend this API for devices like Smart TVs which lack IMA SDK support. [Learn more](#).
- **Server-side beaconing (SSB)**
SSB enables the player to make a direct HTTP request to DAI, where Ad Manager does estimates for ad metric reporting during the stream play. Any HLS player should be able to play Ad Manager DAI SSB streams, but there are some limitations, including lack of programmatic support. [Learn more](#).

Optimal player development

When you develop an application, your aim is to offer the best possible user experience while also maximizing your revenue. Here are some of the video player controls you can develop into your app to make both happen.

- **Total Ads Countdown and Per Ad Countdown**

This lets you add a countdown of time remaining for the total number of ad(s).

- **HTML5, iOS and Android:** The IMA SDK offers this optional feature.
- **Roku and tvOS:** Within the IMA SDK, the advanced sample app has an example of this feature, but you'll need to include custom code if you want to use it.
- **Other devices:** For other devices without IMA SDK, DAI API has a payload that gets returned containing the above information.

- **Closed captions/subtitles**

This gives you the ability to add closed captioning within the content stream. These are the subtitle programs that are supported:

- **Timed Text Markup Language (TTML) (side-car):** [TTML](#) is intended for transcoding or exchanging timed text information among legacy distribution content formats that are being used for subtitling and captioning.
- **WebVTT (side-car):** [WebVTT](#) files provide captions or subtitles for video content, textual video descriptions, chapters for content navigation, and, more generally, any type of metadata that's time-aligned with audio or video content.

- **Bookmarking**

Bookmarking is the ability to save and then return to a specific point in the content stream. Suppose a user watches five minutes of content, leaves the same video stream, and then returns to it. Bookmarking saves the user's position in the stream so the stream can pick up from where it left off, providing a seamless experience to the viewer.

- **Snapback**

[Snapback](#) is the ability to take the user back to the start of the ad break they seeked past and then return them to their seek location after that ad break is over.

- **Scrubbing**

Typically a long-form content feature, this control shows the first frame of each piece of content as the user scrolls through the content bar.

All [advanced sample apps](#) demonstrate feature functionality across the various supported platforms. The cue point information feature can also be demonstrated within [basic sample apps](#).

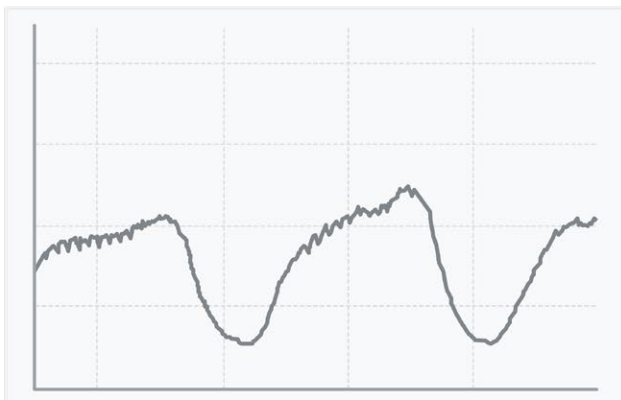


Part 3: Best practices for managing live events with DAI

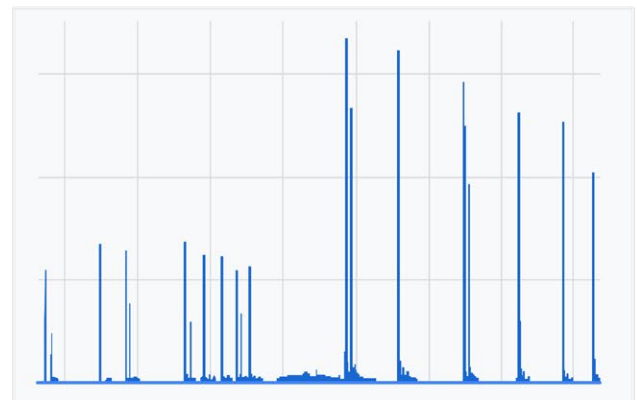
As audiences watch TV and video across more screens and platforms, delivering a seamless ad experience can be challenging, given the growing number of ad formats, operating systems, measurement requirements, and streaming formats that must be taken into account.

