



OOYALA PLAYER V4 DEVELOPER GUIDE

CONTENTS

COPYRIGHT NOTICE	5
PLAYER V4	6
USER INTERFACE FOR THE PLAYER V4 SKIN	11
SUPPORTED BROWSERS AND OPERATING SYSTEMS FOR PLAYER V4	22
SYSTEM REQUIREMENTS BY PLATFORM FOR PLAYER V4	23
FUNCTIONAL SUPPORT ACROSS ENVIRONMENTS FOR PLAYER V4	25
INTRODUCING PLAYER V4	29
Player V4 FAQ	29
User Interface for the Player V4 Skin	33
Supported Browsers and Operating Systems for Player V4	43
System Requirements by Platform for Player V4	44
Functional Support Across Environments for Player V4	46
Integrations in Player V4	49
GETTING STARTED WITH PLAYER V4	51
Prerequisites for Using Player V4	52
Basic Tutorial for Player V4	53
MIGRATING TO PLAYER V4	57
Migrating from Player V3	57
Differences Between Player Versions	61
Player Migration FAQ	66
EMBEDDING PLAYER V4 ON A WEB PAGE	69
Player V4 Plugins	69
Specifying Plugins for the Standard Embed Code	75
Ooyala-hosted Player V4 Resources	77
Hosting Player V4 Resources	81
Cross-Origin Resource Sharing (CORS)	88
Basic Embedding Information	88
Loading Video Plugins	89
Using Flash-free Playback with Player V4	94



Embedding Player V4 in an HTML iframe	95
Playing YouTube Videos in Player V4 (Deprecated)	97
Page-level Parameters for Player V4	99
Examples of Player V4 Web Page Embedding	113
CUSTOMIZING THE PLAYER V4 APPEARANCE	116
Order of Precedence in Player V4 Settings	118
Customizing the Player V4 Skin with skin.json	120
Customizing the Player V4 Appearance Using CSS	122
Customizing Player V4 Controls with Graphics	124
Localizing the Player V4 UI	124
Player V4 Skin and the React Application Framework	125
Closed Captions in Player V4	125
Customizing VR 360 Player Controls	131
Adding a Custom Branding Overlay to Player V4	132
DISCOVERING CONTENT IN PLAYER V4	134
USING PLAYLISTS IN PLAYER V4	136
ADS IN PLAYER V4	142
FreeWheel Ad Plugin	143
Integrating FreeWheel Ads	143
FreeWheel Ad Parameters	146
Google IMA Ad Plugin	147
Integrating Google IMA Ads	149
Google IMA Ad Parameters	157
Ooyala Pulse Ad Plugin	160
Integrating Ooyala Pulse Ads	161
Ooyala Pulse Ad Parameters	164
VAST and VPAID Ad Plugin	171
Integrating VAST and VPAID Ads	172
VAST and VPAID Ad Parameters	175
Integrating Ads	176
Ads Integration Workflow	176
Integrating Multiple Ad Managers	177
Configuring Ad Parameters	180
Assigning Ad Sets Dynamically	182
Ads and Live Streams	184
Delivering My Ads	184
About Ads	186
Ad Manager Framework	186
Ad Types	187
Ad-Related Events	188
ANALYTICS IN PLAYER V4	189
Adobe Analytics (Omniture) Plugin	190
comScore Analytics Plugin	196
Conviva Analytics Plugin	197
Google Analytics Plugin	199
Nielsen Analytics Plugin	204



YOOBORA Analytics Plugin	207
Analytics Framework	210
PLAYER V4 JAVASCRIPT API	211
About the Player V4 JavaScript API	211
Programming Closed Captioning Using Player V4 JavaScript	212
Event Model for Player V4	212
Message Bus API	213
Player V4 Message Bus Events	213
Event Publish and Subscribe	213
Listening to a Message Bus Event	214
Listening to Multiple Message Bus Events	216
Writing to the Message Bus	220
Information with Listeners and Method Calls	224
Programming Bitrates and Buffering for Player V4 Using JavaScript	225
Handling Errors for Player V4 Using JavaScript	227
Debugging Player V4	228
EMBEDDING PLAYER V4 IN ACCELERATED MOBILE PAGES (AMP)	230
USING PLAYER V4 IN FACEBOOK INSTANT ARTICLES	233
OOYALA PLAYER V4 RELEASE NOTES	235



COPYRIGHT NOTICE

All contents of the Ooyala website and Services are:

Copyright Ooyala, Inc. 2008-2018. All rights reserved.

Ooyala, Backlot, Ooyala Actionable Analytics, the Ooyala logo, and other Ooyala logos and product and service names are trademarks of Ooyala, Inc. ("Ooyala Marks"). Company agrees not to remove any Ooyala Marks that are contained within and/or affixed to the Services as provided to Company. Except with respect to the foregoing, Company agrees not to display or use in any manner the Ooyala Marks without Ooyala's prior written permission.

Ooyala and Backlot are registered trademarks of Ooyala, Inc. in the United States, Japan, and European Community.

For complete information on terms of service, see: <http://www.ooyala.com/tos>.

All other trademarks are the property of their respective companies.

This content was last updated on 2018-Jan-03.



PLAYER V4

Ooyala Player V4 is a next-generation video player that provides an engaging, personalized, responsive, content aware, and robust video playback experience that is consistent across devices and platforms.

Player V4 implements a customizable skin that can be applied to three products:

- **Ooyala Player V4 for HTML5** is a Web-based player that runs in popular browsers across desktop and mobile devices.
- **Ooyala Player Skin for iOS** enhances the Ooyala (video player) Mobile SDK for iOS by integrating the Player V4 skin in iOS native apps.
- **Ooyala Player Skin for Android** enhances the Ooyala (video player) Mobile SDK for Android by integrating the Player V4 skin in Android native apps.

For an introduction, see [Basic Tutorial for Player V4](#) on page 53.



BENEFITS

This version of the Ooyala player delivers the following benefits:

- High-quality, consistent performance across devices (robust video streaming support with performance enhancements)
- A robust, modular, flexible player skin template that follows unified Google Material UX Guidelines and can be customized for a branded playback experience
- Easy integration with advertising and analytics services

SUPPORTED PLAYBACK FEATURES

Player V4 supports the following features:

- An enhanced UX for the default player
- Customizable player UX
- Content-aware player controls that are able to distinguish between the supported content types (VoD, Live, In stream video ads) and display respective UX controls accordingly
- Responsive design (by default the player will resize dynamically when the size of the container of the player is changed)
- Full-screen and inline mode
- Closed captions (see [Closed Captions in Player V4](#) on page 125)
- Social sharing - email, embed (web only), Facebook (sharing the web page on Facebook), Twitter, and Google Plus by pointing to the hosted_at URL



- Discover related or recommended content on the player with the UpNext feature and Discovery tray on the Pause and End screens
- Cross-device Resume (XDR)
- Localization of player controls and error messages (English, Spanish, Simplified Chinese, or your custom localizations)
- Video plugins to support HLS and MP4, DASH and HLS, OSMF Flash HDS (Deprecated), and Akamai packaged HDS (Deprecated)
- Configurable HTML5 or Flash video rendering. Note that all of the plugins, ad logic, UI, Player APIs, etc. are always in HTML5.
- Live DVR, which allows the user, during live HLS streaming, to drag the slider backwards in time and to drag it all the way to the right to catch up to present time.
- Integration with playlists created using Ooyala Theme Builder

SUPPORTED AD FEATURES

Ads for Player V4 are supported through the [Ad Manager Framework](#) on page 186. This framework allows Ad Servers to develop ad plugins for Player V4. See below of a list of ad plugins that support major ad managers.

Ad Features

The following ad features are supported in Player V4:

- Pre-, mid-, and post-roll ads
- Clickthrough, overlay, and companion ads
- Podded and skippable ads
- Configurable ad countdown, timeout, and ad replay
- Basic ad settings and configurations via the Monetize tab in Backlot
- Page-level parameters that override Backlot settings
- Overlay Close and Learn More buttons
- The ability to configure ad plugins on the page level without involving Ooyala Technical Support
- My Ads configured in Backlot
- Live Server Side Ad Insertion with Ooyala Pulse

Ad Integrations

The following ad integrations are supported in Player V4:

- [FreeWheel Ad Plugin](#) on page 143
- [Google IMA Ad Plugin](#) on page 147 (DFP)
- [Ooyala Pulse Ad Plugin](#) on page 160
- [VAST and VPAID Ad Plugin](#) on page 171 (includes LiveRail - VAST ads only)

Note: The Ooyala mobile SDKs already support all major ad managers. We will continue our existing ad functionality support for the mobile SDKs.

IAB Standard Support

The following IAB standards are supported in Player V4:

- VAST 2.0 and 3.0 through the [VAST and VPAID Ad Plugin](#) on page 171
- VPAID 1.0 for Google IMA through the [Google IMA Ad Plugin](#) on page 147
- VPAID 2.0 through the [VAST and VPAID Ad Plugin](#) on page 171
- VMAP 1 through the [VAST and VPAID Ad Plugin](#) on page 171



SUPPORTED ANALYTICS FEATURES

Ooyala IQ automatically captures and reports Player V4 analytics events. Player V4 also supports integrations with third-party analytics providers through the [Analytics Framework](#). This framework allows analytics providers to develop analytics plugins for Player V4.

Analytics Integrations

The following out-of-the-box third-party analytics integrations are supported in Player V4:

- [Adobe Analytics \(Omniture\) Plugin](#) on page 190
- [comScore Analytics Plugin](#) on page 196
- [Google Analytics Plugin](#) on page 199
- [Nielsen Analytics Plugin](#) on page 204
- [YOUBORA Analytics Plugin](#) on page 207
- [Conviva Analytics Plugin](#) on page 197

SUPPORTED VIDEO STREAMS

The following platforms and streams are supported for this release. Future releases will support additional platforms. For details on which plugins to load for each stream type and browser type, see [Loading Video Plugins](#) on page 89. For details on which environments are supported for each platform, see [Supported Browsers and Operating Systems for Player V4](#) on page 22.

Ooyala Player	Stream Type	Supported Platform	Video Plugin
V4 HTML5	MP4	Desktop Web, Mobile Web (iOS and Android)	main_html5.min.js for all platforms and bit_wrapper.min.js for all platforms except iOS Mobile Web
V4 HTML5	HLS	Desktop Web, Mobile Web (iOS and Android)	main_html5.min.js for Desktop Web Safari, Desktop Web Edge, iOS and Android Mobile Web and bit_wrapper.min.js for all Desktop Web browsers
V4 HTML5	HDS	Desktop Web	osmf_flash.min.js (Deprecated) (Flash component used for video decoding and rendering of HDS) and akamaiHD_flash.min.js (Deprecated) for Akamai packaged HDS only
V4 HTML5	DASH	Desktop Web, Mobile Web (Android)	bit_wrapper.min.js
iOS SDK	MP4	Mobile App	N/A
iOS SDK	HLS	Mobile App	N/A
Android SDK	MP4	Mobile App	N/A
Android SDK	HLS	Mobile App	N/A



Ooyala Player	Stream Type	Supported Platform	Video Plugin
Android SDK	DASH	Mobile App	ExoPlayer (see Integration with Google ExoPlayer on Android for details)

SUPPORTED ACCESSIBILITY FEATURES

The Ooyala Player has been built to support [WCAG 2.0 AA standards](#).

Closed Captions Support

The Ooyala Player supports closed captions for playback, including popular input formats, language selection features, and configuration options. See [Closed Captions in Player V4](#) on page 125 for details.

Screen Reader Support

Validation testing for the Ooyala Player has focused on the following combinations of browsers and screen readers.

Browser	Screen Reader
Internet Explorer	JAWS
Microsoft Edge	Windows Narrator
Firefox	NVDA
Safari/Safari Mobile	VoiceOver
Google Chrome	ChromeVox
Chrome Mobile	Android TalkBack

Note: Other screen readers and browser combinations may work fine, but because screen reader and browser implementations vary significantly, other combinations may not work as well.

Keyboard Controls

The Ooyala Player supports keyboard controls. Using the tab key to move the focus to the various player controls, player functions can be activated as follows:

Action	Keyboard Control(s)
Play (Icon in video thumbnail)	Space or Enter key to Play/Pause
Play/Pause (control bar)	Space or Enter key to Play/Pause
Mute	Space or Enter key on the Speaker icon
Full screen	Space or Enter key on the full screen icon
Volume Bar	Left/Down to decrease volume, Up/Right to increase volume
Scrubber/Progress bar:	Up/Right to go forward 5 seconds. Left or Down to rewind 5 seconds (per action).

CUSTOMIZATION OPTIONS

For this release, the players (HTML5, iOS, and Android) offer the following degrees of customization:

- **No customization.** Use the default player with standard, out of the box settings.
- **Easy customization** via a JSON config file that applies across environments.



- **Advanced customization.** Developers can fork the player repositories and customize the source code.

The degree of customization depends entirely on your requirements. For details, see [*Customizing the Player V4 Appearance*](#) on page 116.

MORE ABOUT PLAYER V4



USER INTERFACE FOR THE PLAYER V4 SKIN

This topic describes the visual components of the new Ooyala player.

KEY COMPONENTS

Here are the key components of the player.