It's even more challenging to deliver a seamless, reliable ad experience within live content, whether it's sports, news, or other big events. One challenge is that all your users are tuned in at the same time; it's critical that your technology partner has the infrastructure and serving capacity to manage millions of concurrent viewers. This is especially important as the stakes are much higher for live events – stream failure means not only loss of ad revenue but, worse, loss of user confidence in your platform.

Traditionally, this is how traffic differs between on-demand traffic and linear/live traffic:



Ad requests for on-demand video



Ad requests for live / linear video

As you can see, live events have several simultaneous ad request spikes, whereas nonlinear or on-demand video ad requests form more of a rolling pattern, which puts less pressure on the ad server.

We've invested in end-to-end infrastructure, product features, and support to ensure that ads during your live events are delivered successfully:



- Our **protection layer** intelligently manages live traffic spikes by utilizing Google's global infrastructure
- We enable you to maximize your revenue through [DAI Prefetch](#), which distributes spikey ad requests more evenly so you can monetize more concurrent viewers, and [Early Break Notification API](#), which allows you to maximize yield and programmatic demand.
- **Sophisticated detection and mitigation** lets us ensure that content streams always continue, even if an ad stream fails.
- We offer **24/7 real time end-to-end support** specialized in network infrastructure and systems integrations before, during, and after the event to ensure full coverage.

These best practices will also help you optimize your ads while delivering a quality viewing experience to audiences everywhere.

1. Plan for live events

Live events are exciting, but it's less exciting – or the wrong kind of exciting – to have a content or ad delivery issue during an event. A critical aspect of provisioning your content and planning ahead is determining your number of projected concurrent users – the maximum number of users who watch the stream at the same time. This is one of the biggest concerns for ad serving and server side ad insertion stream scaling. Here are a few tips on making this challenging estimate.

- **Leverage Historical Data**

- **Use estimates based on prior events**

- These typically accurate numbers are the gold standard. Here are some questions to consider when you're studying historical data:

- **Are the implementations the same?**

- Was traffic only on desktop last year but now on mobile, Roku, and AppleTV as well? You'll want to track any factors that could change traffic levels from prior events.

- **Is the stream authenticated?**

Unauthenticated streams get higher traffic. If a prior event had authentication and this new one doesn't, this obviously could affect your concurrent estimates.

- **Is the player distributed in the same way?**

If last year's event only appeared on your own site, and this year's event is syndicated across 200 partner sites, you should expect more traffic.

- **Is the new event news or sports- related?**

Traffic in these categories can change for unforeseen reasons. This can also impact news linear streams, which can go from few users to many based on breaking news.

- **Regionality**

Is the live event specific to a region? This is important information to provide to Ad Manager DAI because it will help ensure that sufficient resources have been provisioned for that region. Unexpected traffic spikes in different regions can often be challenging. So plan ahead!

- **Optimal DVR window**

The DVR Window is the duration a viewer is able to scrub back to when playing a live stream. Depending on the size of this window, this can impact serving and storage capacity. Make sure you discuss this issue with your Google account team.

- **Ad inventory**

When it comes to popular live events, we've seen partners burn through their ad inventory quickly. The best practice here is to have a conversation with the Ad Operations team so it's aware of the live event and is prepared with sufficient inventory.

- **DAI Prefetch**

For large events, enable [DAI Prefetch](#) to ensure there's sufficient time to decision the ads. This feature uses a 2-staged approach, first allowing partners to prefetch a portion of their next ad break, say the first ad for each viewer, at the end of the prior ad break. Then, when the ad break begins, Ad Manager serves the first ad that was prefetched and in parallel fetches the subsequent ads based on information such as ad break duration or custom parameters related to events in a game like a goal or touchdown. DAI Prefetch distributes ad requests more evenly and accommodates longer ad decisioning times, which helps solve for surges in traffic during live events. This enables you to deliver a high-quality ad experience to millions of concurrent viewers while maximizing fill rates and revenue, and increases the ability for the broader ads ecosystem to handle the scale of large live events.

- **Early Break Notification API (EBN)**

[Early Break Notification API](#) makes it easier to use real-time ad break data to monetize live events. This API allows you to send ad break data to Ad Manager, such as program information from SCTE-35 or custom parameters, in advance of an ad break so that you can deliver relevant ads and adhere to brand safety requirements. Data may include program ID, break number, or custom parameters such as whether the ad break is at half-time or the end of a game. This enables you to deliver ads based on these circumstances, such as a food delivery ad, sponsorship ad, or an ad based on who won the game. Sending ad break data in advance also provides

additional ad decisioning time across programmatic demand sources, which is especially important during live events and can help you increase fill rates. You can use DAI Prefetch and Early Break Notification API together to optimize for a great viewing experience while maximizing yield.

2. Build for failure

We live in an imperfect world. Unexpected events will happen, so it's important to understand where things can go wrong and what you can do about it. In other words: What's your fault tolerance strategy? Although this is especially critical for live events, it's also important to consider for VOD content as well. Here are some common failures you should be ready for:

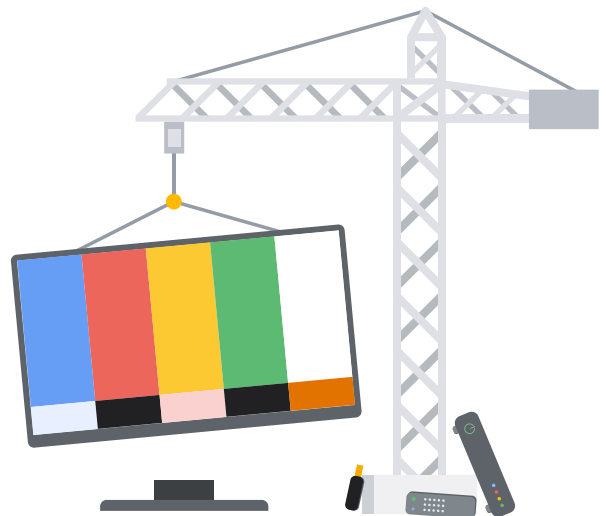
- **Encoder failover**

In a typical SSAI scenario, you can have a primary and backup encoder. While both streams are presented to the SSAI solution, at times one encoder goes offline for various reasons. Address these questions to make sure you're prepared:

- How is ingest or stream playback affected when you have an encoder failure?
- Is the user's viewing experience affected?
- Are alerts set up to ensure that you're aware of and have planned for this type of failure?

- **Stream request failure**

Unlike a client-side video implementation where the video stream and ads are requested independent of one another, Google Ad Manager DAI's server-side implementation requests only one stream, with ads dynamically stitched into the video content. This means that if an error occurs the stream can be blocked, which means both a poor user experience and lost revenue. That's why DAI provides you with HTTP errors detection when you create a stream as well as when the stream is running, and lets you specify a backup stream in case you don't get a valid response from the server.



- **Handle HTTP errors detected when a stream is created**

Error type	Recommendation
4XX errors (excluding 429)	When you see an HTTP 4XX error, you should not fall back to the raw streams, as the client likely experienced an error in these instances. Make sure that the request you're sending to the server is correct and contains all of the required parameters.
429 or 5XX errors	When you see an HTTP 429 or HTTP 5XX error, you should fall back to the raw backup stream, without monetization. In the IMA SDK, these errors can be caught with an error handler, and the default stream can be switched to the backup stream.