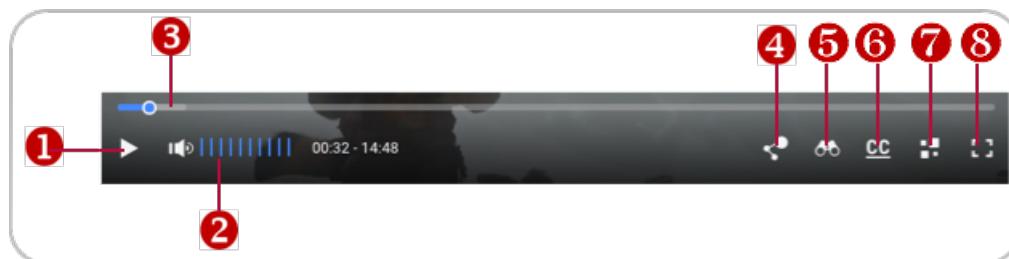
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	display area	Where the video is displayed.
2	control bar	Enables viewers to manually configure settings and move forward or backward through a video.

CONTROL BAR

The control bar (also known as a scrubber or slider) enables viewers to manually adjust settings and move forward or backward through a video.

Here are visual elements of the control bar in the READY or PAUSED state:



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	play button	Click to start video or audio playback.
2	volume control	<ul style="list-style-type: none">Drag the slider to adjust the volume.Click the volume control to mute the sound.



#	Name	Description
---	------	-------------

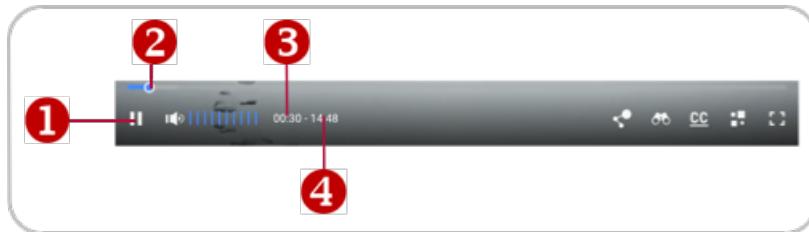


- (Safari only) Click the following control to unmute the sound:

SELECT TO UNMUTE

③	buffered content	Shows how much of the video has been retrieved in the buffer.
④	share button	Click to share this video with others via email, embed URL, or social media.
⑤	discover button	Click to discover related videos.
⑥	closed caption button	Click to select closed captions.
⑦	bitrate selection button	Click to display bitrate options.
⑧	full screen button	Click to display the video in fullscreen mode.

Here are visual elements of the control bar in the PLAYING state:



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
---	------	-------------

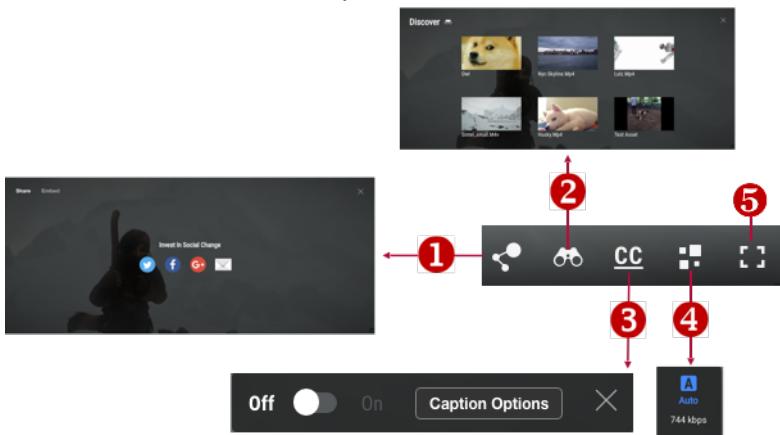
①	pause button	Click to pause playback.
②	scrubber (slider)	Drag and slide to manually move forward or backward through a video.
Note: Scrubber behavior depends on the playback context:		
<ul style="list-style-type: none"> • If a live stream is DVR, then the initial scrubber position starts at the end (right-most part) of the control bar (current time), and the user can seek backwards all the way to the earliest point available (left-most part). • If a live stream is <i>not</i> DVR, then the scrubber position starts at the left-most part of the control bar and has a duration of zero (the user cannot move the scrubber). • For VOD, the initial scrubber position starts at the left-most portion of the control bar, and the user can seek forwards (and subsequently backwards). 		



#	Name	Description
③	play head position	Current position during playback (minutes:seconds).
④	content duration	Total length of the video (minutes:seconds).

BUTTONS

The player displays buttons that can provide options to share the video on social networks, discover related videos, set closed captions and bitrates, and view the video in full-screen mode.



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	share button	Click to share this video with others via email, embed URL, or social media.
②	discover button	Click to discover related videos.
③	closed caption button	Click to display closed caption settings.
④	bitrate selection button	Click to display bitrate options.
⑤	full screen button	Click to display the video in full-screen mode.

START SCREEN

The start screen shows before a video begins playback.





Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	title	Asset title.
2	play button	Click to start playing video.

PAUSE SCREEN

The pause screen that displays when the video is paused during playback.

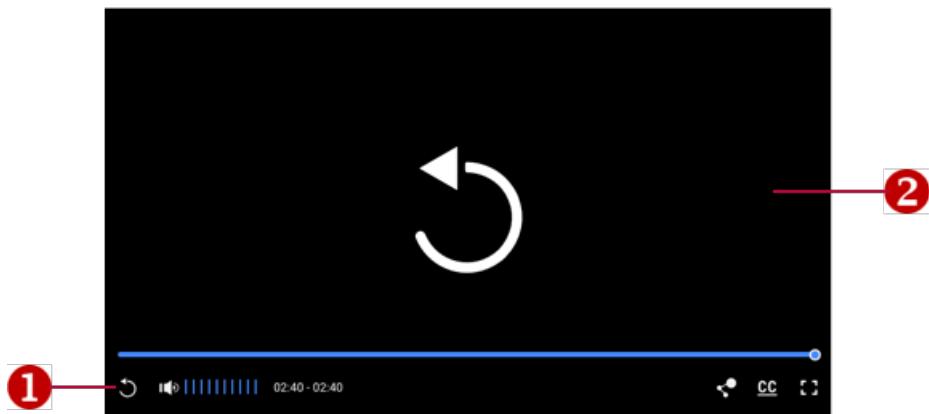


#	Name	Description
1	play	Click to resume playing from the current playhead position.

END SCREEN

The end screen shows after a video completes playback.



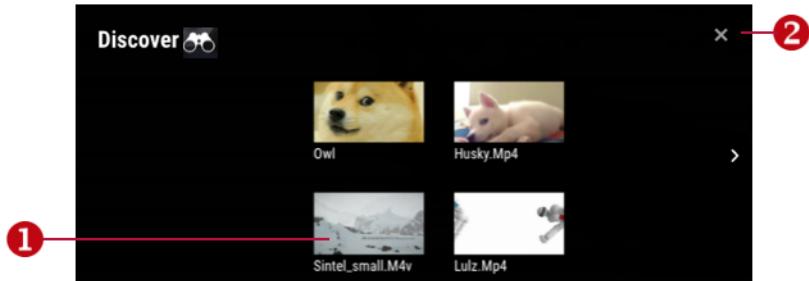


Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	replay button	Click to replay from the beginning.
②	screen to show at end	Screen to show at the end. In this example, the screen is blank.

DISCOVERY SCREEN

The discovery screen shows related or relevant content.



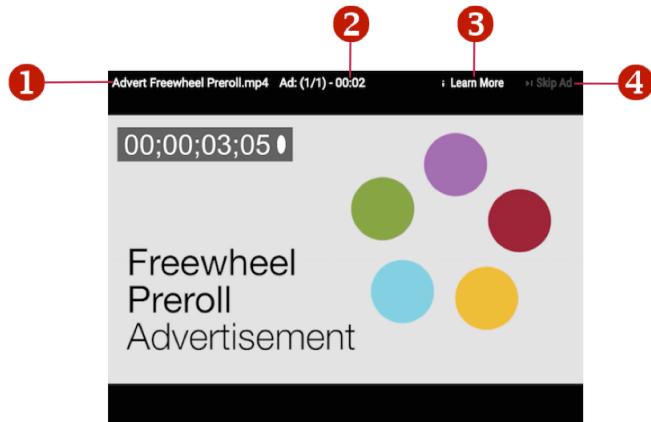
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	related videos	Scroll and click a related video.
②	close	Click to close the discovery list.

AD SCREEN

The ad screen shows ads in the player.



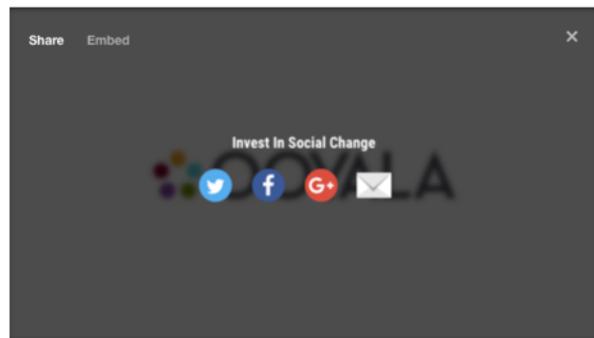


Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	ad title	Title of the current ad.
2	duration	Length of the ad, in seconds.
3	learn more	Click to learn more about this ad.
4	skip ad	Click to skip the current ad.

SOCIAL SHARING SCREEN

The social sharing screen provides options to share this video with others.



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

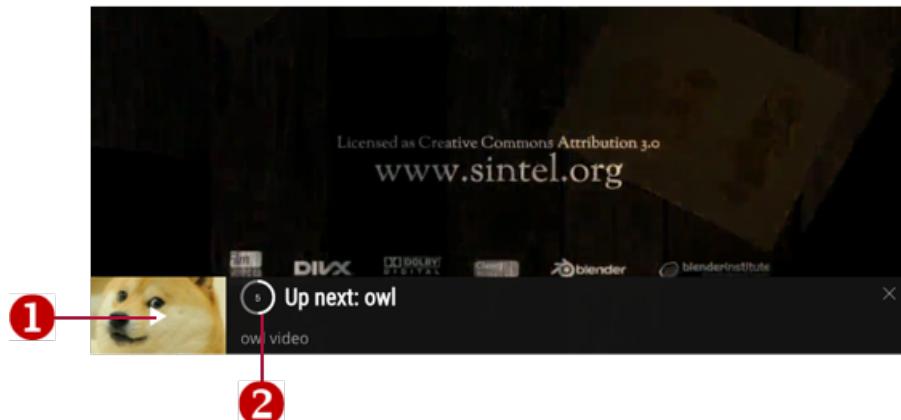
#	Name	Description
	share on Twitter	Click to share this video on Twitter.
	share on Facebook	Click to share this video on Facebook.
	share on Google+	Click to share this video on Google+.



#	Name	Description
▼	share via email	Click to share this video via email.

UP NEXT SCREEN

The up next screen shows the next video to play.



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	up next	Display the next upcoming video.
②	countdown	Time remaining before the next video plays.

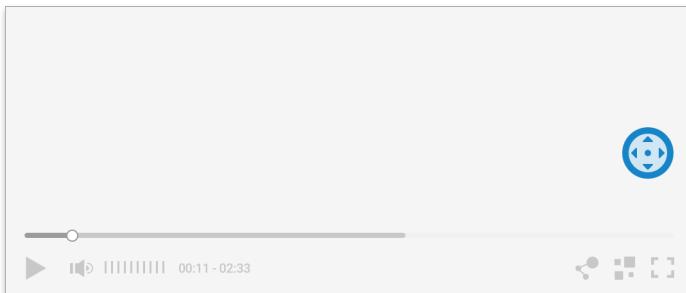
LIVE DVR

With live DVR, the LIVE indicator in the control bar means live streaming, and a duration (greater than zero) indicates that DVR is available. With live DVR, you can drag the slider to the left to go backwards in time, and drag it to the right to go forwards. Dragging it all the way to the right catches you up to the present time. Live DVR is used with HLS streams only.



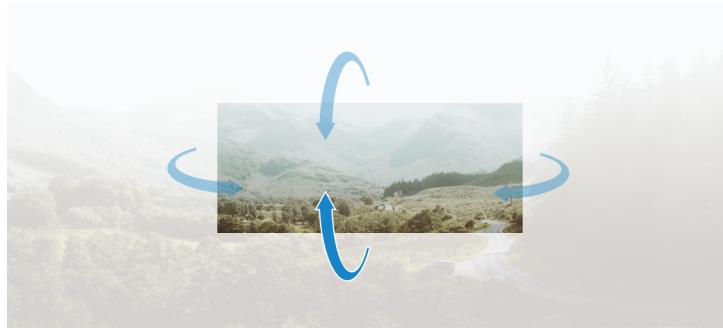
PLAYING VR 360 VIDEOS

The Ooyala Player automatically detects VR 360 streams and displays the applicable playback controls to navigate VR 360 videos. The following example shows the VR 360 icon on screen.

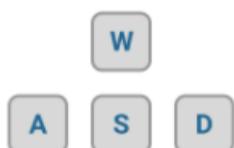


Changing the Camera Angle

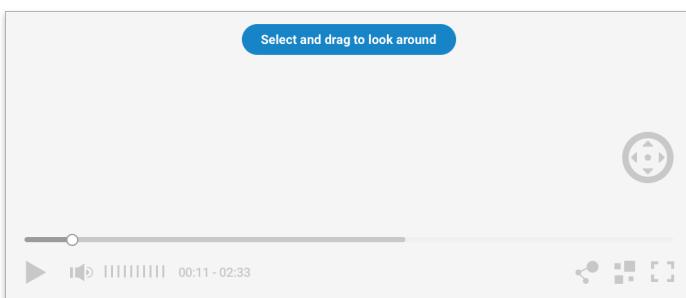
Viewers have several ways in which to change the camera angle.



Keyboard. Viewers can use the keyboard controls (WASD) to change the camera angle.



Mouse. Viewers can select and drag with a mouse to change the camera angle.



Mobile Devices. Viewers change the camera angle by touching and swiping.