- **Ad request failure**

Your Ad Ops team will help you configure the appropriate ad-tag for your Google Ad Manager DAI configuration. But sometimes the ad server response doesn't exactly match the ad break duration, so you still need to answer these questions in order to mitigate potential failure:

- **What do you do when you have a 120-second ad break but the ads that were returned are only 80 seconds?**

Google Ad Manager DAI lets you handle this in multiple ways while you create your live stream. You will find the following two config options in the Live Stream UI:

- **Empty ad break fill type**

- This is the kind type of video content displayed in an ad break when the VAST response is empty. When an empty VAST response is received, you can choose whether you want to display slate or underlying content.
- **Slate**
Displays video content that you've created as slate to fill the break.
- **Underlying content**
Displays live stream content to fill the break.

- **Underfill ad break fill type (beta)**

- The type of video content displayed in an ad break when there aren't enough ads to fill the entire break. This is currently only supported for HLS.

- **Slate**
Displays video content that you've created as slate to fill the break.

- **Underlying content**
Displays live stream content to fill the break.

- **Use Ad Buffet fallback functionality**

[Ad Buffet](#) can be used to provide fallback functionality to [optimized pods](#). It refers to standalone ads that are available to serve in the pod but weren't part of the original ads to be served in that pod. Optimized pods serve ads in a specific sequence. Ad buffet ads are served using their own sequence and can ensure there's a fallback ad available in cases of ad failure.

3. Leverage data and standards

Analytics

Analytics insights can help you optimize your DAI setup. Although this is important for all content types, these insights are especially critical for ensuring that your live events run smoothly.

- **Bitrate profiles**

One important decision you'll make while setting up any SSAI solution is which bitrates to configure.

- What are the minimum and maximum bitrate/resolutions that the system should serve to users?
- There are a number of recommended bitrates for various content types. Are you fully utilizing them?

The best practice is to start by reviewing your bitrate profile usage. Let's say you have 8 bitrates configured but only six of them are heavily being used by your audience. Removing some unused bitrates can save you transcoding as well as hosting costs.

- **Bandwidth**

The best practice here is to leverage analytics to get critical insights on user behavior such as where they're consuming your content. You should ask yourself:

- What types of bandwidth are the users on?
- Are most of them consuming content at home on an AppleTV or Roku, or are they on the go in an area where they lack high-speed internet?

Google Ad Manager DAI enables you to tackle these issues by overriding stream variant parameters. Overriding the variant selection and order parameters during stream creation time allows you to specify the subset of variants that should be included in the response to a master playlist request and the order of those variants. The subset and order of variants will remain the same for the life of the stream, unless you explicitly override them. Learn how to [override stream variant parameters](#).

- **Latency**

Measuring latency is easy on a browser but a bit more challenging on OTT devices. The seconds from when users click play to when the stream actually starts playing are critical, and it's essential that you have a plan to measure them. Ask yourself:

- What is acceptable latency for my users?
- Do I have analytics in place that measure the latency that our users are facing?
- How many users are dropping off due to slow load times before my video even starts playing?

VAST reporting and features

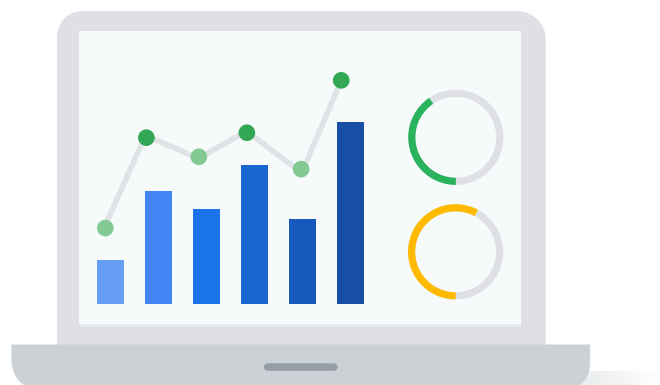
Video Ad Serving Template (VAST) is a specification that sets the standard for communication requirements between ad servers and video players. IAB has released a number of updates in the 4.0 release, and they can help address some of the challenges we face in the evolving video landscape. Here's how.

- **VAST Error Codes in DAI Reporting**

Video players can trigger both fatal and nonfatal errors when it comes to playing video ads. VAST 4 increases the error codes that can be used to debug problems with ad serving. It's important to be able to report on these errors and have an action plan to address them.