Switching Between Monoscopic and Stereoscopic Modes

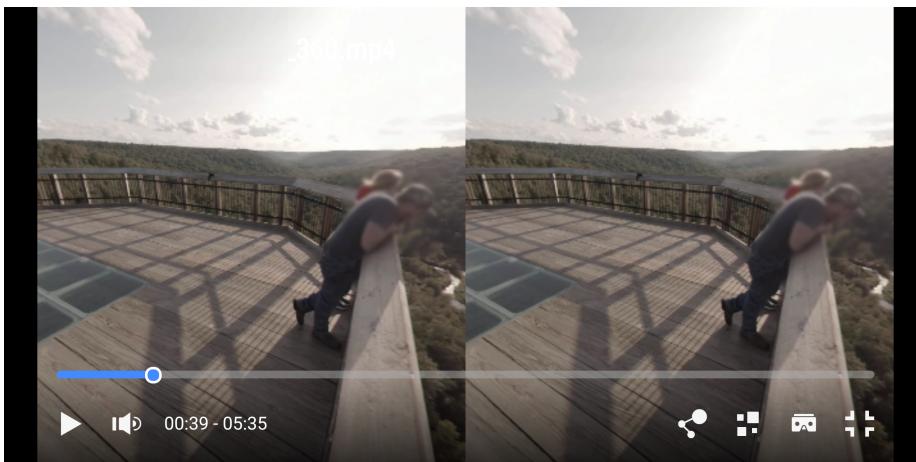
Use the following icon to switch between monoscopic and stereoscopic modes.



Here is an example in monoscopic mode.



Here is an example in stereoscopic mode.



THUMBNAIL-BASED SEEKING

Player V4 displays thumbnails to help you search through a video.

- For desktop and mobile devices, if you drag the scrubber bar, the player pops up a carousel of thumbnails. The player displays as many as will fit on the screen.





- For desktop devices (browsers in mobile devices do not support this behavior), if you hover your mouse over the scrubber bar, you will see the nearest thumbnail image associated with that location in the video.



This functionality is available for Ooyala-encoded assets that have associated thumbnails. For more information, see [Auto-generated Preview Images \(Thumbnails\)](#).

BUTTONS IN THE PLAYER

Image	Name	Description
▶	play button	Play the video.
⏸	pause button	Pause playback.
⟳	replay	Replay the current video.
🔊	volume control (mobile)	Control the volume level.
🔊	volume control (web)	Control the volume level.
🔇	mute	Mute the audio.
[]	full screen	Change display to full-screen mode.
[]	standard screen	Change display to standard screen mode.
🔗	share	Share this video with others.



Image	Name	Description
	bitrate selection	Select a bitrate.
	discovery	Discover related videos
	left arrow	Scroll left through the available videos.
	right arrow	Scroll right through the available videos.
	closed captions	Display closed captions for the video.
	close	Close the player.



SUPPORTED BROWSERS AND OPERATING SYSTEMS FOR PLAYER V4

Player V4 features a new skin that is implemented in three products:

- **Ooyala Player V4 for HTML5** is a Web-based player that runs in popular browsers across desktop and mobile devices.
- **Ooyala Player Skin for iOS** enhances the Ooyala (video player) Mobile SDK for iOS by integrating the Player V4 skin in iOS native apps.
- **Ooyala Player Skin for Android** enhances the Ooyala (video player) Mobile SDK for Android by integrating the Player V4 skin in Android native apps.

Player V4 and the new player skin are currently supported in the environments that follow.

WINDOWS SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers running in the Windows environment:

- Chrome 47+ on Windows 7+
- Internet Explorer 11+ on Windows 7+ (or Windows 8.1+ if MPEG-DASH or HLS stream packaging is used)
- Microsoft Edge on Windows 10
- Firefox v48+ on Windows 7+

Note: There are currently some known issues with touch handling on Windows tablets running IE/Edge, which require resolution from Microsoft.

MAC OS SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers running in the Mac OS 10.9+ environment:

- Chrome v47+
- Firefox v48+
- Safari v8+

MOBILE WEB SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers for mobile web:

- Safari on iOS v9+
- Chrome on Android v4.4, v5.x+

IOS APP SUPPORT

Ooyala Player Skin for iOS is supported on the iOS v9+ environment.

ANDROID APP SUPPORT

Ooyala Player Skin for Android is supported on devices running Android v4.4+.

Note: As of 2017-10-01, the supported Android environments start with Android version 4.4. Issues on Android versions previous to this version will be reviewed on a case-by-case basis.



SYSTEM REQUIREMENTS BY PLATFORM FOR PLAYER V4

Player V4 provides consumers with the highest quality stream possible based on available bandwidth, CPU, device, and throughput. If you are seeing less than ideal quality, make sure that your environment meets these minimum requirements.

Note: These configurations apply only to non-DRM (clear) streams only. For supported platforms for DRM-protected content, see [Content Protection Options By Device](#) instead.

Note: These configurations undergo regular testing with each release. Other environments (such as Linux or Chrome beta channels) might work, but are not actively tested.

WINDOWS

Requirement	Description
Processor	Intel Pentium 4 or equivalent with 1 GHz processor
Operating System	<ul style="list-style-type: none">Windows 10 with all available updatesWindows 8 with all available updatesWindows 7 with all available updates
RAM	512 MB minimum (1 GB recommended)
Video Card	32 MB minimum (128 MB recommended)
Browser	<ul style="list-style-type: none">Chrome v47+Firefox v43+Internet Explorer v11+Microsoft Edge (Windows 10)
Flash	<ul style="list-style-type: none">For Real Time Messaging Protocol (RTMP) playback: 10.0 or higher (10.1 or later recommended)For HTTP Dynamic Streaming (HDS) playback: 10.1.0 or higher
Screen Resolution	800 x 600 minimum (1280 x 1024 or greater recommended)
Internet Connection	Broadband/High Speed (DSL, Cable, T1, etc.). Dial-up connections are insufficient for video playback.

MAC OS X

Requirement	Description
Processor	Intel Core2 Duo 1.83GHz or faster processor
Operating System	Mac OS X 10.5 or greater with Intel Processors of 1GHz or better. No PowerPC support.
RAM	512 MB minimum (1 GB recommended)
Video Card	32 MB minimum (128 MB recommended)



Requirement	Description
Browser	<ul style="list-style-type: none"> • Chrome v47+ • Firefox v43+ • Safari v8+
Flash	10.0 or higher (10.1 or later recommended)
Screen Resolution	800 x 600 minimum (1280 x 1024 or greater recommended)
Internet Connection	Broadband/High Speed (DSL, Cable, T1, etc.). Dial-up connections are insufficient for video playback.

APPLE IOS

Requirement	Description
Operating System	iOS v8+
Browser	Safari
Internet Connection	WiFi or 3G

ANDROID

Requirement	Description
Operating System	Android v4.1+
Browser	Chrome for Android
Internet Connection	WiFi or 3G

Note: Android device testing is limited to popular Android devices from Samsung, Google, and HTC.



FUNCTIONAL SUPPORT ACROSS ENVIRONMENTS FOR PLAYER V4

This topic summarizes Player V4 functional support across environments.

AD SUPPORT

VAST 2.0 and 3.0

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

VPAID 2.0

- Ooyala Player V4 for HTML5

Ooyala Pulse

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS

Google IMA (including Google DFP and VPAID 1.0 Support)

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

FreeWheel

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

LiveRail

- Ooyala Player V4 for HTML5

ANALYTICS SUPPORT

Ooyala IQ Analytics

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

comScore Analytics

- Ooyala Player V4 for HTML5

Adobe Analytics (Omniture)

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android



BRANDING SUPPORT (VISUAL ELEMENTS, SUCH AS COLORS, IMAGES, AND BUTTONS)

Skin Branding

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

CSS Branding

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

SECURITY SUPPORT

Ooyala Player Token

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Secure Playback Token

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Encrypted Streams

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Syndication Restrictions

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

CLOSE CAPTIONS SUPPORT

Closed Captions - VOD

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Closed Captions - Live

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

PLAYBACK FEATURES SUPPORT

Adaptive Bitrate Support



- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Full Screen Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Social Sharing Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Responsive Design Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Content-aware Controls Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Live DVR Support (HLS Streams)

- Player V4 (requires the Bitmovin video plugin)
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

DISCOVERY SUPPORT

Discovery Interface Customization

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Discovery Recommendations Engine Configuration

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

DEVELOPER SUPPORT

Player JavaScript API

- Ooyala Player V4 for HTML5

Backlot REST API

- Ooyala Player V4 for HTML5

Authorization REST API

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS



- Ooyala Mobile SDK for Android

LIVE STREAMING SUPPORT

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

LOCALIZATION SUPPORT

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android



INTRODUCING PLAYER V4

Ooyala Player V4 is a next-generation video player that provides an engaging, personalized, responsive, content aware, and robust video playback experience that is consistent across devices and platforms.

PLAYER V4 FAQ

Topics:

- [*Player V4 Overview*](#)
- [*Supported Features*](#)
- [*Getting Started with Player V4*](#)
- Migrating from Player V3 to Player V4 (see [*Player Migration FAQ*](#) on page 66)
- [*Customizing Player V4*](#)
- [*Using Video Plugins to Decode and Render Video Streams*](#)
- [*Using Ad Plugins to Play Ads*](#)
- [*Troubleshooting Player V4*](#)

PLAYER V4 OVERVIEW

Q: What is Player V4?

A: Ooyala's next generation video player for HTML5 is designed to provide an engaging, personalized, content aware, and robust video playback experience that is consistent across devices and platforms. See [*Player V4*](#) on page 6 for a full list of features.

Q: How does Player V4 differ from earlier Ooyala players?

A: Player V4 has a responsive and customizable UI that you can use out-of-the-box or customize to your branding needs. It's very simple to embed in your web site, with the user interface configurable from our Backlot CMS. Its modular architecture allows a wide range customizations, simply by configuring the player in the Backlot UI, by modifying a JSON configuration file, or even updating player CSS to make branding and UI customizations perfect for your brand.

While the Ooyala Player V3 was a Flash-first player by default, Player V4 is built and optimized for HTML5 playback. The Player V4 has built-in Flash-failover as needed using the bit_wrapper.js plugin (recommended), or OSMF plugin where Adobe Access DRM is still needed. See [*Flash Video Rendering*](#) on page 94 for details on Flash video rendering.

Player V4 is a modular player consisting of plugins, which means that you can add or remove certain functionality depending on which plugins you load on your HTML page. The modular architecture of Player V4 allows for a much smaller embed footprint, faster load times, and many more customization options.

Player V4 features a new [*Ad Manager Framework*](#) on page 186 with robust ad support, and an Analytics Framework that allows you to integrate third-party analytics providers with the player. See [*Player V4*](#) on page 6 for a full list of features.

Q: How does Player V4 look to users?

A: Player V4 has a simple and accessible user interface that is optimized to run in many environments. See [*User Interface for the Player V4 Skin*](#) on page 11 for descriptions of the Player V4 UI components.



SUPPORTED FEATURES

Q: What features are supported in this version of Player V4?

A: See [Ooyala Player V4 Release Notes](#) on page 235 for a list of features released in the latest version of Player V4.

Q: Does Player V4 integrate with social networks?

Q: Yes, Player V4 supports integrations with Facebook, Twitter, and Google Plus.

Q: Can Player V4 controls be translated (localized)?

A: Yes. We offer localization files for English, Spanish, Japanese, and Simplified Chinese, but you can also add your own localization files.

Q: What closed caption languages does Player V4 support?

A: You can upload your own closed caption languages. Supported languages are listed at [Supported Closed Captions](#).

Q: Is QoS analytics supported in Player V4?

A: Yes. For details, see [YOUBORA Analytics Plugin](#) on page 207 and [Conviva Analytics Plugin](#) on page 197.

Is a comScore integration supported for Player V4?

A: Yes, Player V4 supports an out-of-the-box integration with comScore. Please contact your Ooyala account manager to use the Player V4 comScore integration. For details, see [comScore Analytics Plugin](#) on page 196

GETTING STARTED WITH PLAYER V4

Q: Once I am up set up with Player V4, do I need to do continuous developer work to keep up with Ooyala monthly releases?

A: No. If you are using the Ooyala Player V4 Standard embed code, or the resource paths for Production or Production/Latest, then your embed players will automatically get updates. Ooyala hosts two release paths (production and latest, as described in [Ooyala-hosted Player V4 Resources](#) on page 77) so that you are automatically upgraded to the latest version on that path.

If you are self-hosting ([Hosting Player V4 Resources](#) on page 81), all you need to do is pull/clone the latest versions of the plugins and player files.

Q: Can I mix and match files from different release versions (for example, skin.json from 4.11.13 and ad_manager_vast.js from 4.10.6)?

A: We strongly recommend using all files from the same release version.

Q: What does Player V4 have to do with the video mobile SDKs for iOS and Android?

A: Player V4 and the mobile SDKs for iOS and Android share the same look and feel (skin). When we refer to the Player V4 skin, we are referring typically to the skin.json file that can be used to apply player UX to the HTML5 Player V4 and the mobile SDK players. However, there are some differences in skin configurations that are supported by platform.

Q: Are there any special cases where Player V4 will be forced to load Flash instead of HTML5? In case the player is handling VPAID ads what would be the behavior? Will a Flash player be loaded at the beginning or only when serving the Ad? Would the actual content still be played in HTML5 player?

A: The core player will **always** be HTML5-based in Player V4. If the selected encoding is not supported in HTML5 on a certain browser and if Flash is supported, the main video will render in Flash. The rest of



the player including the UI will remain HTML5-based. For VPAID 1.0, the VPAID ad will render in a Flash-based element, but the rest of the player will remain HTML5-based, including the main video, assuming the main video wasn't already playing in a Flash renderer. See [Flash Video Rendering](#) on page 94 for details on Flash video rendering.

Q: Which browsers and browser versions work with Player V4?

A: See [Supported Browsers and Operating Systems for Player V4](#) on page 22 for a full list of browsers to use with Player V4.

Q: How do I embed Player V4 into my website?

A: See [Embedding Player V4 on a Web Page](#) on page 69 for detailed developer documentation on how to embed Player V4 into your website. See [Configuring Player Embed Settings in Backlot](#) for documentation on how to copy the Player V4 embed code from Backlot.

CUSTOMIZING PLAYER V4

Q: What can I configure in the new player?

A: Player V4 is highly customizable. You can:

- Enable Discovery content recommendation.
- Configure the player look and feel (skin) using a JSON config file, by specifying player settings directly in HTML pages, or by modifying the Player CSS and resources (see [Customizing the Player V4 Appearance](#) on page 116).
- Use the message bus to make additional event-driven customizations.
- Localize player controls and error messages.
- Configure HTML5 or Flash video rendering. Note that all of the plugins, ad logic, UI, Player APIs, etc. are always in HTML5. The only Flash component is video decoding and rendering for HDS and DASH. See [Flash Video Rendering](#) on page 94 for details on Flash video rendering.

Q: Can I customize the Player V4 UI with Backlot

A: Yes - see [Configuring a Player in Backlot](#). You can also make customizations to the player by customizing the player CSS or the skin.json config file (see [Configuring a Player in Backlot](#)).

Q: How can I add another social sharing option to the player (beyond the default Facebook, Twitter, Google Plus)?

A: You can add additional social sharing options by altering the skin code directly. The code associated with social sharing is located in the html5-skin repository at `html5-skin/js/views/shareScreen.js`.

Q: How can I add a watermark image clickthrough URL?

A: You can specify a watermark image clickthrough URL in Backlot (PUBLISH > Player Branding > Branding), or in skin.json (specifically with controlBar.watermark.clickUrl). See [Customizing the Player V4 Skin with skin.json](#) on page 120 for details on how to modify skin.json.

Q: How do I configure autoplay with embedded parameters?

A: See the [player embedded parameter documentation](#) for details on how to use the `autoplay` parameter. Set `autoplay` to true to enable autoplay behavior.

Q: How can I remove items from the control bar?

A: You can delete unneeded items from the buttons array in the skin.json config file.

Q: What are best practices for setting my video container size?

A: If you know your video aspect ratio, set the video container size to a size that corresponds to the aspect ratio of the video.

Q: If I customize the player, will Ooyala Technical Support help me debug my code?



A: Ooyala will support any pre-defined and valid modifications to the skin.json file. If you fork the skin-config repository and make non-valid modifications, or if you fork the html5-skin repository and make changes, we will not offer technical support for these changes.

USING VIDEO PLUGINS TO DECODE AND RENDER VIDEO STREAMS

See [Loading Video Plugins](#) on page 89 for details.

Q: Does "encodingPriority": ["hls", "mp4", "dash"] mean that first the player will try to load an HLS stream, then if it is not available, it will try to load an MP4 stream, then if it is not available, it will try to load a dash stream, and so on?

A: The video controller will only attempt to select a single stream. Encoding priority is a way to set the stream preference order. Internally the player will look at what stream type is available and what stream type is playable by the video plugins you've loaded on the page and will try to play the first available stream.

Q: Can I specify the starting bitrate for adaptive bitrate (ABR)?

A: Yes, see [Programming Bitrates and Buffering for Player V4 Using JavaScript](#) on page 225 for details on how to do this.

Q: Should I use DASH?

A: For many applications, you can use HLS for most common scenarios. However, there are situations where using DASH can provide better performance and/or additional capabilities such in conjunction with Widevine Modular DRM. If DASH encoding is added to your processing profile, you might have to re-process your assets in order to leverage DASH encoding for your existing assets. Please note that DASH migration is not mandatory.

Q: Why do some of the video plugins (Bitmovin and OSMF Flash) still have the ability to render video in Flash? I thought the industry was moving away from Flash?

A: Migrating assets to be HTML5-compatable stream formats can be expensive and time-consuming. We are continuing to support Flash playback for customers who are still in the process of migrating their asset libraries.

USING AD PLUGINS TO PLAY ADS

See [Ads in Player V4](#) on page 142 for details.

Q: Do I need to ask Ooyala Technical Support to enable anything for Player V4 ad plugins to work?

A: If you are creating ad sets in Backlot, you need to ask Ooyala Technical Support to enable ad managers for your account, but other than that, all you need to do is load the plugins on your page.

How do the ad plugins interact with the player?

A: Ad plugins interact with the V4 player through the [Ad Manager Framework](#) on page 186, a unified framework that handles basic ad functionality, which allows ad managers to plug into Ooyala with less effort.

Q: How is integrating with ad managers different in Player V4 compared to Player V3?

A: Player V4 is built in a modular way, where ad functionality is implemented using different plugins. See [Player V4 Plugins](#) on page 69 for the full list of plugins available. With the new Ad Manager Framework for Player V4, configuration parameters and some page level settings are different from Player V3. If you don't have any page level overrides to Backlot settings, your ads will have full backwards compatibility. See [Migrating to Player V4](#) on page 57 for full details on what configuration parameters and page level settings have changed.

Q: Does Player V4 integrate with Pulse?



A: Yes. There is a Pulse plugin for Player V4 (HTML5). Pulse also currently integrates with the iOS and Android video SDKs. See [Ooyala Pulse Ad Plugin](#) on page 160 for details.

Q: Does Player V4 support Ooyala Ads / My Ads (Backlot content served as video ads)?

A: Yes, but only as pre-rolls, and only with MP4-type assets currently. For more extensive options, please consider our VAST or Ooyala Pulse ad plugins.

TROUBLESHOOTING PLAYER V4

Q: I'm seeing the player frame expand to an extremely large size. What's happening?

A: Player V4 has new responsive design features. By default, as long as the container's width is bounded, the height will dynamically adjust according to the video's aspect ratio. If you already know the aspect ratio of the video, you can also try to override the skin.json setting to fix the skin's aspect ratio. Change the value of aspectRatio (shown below) in skin.json to your video's aspect ratio. You can calculate the aspect ratio using height / width * 100. For a 4:3 video, $3 / 4 * 100 = 75$, and for a 16:9 video, $9 / 16 * 100 = 56.25$.

```
"responsive": {  
    "breakpoints": {  
        ...  
    },  
    "aspectRatio": "auto"  
}
```

Q: I copied from your example on [GitHub](#) but I'm getting 404 errors related to the page referencing local files. Why is this happening?

A: Our example on GitHub uses the default skin.json file (see [Ooyala-hosted Player V4 Resources](#) on page 77), which references 3 files: config/es.json, config/en.json, and config/zh.json, which are our default localization files for the player. If you want to use our example out of the box, you'll need to get those files to resolve. You can either download and modify skin.json to point to absolute file paths, or copy the three language files into your website's file structure (such as <http://your-site.com/testpage/config/en.json>).

Q: I'm seeing poor video quality when the player starts up. How can I improve the video quality?

A: You can use the initialBitrate parameter to set the initial minimum bitrate level (immediately after video playback) and to sustain that level for a specific period of time. For details, see [Page-level Parameters for Player V4](#) on page 99).

Q: Why are the player buttons now appearing as letters?

A: If you are customizing the player and are now seeing the play button appear as an 'h', then you likely have made reference to the new location of the html5-skin.css file but not the necessary assets directory (particularly the font files within the assets directory). By hosting the assets directory in the same location as your html5-skin.css file, you can resolve the missing icons and even update the icons file with new icons of your own.

USER INTERFACE FOR THE PLAYER V4 SKIN

This topic describes the visual components of the new Ooyala player.

KEY COMPONENTS

Here are the key components of the player.





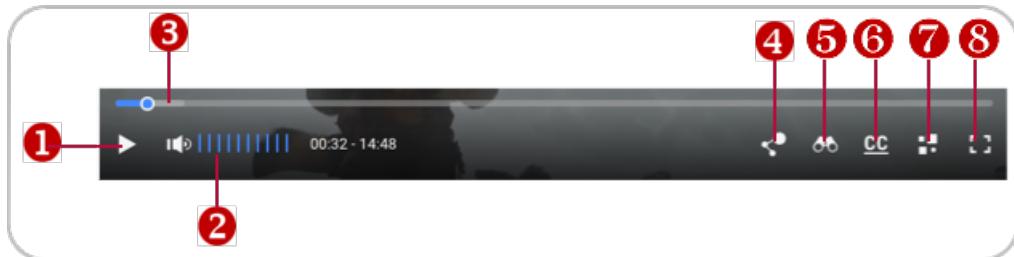
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	display area	Where the video is displayed.
②	control bar	Enables viewers to manually configure settings and move forward or backward through a video.

CONTROL BAR

The control bar (also known as a scrubber or slider) enables viewers to manually adjust settings and move forward or backward through a video.

Here are visual elements of the control bar in the READY or PAUSED state:



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	play button	Click to start video or audio playback.
②	volume control	<ul style="list-style-type: none"> Drag the slider to adjust the volume. Click the volume control to mute the sound.



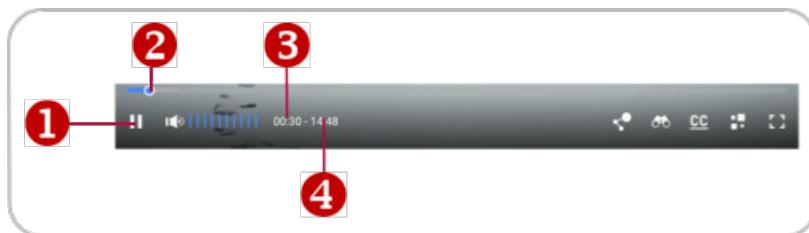
- (Safari only) Click the following control to unmute the sound:

SELECT TO UNMUTE



#	Name	Description
③	buffered content	Shows how much of the video has been retrieved in the buffer.
④	share button	Click to share this video with others via email, embed URL, or social media.
⑤	discover button	Click to discover related videos.
⑥	closed caption button	Click to select closed captions.
⑦	bitrate selection button	Click to display bitrate options.
⑧	full screen button	Click to display the video in fullscreen mode.

Here are visual elements of the control bar in the PLAYING state:



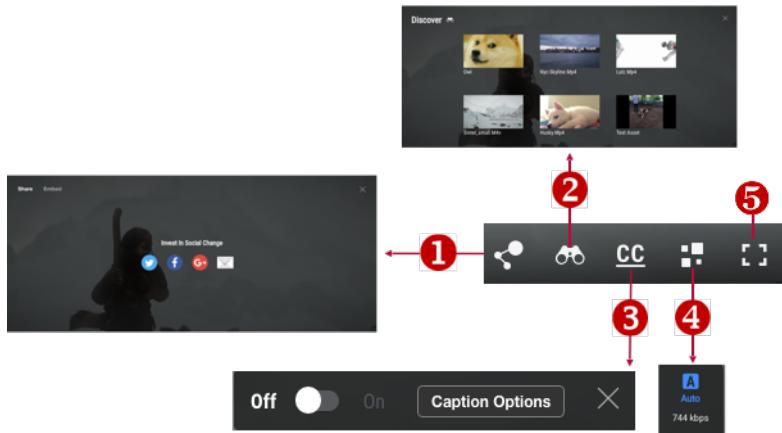
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	pause button	Click to pause playback.
②	scrubber (slider)	Drag and slide to manually move forward or backward through a video. Note: Scrubber behavior depends on the playback context: <ul style="list-style-type: none"> If a live stream is DVR, then the initial scrubber position starts at the end (right-most part) of the control bar (current time), and the user can seek backwards all the way to the earliest point available (left-most part). If a live stream is <i>not</i> DVR, then the scrubber position starts at the left-most part of the control bar and has a duration of zero (the user cannot move the scrubber). For VOD, the initial scrubber position starts at the left-most portion of the control bar, and the user can seek forwards (and subsequently backwards).
③	play head position	Current position during playback (minutes:seconds).
④	content duration	Total length of the video (minutes:seconds).

BUTTONS

The player displays buttons that can provide options to share the video on social networks, discover related videos, set closed captions and bitrates, and view the video in full-screen mode.



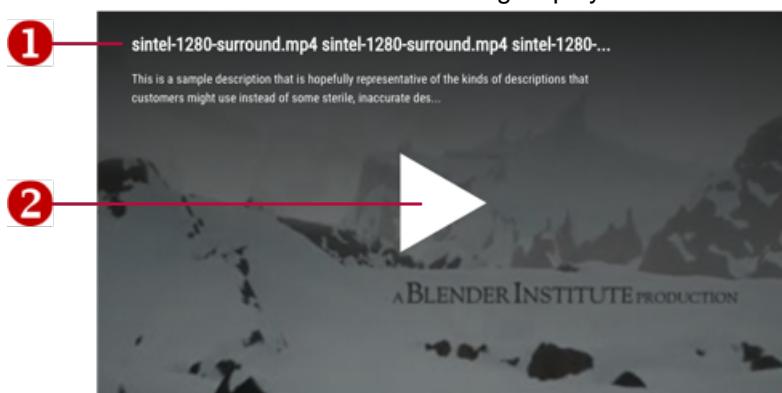


Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	share button	Click to share this video with others via email, embed URL, or social media.
2	discover button	Click to discover related videos.
3	closed caption button	Click to display closed caption settings.
4	bitrate selection button	Click to display bitrate options.
5	full screen button	Click to display the video in full-screen mode.

START SCREEN

The start screen shows before a video begins playback.



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	title	Asset title.