- One of the most common errors faced by our partners is the VAST error 402, i.e., "Unable to download or timeout of MediaFile."
- While the cause for this error isn't always clear right away, it is important to look for patterns associated with this type of error.
- Is this timeout occurring for a particular third party ad server, or is there something else that's troublesome with that creative type.

We encourage our partners to spend time reviewing these error codes and have plans to address them.



- **Mezzanine file**

To support advertising across video platforms that include long-form content and high-resolution screens, VAST 4 features include support for a raw, high-quality mezzanine file. The mezzanine file is very large and can't be used for ad display, but ad-stitching services and other ad vendors use it to generate files at appropriate quality levels for the environment in which they play.

- **Universal ad ID**

Today, different identifiers can be used for the same creative, or the same identifier can be used for multiple creatives. Not only can this lead to a poor user experience, it can also be difficult for your ad-ops team to troubleshoot creative issues. In VAST4, the new Universal Ad ID feature can be used to identify a creative that's maintained across different systems.



Part 4: Troubleshooting common challenges with DAI

Locate a DAI session ID or debug key

Every user stream is given a session ID, which is saved for six hours after an event. Monitored sessions are saved for 60 days. You can enter the session ID into the “Sessions identifier” field of the **Stream Activity Monitor (SAM)**, and the search results will allow you to see details about the user ad requests in order to troubleshoot a video stream.

Find a session ID in an app that uses the IMA SDK

When you use the IMA SDK to create a video stream, you can call a method to obtain the session ID. Here are links to details about how to obtain the ID on each platform’s SDK:

- [Android SDK](#)
- [iOS SDK](#)
- [tvOS SDK](#)
- [HTML5 SDK](#)
- [Chromecast SDK](#)
- [Roku SDK](#)

Many application developers choose to capture and log the session ID so that it’s available for stream troubleshooting. Because the session ID is essential for looking up details about ad delivery in a user’s stream, it’s helpful to implement a method or hidden feature (e.g. Konami code) in your application to easily display the DAI Session ID.

Find a session ID when you inspect HTTP traffic

You can monitor HTTP requests in a proxy tool from an application that uses the IMA SDK. When you use the IMA SDK to request a DAI video stream, there are two HTTP requests:

- The HTTP POST, which should return with a 201 status, indicating that a new stream has been created.
- The HTTP GET request to the URL in the stream_manifest field.

Additional information is returned in a JSON structure, including the stream_id. This is the session ID you should enter into the “Sessions identifier” field in SAM. Learn more here.

Stream activity monitor (SAM)

You can monitor and debug a live stream video for a specific stream session. The stream activity monitor (SAM) allows you to see details about the numbers of ads, ad breaks, fill and served durations, and troubleshoot issues in real time. For live and linear streams, SAM automatically updates as new ad breaks occur in the session.

How do I access SAM?

1. Sign into Google Ad Manager
2. Click **Video**
3. Click **Stream Activity Monitor**
4. Enter a [session ID](#)
5. Click **Debug Stream**

When you click to debug a session ID, SAM displays information about the session, including your request URL, content, and filled percentages, broken down by ad break. The following metrics are displayed:

Metric	Description
Streaming time	Total time elapsed including content and ads
Number of ads	Total number of ads available for the associated ad break
Filled duration	Combined duration of the ads that were chosen to serve during the ad break
Served duration	Combined duration of ads that actually served during the ad break
Served percentage	Percentage of ad break duration that actually served

Stream session ID ? 1234ABCD-5678-EFGH-9999-11AA5555

[Debug session](#)

Demo Event

Event code: 0000000001

User agent:

Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36

Request URL: /ssai/event/hml0HxBhQWSGu7f1Kb2LHg/master.m3u7

Session start time ?

11:08 AM EDT
10/13/2016

Session duration ?

15 mins 10 secs

Event start time ?

3:40 PM EDT
10/12/2016

Event end time ?