#	Name	Description
2	play button	Click to start playing video.

PAUSE SCREEN

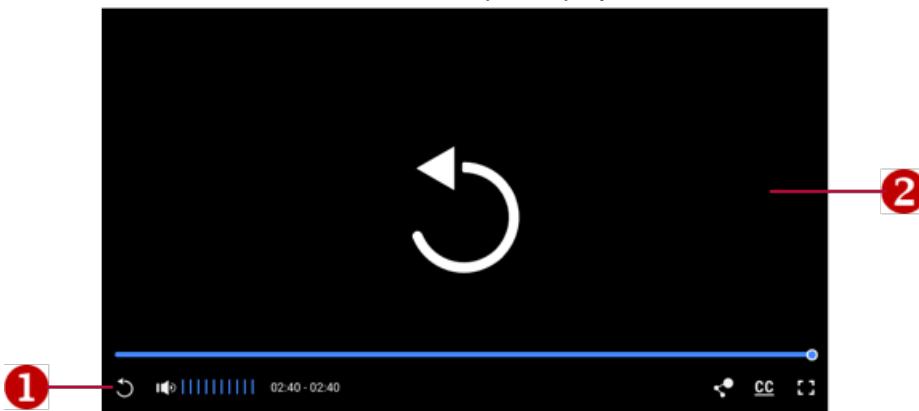
The pause screen that displays when the video is paused during playback.



#	Name	Description
1	play	Click to resume playing from the current playhead position.

END SCREEN

The end screen shows after a video completes playback.



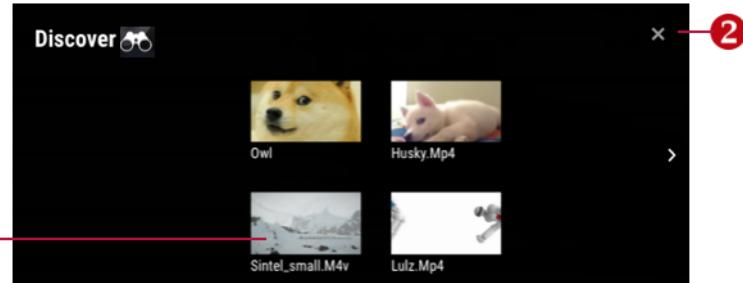
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
1	replay button	Click to replay from the beginning.
2	screen to show at end	Screen to show at the end. In this example, the screen is blank.

DISCOVERY SCREEN

The discovery screen shows related or relevant content.



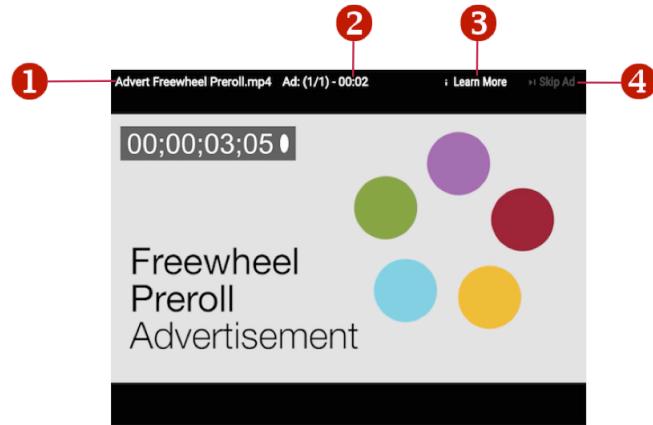


Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	related videos	Scroll and click a related video.
②	close	Click to close the discovery list.

AD SCREEN

The ad screen shows ads in the player.



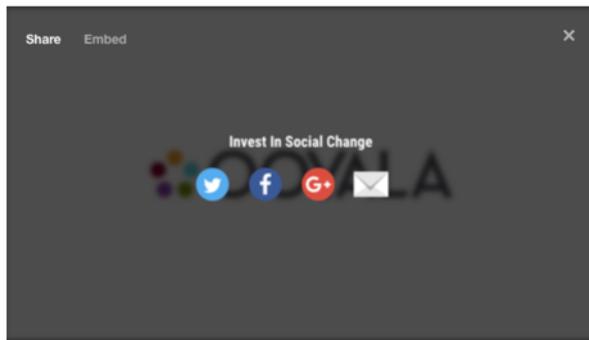
Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
①	ad title	Title of the current ad.
②	duration	Length of the ad, in seconds.
③	learn more	Click to learn more about this ad.
④	skip ad	Click to skip the current ad.

SOCIAL SHARING SCREEN

The social sharing screen provides options to share this video with others.



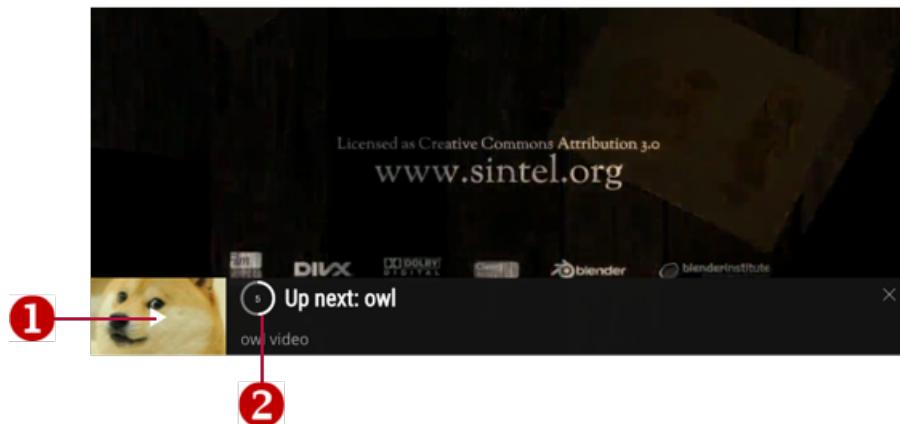


Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
	share on Twitter	Click to share this video on Twitter.
	share on Facebook	Click to share this video on Facebook.
	share on Google+	Click to share this video on Google+.
	share via email	Click to share this video via email.

UP NEXT SCREEN

The up next screen shows the next video to play.



Note: The above example screen uses Ooyala default desktop web settings. Your screen might look different.

#	Name	Description
	up next	Display the next upcoming video.
	countdown	Time remaining before the next video plays.



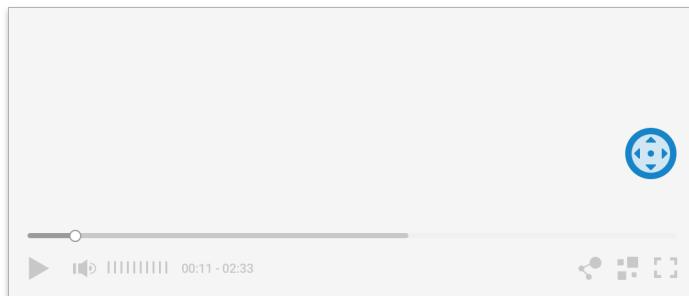
LIVE DVR

With live DVR, the LIVE indicator in the control bar means live streaming, and a duration (greater than zero) indicates that DVR is available. With live DVR, you can drag the slider to the left to go backwards in time, and drag it to the right to go forwards. Dragging it all the way to the right catches you up to the present time. Live DVR is used with HLS streams only.



PLAYING VR 360 VIDEOS

The Ooyala Player automatically detects VR 360 streams and displays the applicable playback controls to navigate VR 360 videos. The following example shows the VR 360 icon on screen.



Changing the Camera Angle

Viewers have several ways in which to change the camera angle.

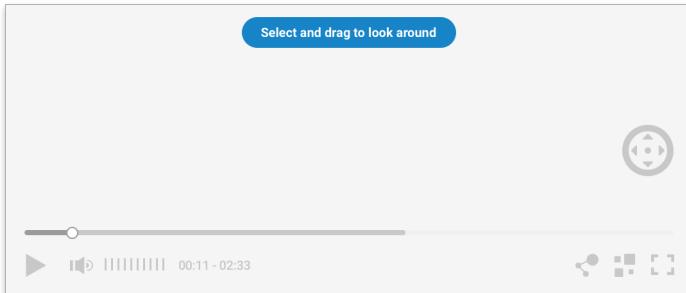


Keyboard. Viewers can use the keyboard controls (WASD) to change the camera angle.





Mouse. Viewers can select and drag with a mouse to change the camera angle.



Mobile Devices. Viewers change the camera angle by touching and swiping.



Switching Between Monoscopic and Stereoscopic Modes

Use the following icon to switch between monoscopic and stereoscopic modes.

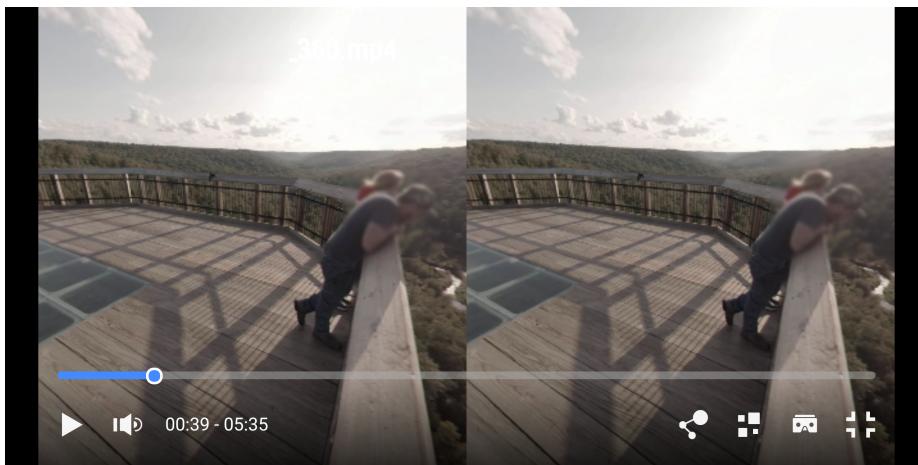


Here is an example in monoscopic mode.



Here is an example in stereoscopic mode.

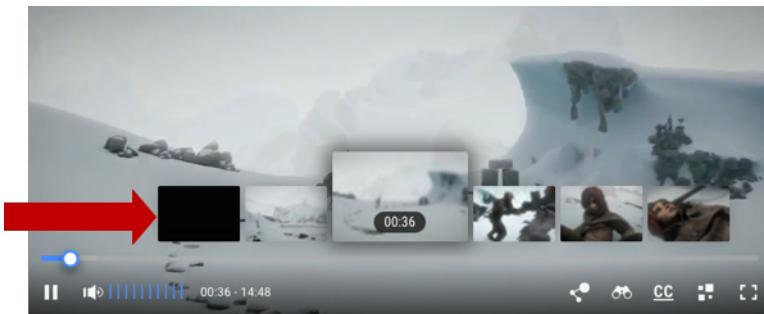




THUMBNAIL-BASED SEEKING

Player V4 displays thumbnails to help you search through a video.

- For desktop and mobile devices, if you drag the scrubber bar, the player pops up a carousel of thumbnails. The player displays as many as will fit on the screen.



- For desktop devices (browsers in mobile devices do not support this behavior), if you hover your mouse over the scrubber bar, you will see the nearest thumbnail image associated with that location in the video.



This functionality is available for Ooyala-encoded assets that have associated thumbnails. For more information, see [Auto-generated Preview Images \(Thumbnails\)](#).

BUTTONS IN THE PLAYER

Image	Name	Description
	play button	Play the video.



Image	Name	Description
	pause button	Pause playback.
	replay	Replay the current video.
	volume control (mobile)	Control the volume level.
	volume control (web)	Control the volume level.
	mute	Mute the audio.
	full screen	Change display to full-screen mode.
	standard screen	Change display to standard screen mode.
	share	Share this video with others.
	bitrate selection	Select a bitrate.
	discovery	Discover related videos
	left arrow	Scroll left through the available videos.
	right arrow	Scroll right through the available videos.
	closed captions	Display closed captions for the video.
	close	Close the player.

SUPPORTED BROWSERS AND OPERATING SYSTEMS FOR PLAYER V4

Player V4 features a new skin that is implemented in three products:

- **Ooyala Player V4 for HTML5** is a Web-based player that runs in popular browsers across desktop and mobile devices.
- **Ooyala Player Skin for iOS** enhances the Ooyala (video player) Mobile SDK for iOS by integrating the Player V4 skin in iOS native apps.
- **Ooyala Player Skin for Android** enhances the Ooyala (video player) Mobile SDK for Android by integrating the Player V4 skin in Android native apps.

Player V4 and the new player skin are currently supported in the environments that follow.



WINDOWS SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers running in the Windows environment:

- Chrome 47+ on Windows 7+
- Internet Explorer 11+ on Windows 7+ (or Windows 8.1+ if MPEG-DASH or HLS stream packaging is used)
- Microsoft Edge on Windows 10
- Firefox v48+ on Windows 7+

Note: There are currently some known issues with touch handling on Windows tablets running IE/Edge, which require resolution from Microsoft.

MAC OS SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers running in the Mac OS 10.9+ environment:

- Chrome v47+
- Firefox v48+
- Safari v8+

MOBILE WEB SUPPORT

Ooyala Player V4 for HTML5 is supported on the following browsers for mobile web:

- Safari on iOS v9+
- Chrome on Android v4.4, v5.x+

IOS APP SUPPORT

Ooyala Player Skin for iOS is supported on the iOS v9+ environment.

ANDROID APP SUPPORT

Ooyala Player Skin for Android is supported on devices running Android v4.4+.

Note: As of 2017-10-01, the supported Android environments start with Android version 4.4. Issues on Android versions previous to this version will be reviewed on a case-by-case basis.

SYSTEM REQUIREMENTS BY PLATFORM FOR PLAYER V4

Player V4 provides consumers with the highest quality stream possible based on available bandwidth, CPU, device, and throughput. If you are seeing less than ideal quality, make sure that your environment meets these minimum requirements.

Note: These configurations apply only to non-DRM (clear) streams only. For supported platforms for DRM-protected content, see [Content Protection Options By Device](#) instead.

Note: These configurations undergo regular testing with each release. Other environments (such as Linux or Chrome beta channels) might work, but are not actively tested.



WINDOWS

Requirement	Description
Processor	Intel Pentium 4 or equivalent with 1 GHz processor
Operating System	<ul style="list-style-type: none"> Windows 10 with all available updates Windows 8 with all available updates Windows 7 with all available updates
RAM	512 MB minimum (1 GB recommended)
Video Card	32 MB minimum (128 MB recommended)
Browser	<ul style="list-style-type: none"> Chrome v47+ Firefox v43+ Internet Explorer v11+ Microsoft Edge (Windows 10)
Flash	<ul style="list-style-type: none"> For Real Time Messaging Protocol (RTMP) playback: 10.0 or higher (10.1 or later recommended) For HTTP Dynamic Streaming (HDS) playback: 10.1.0 or higher
Screen Resolution	800 x 600 minimum (1280 x 1024 or greater recommended)
Internet Connection	Broadband/High Speed (DSL, Cable, T1, etc.). Dial-up connections are insufficient for video playback.

MAC OS X

Requirement	Description
Processor	Intel Core2 Duo 1.83GHz or faster processor
Operating System	Mac OS X 10.5 or greater with Intel Processors of 1GHz or better. No PowerPC support.
RAM	512 MB minimum (1 GB recommended)
Video Card	32 MB minimum (128 MB recommended)
Browser	<ul style="list-style-type: none"> Chrome v47+ Firefox v43+ Safari v8+
Flash	10.0 or higher (10.1 or later recommended)
Screen Resolution	800 x 600 minimum (1280 x 1024 or greater recommended)
Internet Connection	Broadband/High Speed (DSL, Cable, T1, etc.). Dial-up connections are insufficient for video playback.



APPLE IOS

Requirement	Description
Operating System	iOS v8+
Browser	Safari
Internet Connection	WiFi or 3G

ANDROID

Requirement	Description
Operating System	Android v4.1+
Browser	Chrome for Android
Internet Connection	WiFi or 3G

Note: Android device testing is limited to popular Android devices from Samsung, Google, and HTC.

FUNCTIONAL SUPPORT ACROSS ENVIRONMENTS FOR PLAYER V4

This topic summarizes Player V4 functional support across environments.

AD SUPPORT

VAST 2.0 and 3.0

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

VPAID 2.0

- Ooyala Player V4 for HTML5

Ooyala Pulse

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS

Google IMA (including Google DFP and VPAID 1.0 Support)

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

FreeWheel

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

LiveRail



- Ooyala Player V4 for HTML5

ANALYTICS SUPPORT

Ooyala IQ Analytics

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

comScore Analytics

- Ooyala Player V4 for HTML5

Adobe Analytics (Omniture)

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

BRANDING SUPPORT (VISUAL ELEMENTS, SUCH AS COLORS, IMAGES, AND BUTTONS)

Skin Branding

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

CSS Branding

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

SECURITY SUPPORT

Ooyala Player Token

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Secure Playback Token

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Encrypted Streams

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Syndication Restrictions

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android



CLOSE CAPTIONS SUPPORT

Closed Captions - VOD

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Closed Captions - Live

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

PLAYBACK FEATURES SUPPORT

Adaptive Bitrate Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Full Screen Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Social Sharing Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Responsive Design Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Content-aware Controls Support

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

Live DVR Support (HLS Streams)

- Player V4 (requires the Bitmovin video plugin)
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

DISCOVERY SUPPORT

Discovery Interface Customization

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android



Discovery Recommendations Engine Configuration

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

DEVELOPER SUPPORT

Player JavaScript API

- Ooyala Player V4 for HTML5

Backlot REST API

- Ooyala Player V4 for HTML5

Authorization REST API

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

LIVE STREAMING SUPPORT

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

LOCALIZATION SUPPORT

- Ooyala Player V4 for HTML5
- Ooyala Mobile SDK for iOS
- Ooyala Mobile SDK for Android

INTEGRATIONS IN PLAYER V4

To increase engagement with your content, you can integrate your content and the Ooyala Player V4 for HTML5 with other environments and services.

INTEGRATION WITH OYALA PRODUCTS

The Ooyala Player V4 for HTML5 integrates with other Ooyala products.

Ooyala products that can currently integrate with the Ooyala Player V4 for HTML5 include:

- [Ooyala IQ Analytics User Guide](#)
- [Ooyala Discovery \(content recommendation\)](#)
- Ooyala Content Protection
 - Authentication with [Ooyala Player Token for Player V3 \(Deprecated\)](#)
 - Rights Management with [Rights Locker](#)
 - Authorization ([Limiting Concurrent Streams per Viewer, Stopping Unauthorized Streams](#)) with the [Player Authorization API for Player V3](#)
 - Access Control with [Content Publishing](#)
- [Ooyala Backlot \(asset management\)](#)



- Creating players with the [Backlot REST API](#) or the [Creating a Player in Backlot](#)
- Managing assets with the [Backlot REST API](#) or the [Content Upload and Management](#)
- Applying publishing rules with the [Backlot REST API](#) or [Creating Publishing Rules](#)
- Applying syndication controls with the [Backlot REST API](#) or [Syndication Controls Security Features and Relation to Programming](#)
- Generating a Player V4 embed code with the [Configuring Player Embed Settings in Backlot](#).

Note: Settings on the Player Branding sub-tab affect only Flash Player V3. For Player V4, see [Customizing the Player V4 Appearance](#) on page 116. For HTML5 Player V3, use the JavaScript API (see [Developing with the Player V3 JavaScript API \(Deprecated\)](#) and [apidocs.ooyala.com](#)).

Note: Player customizations made using the Backlot API affect only Flash Player V3. For Player V4, see [Customizing the Player V4 Appearance](#) on page 116. For HTML5 Player V3, use the JavaScript API (see [Developing with the Player V3 JavaScript API \(Deprecated\)](#) and [apidocs.ooyala.com](#)).

- [Ooyala Pulse User Guide](#)

You can also check out our [Mobile SDK for iOS](#) and [Mobile SDK for Android](#).

INTEGRATION WITH THIRD-PARTY PRODUCTS

Player integrations currently supported with the Player V4 include:

- [Plugins for Ad Managers](#)
 - [FreeWheel Ad Plugin](#) on page 143
 - [Google IMA Ad Plugin](#) on page 147
 - [Ooyala Pulse Ad Plugin](#) on page 160
 - [VAST and VPAID Ad Plugin](#) on page 171

Note: Use Liverail for VAST ads only.

IAB Ad Standards

- VAST 2.0 and 3.0
- VPAID 1.0 for Google IMA
- VPAID 2.0
- VMAP 1
- [Third-Party Analytics Providers](#)
 - [Adobe Analytics \(Omniture\) Plugin](#) on page 190
 - [comScore Analytics Plugin](#) on page 196
 - [Google Analytics Plugin](#) on page 199
 - [Nielsen Analytics Plugin](#) on page 204
 - [YOUBORA Analytics Plugin](#) on page 207
 - [Conviva Analytics Plugin](#) on page 197
- Social Media (pointing social networks to the hosted_at URL)
 - Facebook

Note: Currently you can only share the web page on Facebook. You cannot share the HTML5 video player inside Facebook.

 - Twitter
 - [Adobe Pass](#) (with authorization)



GETTING STARTED WITH PLAYER V4

This topic describes how to begin using Ooyala Player V4.

DO YOU WANT TO MIGRATE EXISTING PLAYERS TO PLAYER V4?

To migrate existing players from *Player V3 (Deprecated)*, see *Migrating to Player V4* on page 57.

Note: Customers using Player V2 should contact Ooyala immediately via Ooyala Technical Support or your Ooyala account representative.

HOW DO YOU WANT TO USE OYALA PLAYER V4?

Integrator Player Capabilities Include Level	Skills / Resources Needed	More Information
Standard	<ul style="list-style-type: none">Use Ooyala-hosted resources (software paths, plugins, skin) that are automatically updated.Use Backlot (only) to configure player settings.Assign a video asset to a player.Use the standard Player V4 embed code (see <i>Configuring Player Embed Settings in Backlot</i>).Use Ooyala IQ for analytics.Add standard Discovery configurations.Use Ads (Google IMA, Pulse, Freewheel).Use closed captions.	<p>Basic HTML knowledge - enough to insert HTML code into your web pages.</p> <ul style="list-style-type: none"><i>Basic Tutorial for Player V4</i> on page 53<i>Creating a Player in Backlot</i><i>Configuring a Player in Backlot</i><i>Embedding Player V4 on a Web Page</i> on page 69<i>Ooyala IQ Analytics User Guide</i><i>Discovering Content in Player V4</i> on page 134<i>Ads in Player V4</i> on page 142<i>Closed Captions in Player V4</i> on page 125
Intermediate (Standard capabilities plus: (Standard + Customized Skin))	<p>Basic skills plus:</p> <ul style="list-style-type: none">Specify inline (page-level) overrides of Ooyala default settings.Pull, modify, and host a custom player skin (skin.json file).Pull, modify and host all skin resources (add js and css) - More advanced.	<ul style="list-style-type: none"><i>Customizing the Player V4 Skin with skin.json</i> on page 120<i>Customizing the Player V4 Appearance Using CSS</i> on page 122
Advanced Options (Less Common)	<p>Intermediate capabilities plus:</p> <ul style="list-style-type: none">Configure ads using their third-party Ad providers (including page-level overrides).Add additional Player V4 plugins for Ads (SSAI, VAST), and Analytics (Conviva, NPAW, etc).	<p>Intermediate skills plus:</p> <ul style="list-style-type: none">Third-party ad providersThird-party analytics providersPlaylist management <ul style="list-style-type: none"><i>Ooyala-hosted Player V4 Resources</i> on page 77<i>Ads in Player V4</i> on page 142<i>Analytics in Player V4</i> on page 189<i>Localizing the Player V4 UI</i> on page 124



Integration Player Capabilities Include Level	Skills / Resources Needed	More Information
	<ul style="list-style-type: none"> Configure analytics using their third-party Analytics providers (including page-level overrides). Localize the Player. Configure playlists. Customize Discovery Recommendations. 	<ul style="list-style-type: none"> Using Playlists in Player V4 on page 136 Discovering Content in Player V4 on page 134
Self-hosted Same capabilities as Advanced, but enables using your own CDN for hosting all Player Resources. <small>(Engineering)</small>	<ul style="list-style-type: none"> Host all resources (plugins, skin) yourself Select the Advanced embed code. Package and minify. Requires manual player resource upgrades when Ooyala publishes software / skin updates. Create and host your own plugins. Roll up/consolidate only the plugins you need to use. <p>Note: Self-hosting requires the most life-cycle cost of all hosting options. Customers looking to self-host the Ooyala player should plan to regularly update the Ooyala player to newer releases.</p>	<ul style="list-style-type: none"> Hosting Player V4 Resources on page 81 Cross-Origin Resource Sharing (CORS) on page 88 Loading Video Plugins on page 89

PREREQUISITES FOR USING PLAYER V4

Prior to using Player V4, you must have the following resources.

ACCOUNT MANAGER/TECHNICAL SUPPORT RESOURCES

Please be sure to contact your account manager or technical support before you begin your migration.

TECHNICAL EXPERTISE

If you are customizing the look and feel of the player, you simply need to be able to modify a JSON file. If you are developing with the Player V4, you need experience with JavaScript development. With the



new player architecture, you need to load modules, such as the skin module, Discovery module, and ad module.

DEVELOPER RESOURCES

You can clone, host, fork, and modify forked versions of the [html5-skin](#) and [skin-config](#) repositories.

Warning: Ooyala will support any pre-defined and valid modifications to the skin.json file. If you fork the skin-config repository and make non-valid modifications, or if you fork the html5-skin repository and make changes, we will not offer technical support for these changes.

BASIC TUTORIAL FOR PLAYER V4

This tutorial walks you through the steps of adding a video with an Ooyala player onto a web page. It begins by configuring your player and video in Backlot, embedding the player on a web page, playing the video to verify your setup, and then configuring/testing additional Backlot settings.

Note: This tutorial uses the default Ooyala configuration in the pure SAAS (software as a service) model, in which Ooyala hosts and maintains all resources (software paths, plugins, and skin resources) for you. This provides the simplest way to embed and deploy an Ooyala player. Ooyala bundles all the player resources, minifies, and automatically provides bug fixes and updates if this methodology is used.

BEFORE YOU BEGIN / PREREQUISITES

Before you begin this tutorial, you should:

- Know how to edit HTML code on a web page.
- Have a Backlot user account.
- Upload a video into Backlot that you want to use.
- Have a web page on which you want to add a player to run your video asset.

For background information, see [Player V4](#) on page 6 and [Getting Started with Player V4](#) on page 51.

STEP 1: LOG INTO BACKLOT

Log into Backlot: <https://backlot.ooyala.com/>

STEP 2: CREATE A NEW PLAYER

1. Go to the Player Branding tab (click PUBLISH > Player Branding).
2. Go to the Player Selection sidebar.
3. Verify that **Player - V4** is the selected player type.

Note: If you created an account after May 1st 2017, only Player V4 is an option. If for some reason you need access to Player V3, please contact Ooyala Technical Support.

4. Click the **+New button**. Backlot adds a new player to the list.
5. On the Player tab, change the Player Name from New Player to something more descriptive (such as myPlayer).

For general information, see [Creating a Player in Backlot](#).