3:00 AM EST
1/1/2037

Filled ad break time ● > 95% ● between 50 - 95% ● < 50%

[Select columns](#)

Ad break	Streaming time	Number of ads	Ad break duration ?	Filled duration ?	Served duration ?	Served percentage ?
2	11:20:32	5 ads	120s	120s	120s	● 100%
1	11:12:32	3 ads	120s	120s	120s	● 100%


Rows per page: 20 ▾ 1-2 of 2 < >

Please note that this image is an example, and the statistics for each publisher will vary.

Live stream monitor (LSM)

You can monitor and troubleshoot a live stream video, to view either a quick snapshot of how the stream is performing or an historical log of all the activity that occurred within the stream.

How do I access LSM?

1. Sign into Google Ad Manager
2. Click **Video**
3. Click **Live Streams**
4. Find the live stream you'd like to monitor or troubleshoot
5. Click **Live stream monitor** to see an overview of how the stream is performing, including any errors or warnings.
6. Click the down arrow  in each table row to see details about an activity, including the master playlist status and details about any errors and warnings that may have occurred while fetching or parsing the playlist.

Live streams >
 Live stream: **Sample Stream**
 Status: Active

Settings | Live stream monitor

Feb 9, 2018 | 3:00 PM | NOW | < PREVIOUS HOUR | NEXT HOUR >

Time	Variants	Variant count	Serving status ⓘ	Activity type ⓘ
Feb 9 3:07:48 PM		8		Ad break cue out (30s) ▾
Feb 9 3:07:38 PM		8		Variant ▾
Feb 9 3:06:47 PM		8		- ▾
Feb 9 3:06:29 PM		8		Variant ▾
Feb 9 3:06:19 PM		8		- ▾
Feb 9 3:06:08 PM		8		Variant ▾
Feb 9 3:05:39 PM		0		Variant ▾

Please note that this image is an example, and the statistics for each publisher will vary.

See details about stream activities

The live stream monitor displays a log of all major stream activities that occurred within a specified hour. You can change date or time to view historical data for the last 15 days.


- **Variants:** Provides a representation of the status of variants.
- **Variant count:** Shows the total number of variants from which an activity occurred.
- **Serving status:** Indicates whether the activity has triggered any errors or warnings.

- All variants are serving properly
- Not all variants are serving properly
- The stream is not serving

- **Activity Type:** Describes the type of activity that occurred. All major stream activities are shown, and each can be clicked for more information.

- **Ad break cue out:** A CUE-OUT marker has been observed in at least one variant playlist.
- **Variant:** Variants were updated as part of normal ingest.
- **Variant added:** A new variant is added to the master playlist.
- **Variant removed:** A variant is removed from the master playlist.
- **Short lived variants:** Too many variants are being quickly added and removed. You might consider updating the “Master playlist refresh” live stream setting to “Manual”.
- **Too many variants:** Error or other issue in the encoder setup. This can lead to degraded stream performance.

Status of master and variant playlists



When you click the down arrow  to expand an activity table row, you see details about master and variant playlists.

Master playlist information

- Master playlist status
- URLs
- Errors related to fetching or parsing, along with the HTTP error code

Variants information

- Variant status

-  Variant is serving properly
-  Variant is not serving properly

- URL (Click  to see the contents of a playlist URL)

- Bandwidth

- Resolution

- Codec

- Errors

- “Error parsing the variant HLS”

- “Variant was reset”

- “Variant playlist is stale”

- Other error text and HTTP error code encountered while fetching the variant



Get started with DAI



Now that you know how to prepare to implement Google Ad Manager's DAI, have become familiar with which best practices to follow to manage live events with DAI, and have tips on how to troubleshoot common challenges, you're ready to get started with Dynamic Ad Insertion for your video business.

To get started, contact your Google Ad Manager Account Manager and ask for Dynamic Ad Insertion to be enabled.

You can also visit our [Dynamic Ad Insertion help center](#) or the [Ad Manager website](#) for more information.