STEP 3: ASSIGN THE PLAYER TO A VIDEO ASSET

1. Go to the MANAGE tab.
2. Select a video in the list.



3. Go to the Embed sub-tab.
4. Under Custom Player, click the down arrow and select the player you just created.

Note:

- Content ID is the Ooyala-assigned ID for your selected video.
 - Player ID is the Ooyala-assigned ID for the selected player.
5. Go to the Details subtab and click the play button in the preview window to verify that your video runs correctly.

STEP 4: GET THE PLAYER EMBED CODE

1. Go to the Embed subtab.
2. Under Generate Embed Code, choose the following embed code type:

V4 HTML5 Standard Player Embed Code (recommended)

3. View the embed code by scrolling through the text box below the embed code type.

Note: In place of *YOUR_PLAYER_ID* and *YOUR_CONTENT_ID* below, you will see your Player ID and Content ID, respectively.

```
<!-- This embed code automatically includes and packages standard
Player V4 plugins. If you need a different mix of plugins, please
use the Advanced embed option. To see the included plugins, please
refer to this doc: http://help.ooyala.com/video-platform/concepts/
pbv4_release_notes.html -->
<!-- Include plugins necessary for player -->
<script src="//player.ooyala.com/core/YOUR_PLAYER_ID"></script>

<div id='container' ></div>

<script>
  var playerParam = {
    'autoplay':false,
    'loop':false
  };
  OO.ready(function() {window.pp = OO.Player.create('container',
  'YOUR_CONTENT_ID', playerParam);});
</script>
```

4. Click **Copy** to copy the embed code to your clipboard.

For general information, see [Configuring Player Embed Settings in Backlot](#).

STEP 5: EMBED THE PLAYER ON THE WEB PAGE

1. Open the HTML page on which you want to add the video.
2. If you do not have an existing page, create a blank one using something like the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Test Player V4 Web Page</title>
  </head>
  <body>

    </body>
</html>
```



- Paste the copied embed code onto the page between the `<body>...</body>` tags. Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Test Player V4 Web Page</title>
  </head>
  <body>
    <!-- This embed code automatically includes and packages standard
    Player V4 plugins. If you need a different mix of plugins, please
    use the Advanced embed option. To see the included plugins, please
    refer to this doc: http://help.oyala.com/video-platform/concepts/
    pbv4_release_notes.html -->
    <!-- Include plugins necessary for player -->
    <script src="//player.oyala.com/core/YOUR_PLAYER_ID"></script>

    <div id='container' ></div>

    <script>
      var playerParam = {
        'autoplay':false,
        'loop':false
      };
      OO.ready(function() {window.pp = OO.Player.create('container',
        'YOUR_CONTENT_ID', playerParam);});
    </script>
  </body>
</html>
```

- Save your changes to the page.

For general information, see [Embedding Player V4 on a Web Page](#) on page 69.

STEP 5: TEST PLAYBACK

- Load the page in a browser.
- Click the video to run it.

Your video should play in the page. If it does not, please check our [Player V4 FAQ](#) on page 29.

STEP 6: CONFIGURE AND TEST OTHER SETTINGS IN BACKLOT

- In Backlot, go to PUBLISH > Player Branding > Player.
- On the Player tab, enable (check) the Bitrates option to display the current bitrates during playback.
- On the Player tab, enable (check) the Always show scrubber bar option so that the scrubber bar is always displayed (does not automatically hide).
- Load the page in a browser.
- Click the video to run it.
- During playback, verify that you can now see the bitrates control.
- Review the other configurations to determine what player UI controls are ideal for your website.

Note: Because the player settings are distributed through our CDN, you will not immediately see the changes applied to the embedded player. It may take up to one (1) hour to apply to all geographies.

For general information, see [Configuring a Player in Backlot](#). If you need more controls over your Player's user interface than what you see in the Backlot settings, see other options at [Customizing the Player V4 Appearance](#) on page 116.



STEP 7: CONFIGURE AND TEST WHAT HAPPENS WHEN PLAYBACK ENDS (OPTIONAL)

You can configure what happens when playback ends with a video. Depending on how your Ooyala account is configured, you can configure either Discovery or End Screen settings.

To determine which option you should use:

1. In Backlot, go to PUBLISH > Player Branding.
2. Look at the subtab that is the furthest to the right.
 - **Discovery** means that discovery is enabled for your account.
 - **End Screen** means that discovery is not enabled for your account.
3. Complete the instructions (below) that apply for your account.

Step 7.1 - Configure End Screen Settings (Optional)

1. Go to PUBLISH > Player Branding > End Screen.
2. On the End Screen tab, enable Title and Description.
3. Go to MANAGE > Details and confirm or specify the video Title and Description.
4. When playback ends, verify that you can see the video title and description.

Important: If you use any test on the screens (Title/Descriptions), then the player will blur the video image on a player “pause” event. This is to make sure the text is readable. If you do not want to have the player blur the video image on pause, turn off the Title and Description elements.

For general information, see [Configuring a Player in Backlot](#).

Step 7.2: Configure Discovery Settings (Optional)

If Discovery is enabled for your account, you can configure player settings on the Discovery tab.

1. Go to PUBLISH > Player Branding > Discovery.
2. Select your player from the Player Selection list if it's not already selected.
3. On the Discovery tab, select Turn On Discovery For This Player.
4. Load the page in a browser.
5. Click the video to run it.
6. When playback ends, verify that you can see video recommendations.

For general information, see [Discovering Content in Player V4](#) on page 134.

STEP 8: VIEW ACTIVITY IN OYALA IQ ANALYTICS

1. Go to ANALYZE > Dashboard.
2. Look for statistics indicating that your video was played (see **Trending Past Hour**, **Popular Now**, and **Top 10 Countries**).

For general information, see [Ooyala IQ Analytics User Guide](#).

WHERE TO GO FROM HERE

- [Embedding Player V4 on a Web Page](#) on page 69
- [Customizing the Player V4 Appearance](#) on page 116
- [Ads in Player V4](#) on page 142
- [Analytics in Player V4](#) on page 189



MIGRATING TO PLAYER V4

This section describes how to migrate from Ooyala Player V3 to Player V4.

MIGRATING FROM PLAYER V3

Important: Ooyala Player V3 is deprecated and is scheduled to be *disabled* on 2018-01-31. After that date, Player V3 will no longer play your video or audio content. Customers still using Player V3 need to migrate to Player V4.

For customers familiar with Player V3, this section describes common migration tasks.

- [*Using Existing Video Assets and Players*](#)
- [*Common Types of Migrations from Player V3*](#)
- [*Migration Tasks*](#)
- [*Migrating Content Protection from Player V3*](#)
- [*Migrating Ads*](#)
- [*Determining Streaming Options*](#)

BEFORE YOU MIGRATE

Before you begin, learn about Player V4, starting with:

- [*Player V4*](#) on page 6 for an overview
- [*Getting Started with Player V4*](#) on page 51 to help you assess your technical requirements for Player V4 deployment
- [*Basic Tutorial for Player V4*](#) on page 53 for a walkthrough of basic Player V4 features

The easiest way to begin migrating to Player V4 is to start by using Ooyala as the default host for Player V4 resources (see [*Ooyala-hosted Player V4 Resources*](#) on page 77).

USING EXISTING VIDEO ASSETS AND PLAYERS

If you have already been publishing videos using Ooyala Player V3, you can re-use the following elements in Ooyala Player V4 for HTML5.

Existing Video Assets

If you already have video assets defined in Backlot, you can play them (using their Content IDs) in Player V4 if they are supported by HTML5 Player V3, along with HLS on additional devices and DASH on additional platforms.

Note: The HTTP Dynamic Streaming (HDS), including Akamai HDS (or HD2) streaming protocol, use Flash. Browsers have been disabling and blocking the Flash plugin. As a result, Ooyala strongly recommends that customers encode and use HLS and Dash moving forward.

Existing Players

If you have existing players defined in Backlot, you can re-use them by converting them to a Player V4 player. In the Backlot UI, simply go to PUBLISH > Player Branding > Player, select your player, then change the player version to V4. Note that any existing Player V3 settings that are not supported in Player V4 will be lost. Alternatively, you can simply start over and create a new player (see [*Creating a Player in Backlot*](#)).

Other Assets



You can also re-use images, remote assets, and closed captions (supported formats only; see [Closed Captions in Player V4](#) on page 125).

COMMON TYPES OF MIGRATIONS FROM PLAYER V3

This section describes a few common types of migrations from Player V3 to Player V4.

Basic Migration

If you do not have existing Player V3 ad plugins or customizations, dynamic skinning, or Discovery, your migration tasks are straightforward. For each HTML page on which you have embedded Player V3, replace the HTML Embed Code (from Backlot) with the new V4 HTML5 Standard Player Embed Code (from Backlot, see [Configuring Player Embed Settings in Backlot](#)), then save and test your changes. See [Embedding Player V4 on a Web Page](#) on page 69 for more information.

Customers Using Ads

If you use ads with Player V3, you will need to start using the Player V4 ad plugins.

- To use an ad plugin, [load the ad plugin](#) on the HTML5 page where you load your video player and test to see that your ad functionality works as expected. See [Ads in Player V4](#) on page 142 for details.
- If you use Backlot to generate your HTML embed code (see [Configuring Player Embed Settings in Backlot](#)):
 - The V4 HTML5 Standard Player Embed Code automatically includes the Pulse, Google IMA, Freewheel and VAST ad plugins. Do not add these plugins on the web page. If your player deployment requires a plugin that is not automatically included, you must add any required plugins manually.
 - The V4 HTML5 Player Embed Code (Advanced) does *not* include ad plugins, even if you apply ad settings in Backlot. You must also manually add the desired ad plugins.
 - If you used page-level ad parameters with Player V3, see [Ad Configuration Parameters and Settings](#) for changes to the ad configuration parameters and settings. Some of the parameters you used in Player V3 might be deprecated in Player V4. You might need to update your page-level settings accordingly.

For more information, see [Migrating Ads](#).

Customers Using Discovery

If you used Discovery with Player V3 and want to continue to use it with Player V4, you might need to load the Player V4 Discovery plugin on the HTML5 page where you load your video player. If you use Backlot to generate your HTML embed code (see [Configuring Player Embed Settings in Backlot](#)):

- The V4 HTML5 Standard Player Embed Code automatically includes the Discovery plugins. Do not add the Discovery plugin on the web page. If your player deployment requires a plugin that is not automatically included, you must add any required plugins manually.
- The V4 HTML5 Player Embed Code (Advanced) does *not* include the Discovery plugin, even if you apply discovery settings in Backlot. You must manually add the Discovery plugin. For example:

```
<script src="//player.oyala.com/core/MY_PLAYER_ID"></script>
<script src="//player.oyala.com/static/v4/production/other-plugin/
discovery_api.min.js"></script>
```

See [Discovering Content in Player V4](#) on page 134 for full details.

Customers with Custom Plugins, CSS, and JS

If you use customized plugins, CSS, and JavaScript with Player V3, read through the rest of this topic to identify which Player V4 changes affect your ad plugins, analytics plugins, and the Player API. Take the appropriate actions to implement the changes in Player V4. For CSS customization, see [Customizing the Player V4 Appearance Using CSS](#) on page 122.



MIGRATION TASKS

To use existing video assets with Player V4:

1. Either convert an existing Player V3 player to V4 by changing its version (see above), or create and configure a new Player V4 player.
2. Follow the instructions in [Basic Tutorial for Player V4](#) on page 53 and create a V4 player, substituting your own assetId, playerBrandingId, and pcode from step 1. Alternatively, for a more customized UI for your player, you can go to the Player V4 html5-skin repo README at <https://github.com/ooyala/html5-skin#examples> and substitute your assetId, playerBrandingId, and pcode into the example page.
3. If you have any other assets, such as images, you can specify them when you configure the Player V4, as described in [Customizing the Player V4 Appearance](#) on page 116.

Note: All resources not hosted by Ooyala (video, audio, images, closed captions, css, js, etc.) that are used with your HTML5-based playback must have the appropriate CORS headers configured. For details, see [Hosting Player V4 Resources](#) on page 81 and [Cross-Origin Resource Sharing \(CORS\)](#) on page 88.

MIGRATING FROM ADVANCED TO THE STANDARD EMBED CODE

If you have existing Player V4 pages that use the prior (now Advanced) embed code and want to use the new standard embed option:

1. In the Backlot UI, go to the Embed sub-tab (MANAGE > Embed).
2. Under Generate Embed Code, choose the following embed code type from the list:

V4 HTML5 Standard Player Embed Code (recommended)

3. Click Copy to copy the embed code to your clipboard.
4. In a web page editor, open the web page on which you have embedded a Player V4 player.
5. Paste the new embed code into your page.
6. Copy and paste any of the advanced plugins (only) into the <head> section.

Important: Do not specify any of the standard plugins that have already been included in the standard embed code. Refer to the [Ooyala Player V4 Release Notes](#) on page 235 to see which player plugins are included. Any inline plugin references in your page code should be removed when transitioning to the standard plugin. Any plugins NOT included in the standard embed should remain as added inline.

7. Copy and paste any other code (e.g. page-level parameters or inline overrides) from the old embed code into the new embed code.

For general instructions, see [Configuring Player Embed Settings in Backlot](#).

MIGRATING CONTENT PROTECTION FROM PLAYER V3

You can use the following content protection features with Player V4:

- Ooyala Player Token
- Rights Locker API
- encrypted stream AES-128 for HLS
- Widevine Modular, PlayReady and FairPlay DRM support

If you are currently using Widevine Classic (which is deprecated), please contact Technical Support or your account manager to help you transition to Widevine Modular.



MIGRATING ADS

	VPAID 1.0	VPAID 2.0	VAST 3.0	VMAP
Player Platform	HTML5	HTML5	HTML5	HTML5
Device Reach	desktop only	desktop and mobile	desktop and mobile	desktop and mobile
Ad Presentation	Flash Ads in SWF	JavaScript based	N/A	N/A
Advantages	<ul style="list-style-type: none"> Flash creatives ad inventory still high mobile ads interactivity greater device reach lightweight, JS-based interaction 	<ul style="list-style-type: none"> mobile ads interactivity greater device reach lightweight, JS-based interaction 	<ul style="list-style-type: none"> supports ad pods better error reporting skippable linear ads 	<ul style="list-style-type: none"> create and manage ad playlist program ad breaks VAST 3.0 responses
Disadvantages	<ul style="list-style-type: none"> more risk of blockage (due to Flash) no mobile support 	<ul style="list-style-type: none"> still emerging not much testing 	-	-

DETERMINING STREAMING OPTIONS

The following table describes Player V4 stream options, including migration from Player V3.

Player V3 Usage	Player V4 Stream Options (and Notes)	Notes
Flash Player V3 with (CLEAR) HDS streams	<ul style="list-style-type: none"> Player V4 HLS (bit_wrapper.min.js) (recommended) HDS (osmf_flash.min.js) <p>Note: The HTTP Dynamic Streaming (HDS), including Akamai HDS (or HD2) streaming protocol, use Flash. Browsers have been disabling and blocking the Flash plugin. As a result, Ooyala strongly recommends that customers encode and use HLS and Dash moving forward.</p>	<ul style="list-style-type: none"> HLS requires preprocessing HLS: Use bit_wrapper.min.js (Chrome v47+, Firefox v43+, Safari v8+)
HTML5 Player V3 with MP4 Streams	<ul style="list-style-type: none"> Player V4 HLS (bit_wrapper.min.js) (recommended) or MP4 (main_html5.min.js) 	<ul style="list-style-type: none"> MP4 is progressive. Not using adaptive bitrate (ABR). HLS requires preprocessing
HTML5 Player V3 with HLS on Safari and MP4 on other browsers	<ul style="list-style-type: none"> Player V4 HLS (bit_wrapper.min.js) and MP4 (main_html5.min.js) 	There is an option to use the encoding priority if you are not satisfied with MP4



Player V3 Usage	Player V4 Stream Options (and Notes plugin)	
Flash Player V3 with Adobe Access HDS Streams	Player V4 HLS (<i>bit_wrapper.min.js</i>)	<ul style="list-style-type: none"> • You must preprocess assets • DASH requires preprocessing

DIFFERENCES BETWEEN PLAYER VERSIONS

For customers familiar with Player V3, this topic describes some of the differences you will see in Player V4.

- Customizing the UI
- Third-Party Analytics Plugins from Player V3
- Ad Configuration Parameters and Settings
- Other JavaScript API Changes

CUSTOMIZING THE PLAYER APPEARANCE

To customize the Player V4 UI, you can configure the player interface using a variety of ways - the Backlot UI, page-level parameters, in-line overrides, the `skin.json` config file, and CSS properties. For instructions, see [Customizing the Player V4 Appearance](#) on page 116.

THIRD-PARTY ANALYTICS PLUGINS FROM PLAYER V3

The [Analytics Framework](#) on page 210 allows you to integrate third-party analytics with the player.

Plugins

You can use the Analytics Framework to re-create your integrations with third-party analytics providers. We offer out-of-the box integration plugins for top third-party analytics providers. Alternatively, you could work with Ooyala Professional Services to assist with your migration.

Event Messages

The Analytics Framework uses event messages that are different from the Player V3 and Player V4 events. The table below shows the mapping between the events the Analytics Framework is expecting and the Player V3/Player V4 events. You can use this mapping to create analytics plugins.

Event in the Player OO.EVENTS namespace (Example: OO.EVENTS.PLAYER_CREATED)	Event in the Analytics Framework OO.Analytics.EVENTS namespace (Example: OO.Analytics.EVENTS.VIDEO_PLAYER_CREATED)
PLAYER_CREATED	VIDEO_PLAYER_CREATED
EMBED_CODE_CHANGED	VIDEO_SOURCE_CHANGED
METADATA_FETCHED	VIDEO_STREAM_METADATA_UPDATED
CONTENT_TREE_FETCHED	VIDEO_CONTENT_METADATA_UPDATED
INITIAL_PLAY	INITIAL_PLAYBACK_REQUESTED
PLAYED	PLAYBACK_COMPLETED
PLAY	VIDEO_PLAY_REQUESTED
PLAYING	VIDEO_PLAYING
PAUSE	VIDEO_PAUSE_REQUESTED



Event in the Player OO.EVENTS namespace (Example: OO.EVENTS.PLAYER_CREATED)	Event in the Analytics Framework OO.Analytics.EVENTS namespace (Example: OO.Analytics.EVENTS.VIDEO_PLAYER_CREATED)
PAUSED	VIDEO_PAUSED
REPLAY	VIDEO_REPLAY_REQUESTED
SEEK	VIDEO_SEEK_REQUESTED
SEEKED	VIDEO_SEEK_COMPLETED
DOWNLOADING	VIDEO_STREAM_DOWNLOADING
BUFFERING	VIDEO_BUFFERING_STARTED
BUFFERED	VIDEO_BUFFERING_ENDED
WILL_PLAY_ADS	AD_BREAK_STARTED
ADS_PLAYED	AD_BREAK_ENDED
DESTROY	DESTROY

Each Analytics Framework message may have data sent with it. To see what data is sent with each event, please read the documentation for the [Analytics Framework API](#).

AD CONFIGURATION PARAMETERS AND SETTINGS

With the [Ad Manager Framework](#) on page 186 Ad Manager Framework for Player V4, configuration parameters and some page level settings are different from Player V3. If you don't have any page level overrides to Backlot settings, your ads will have full backwards compatibility.

General Ad Changes

The following global ad parameters are new in Player V4. See [Configuring Ad Parameters](#) on page 180 for full descriptions of these parameters.

- `pauseAdOnClick`
- `replayAds`
- `showNonLinearCloseButton`
- `showLinearAdSkipButton`
- `linearAdSkipButtonStartTime`
- `allowAdClickThroughOnVideo`

The following ad parameters are not supported by Player V4.

Global Ad Parameters Not Supported by Player V4	Player V4 Equivalent
<code>showInAdControlBar</code>	Modify skin.json to modify ad control bar functionality. This applies to the FreeWheel ad plugin and the VAST and VPAID plugin when rendering VAST 2.0 or VAST 3.0 ads.
<code>showAdMarquee</code>	For the Google IMA ad plugin, this parameter will be ignored. It will not display the ad control bar for ads unless you use the page level override (<code>showAdControls</code>) for the plugin. Google provides its own UI and this prevents our UI from obscuring it. For the VAST and VPAID plugin, rendering



Global Ad Parameters Not Supported by Player V4	Player V4 Equivalent
	VPAID 2.0 ads, the ad control also is forcibly set to off. This is done to avoid blocking ad interactivity.

With the new Ad Manager Framework, ad manager plugins are used to integrate ad managers with the player. For Google IMA, VAST, FreeWheel, VPAID, and any other ad manager, you must load the ad manager plugin in the HTML page (for example, ad_manager_vast.min.js) before the player is created for the ad integration to work for your player.

Google IMA-Specific Changes

You no longer load Google IMA ads on the page with:

```
"google-ima-ads-manager": {
    "adTagUrl": "yourAdTagUrl"
}
```

Ad tags are now wrapped in an array, and adTagUrl has been renamed tag_url. See [Integrating Google IMA Ads](#) on page 149 for full details on how to implement Google IMA ads.

The following IMA-specific parameters are not supported by Player V4:

IMA Parameters Not Supported by Player V4	Player V4 Equivalent
adTagUrl	Use tag_url. See Integrating Google IMA Ads on page 149 for full details on how to implement Google IMA ads.
useStyledNonLinearAds	No Player V4 equivalent.
showInAdControlBar	This parameter is deprecated globally and for this ad manager. For the Google IMA ad plugin, the ad control bar configurations in skin.json are ignored (and are forcibly set to off) to avoid blocking ad interactivity.
showAdMarquee	For the Google IMA ad plugin, the ad marquee configurations in skin.json are ignored (and are forcibly set to off) to avoid blocking ad interactivity.
playWhenAdClick	Use the pauseAdOnClick global parameter.
adRequestTimeout	Use the adLoadTimeout global parameter

The following table compares which ad positions are supported in Player V4 (HTML5 players and mobile SDKs for iOS and Android) and Player V3 (HTML players and mobile SDKs for iOS and Android).

Supported Ad Position	Player V4	Player V3
Pre-Roll	✓	✓
Mid-Roll	✓	✓
Post-Roll	✓	✓
Podded	✓	✓
Skippable	✓	✓
Clickthrough	✓	✓
Overlay	✓	



Supported Ad Position	Player V4	Player V3
Companion	✓	

Freewheel-specific Changes

The following table compares which ad positions are supported in Player V4 (HTML5 players and mobile SDKs for iOS and Android) and Player V3 (HTML players and mobile SDKs for iOS and Android).

Supported Ad Position	Player V4	Player V3
Pre-Roll	✓	✓
Mid-Roll	✓	
Post-Roll	✓	✓
Podded	✓	
Skippable	✓	
Clickthrough	✓	✓
Overlay	✓	

VAST-specific Changes

The `vastAds` parameter is not currently supported with the new VAST plugin. See [Integrating VAST and VPAID Ads](#) on page 172 for details about VAST ad integration.

VPAID-Specific Changes

VAST and VPAID ad integration functionality is now combined in one plugin. See [Integrating VAST and VPAID Ads](#) on page 172 for details about VAST and VPAID ad integration. The following VPAID-specific parameters are not supported by Player V4:

VPAID Parameters Not Supported by Player V4	Player V4 Equivalent
<code>vpaid-ads-manager</code>	Use <code>vast-ads-manager</code> . See Integrating VAST and VPAID Ads on page 172 for full details on how to implement VPAID ads.
<code>adLayout</code>	No Player V4 equivalent.
<code>adTag</code>	Use <code>tag_url</code> . See Integrating VAST and VPAID Ads on page 172 for full details on how to implement VPAID ads.
<code>showAdMarquee</code>	For the VAST and VPAID plugin rendering VPAID 2.0 ads, the ad marquee configurations in <code>skin.json</code> are ignored (and are forcibly set to off) to avoid blocking ad interactivity.
<code>showInAdControlBar</code>	For the VAST and VPAID plugin rendering VPAID 2.0 ads, the ad control bar configurations in <code>skin.json</code> are ignored (and are forcibly set to off) to avoid blocking ad interactivity.
<code>playWhenAdClick</code>	Use the <code>pauseAdOnClick</code> global parameter.

LiveRail-Specific Changes

Player V4 supports LiveRail VAST ad tags through the VAST and VPAID plugin. See [Integrating VAST and VPAID Ads](#) on page 172 for details about the VAST and VPAID ad integration.



The VAST and VPAID plugin supports LiveRail ad tag URLs that are VAST XML. This correlates to the LiveRail (VAST) Overlay and LiveRail (VAST) In-Stream ad set types in Backlot.

Note: The VAST and VPAID plugin does not support LiveRail ads that do not use the VAST XML format.

The following LiveRail-specific parameters are not supported by Player V4:

LiveRail Parameters Not Supported by Player V4 Player V4 Equivalent

liverail-ads-manager	Use <code>vast-ads-manager</code> . See Integrating VAST and VPAID Ads on page 172 for full details on how to implement VAST ads.
LR_PUBLISHER_ID	No Player V4 equivalent.
LR_PARTNERS	No Player V4 equivalent.
LR_TAGS	Use <code>tag_url</code> for the LiveRail VAST ad tag. See Integrating VAST and VPAID Ads on page 172 for full details on how to implement VPAID ads.
LR_ADMAP	No Player V4 equivalent.

OTHER JAVASCRIPT API CHANGES

In addition to the changes described above, this section describes changes in the Player JavaScript API for Player V4.

Embedded Parameters

In Player V4, there are some changes to the embedded parameters (called "page-level parameters" in Player V4) that you pass to the `OO.Player.create()` method. See [Page-level Parameters for Player V4](#) on page 99 for full descriptions of supported embedded parameters.

New Embedded Parameters in V4. The following parameters are among those that have been added in Player V4:

- `encodingPriority`
- `initialVolume`
- `initialBitrate`

For a complete list, see [Page-level Parameters for Player V4](#) on page 99.

Embedded Parameters Not Supported in V4. Certain Player V3 parameters are no longer supported in Player V4, including (but not limited to) the following:

Global Parameters Not Supported by Player V4 Player V4 Equivalent

Embedded Parameters

in-stream	No Player V4 equivalent.
locale	Modify <code>skin.json</code> to modify localization of the Player UI. See Localizing the Player V4 UI on page 124 for details.
prefetching	No Player V4 equivalent.
tvRatingsPosition	No Player V4 equivalent.
tvRatingsTimer	No Player V4 equivalent.
useFirstVideoFromPlaylist	No Player V4 equivalent.
enableChannels	No Player V4 equivalent.



Global Parameters Not Supported by Player V4	Player V4 Equivalent
showInAdControlBar	Modify skin.json to modify ad control bar functionality. This applies to the FreeWheel ad plugin and the VAST and VPAID plugin when rendering VAST 2.0 or VAST 3.0 ads.
showAdMarquee	For the Google IMA ad plugin, this parameter will be ignored. It will not display the ad control bar for ads unless you use the page level override (showAdControls) for the plugin. Google provides its own UI and this prevents our UI from obscuring it. For the VAST and VPAID plugin, rendering VPAID 2.0 ads, the ad control also is forcibly set to off. This is done to avoid blocking ad interactivity.

Custom Module Parameters

devModuleCategory	No Player V4 equivalent.
devModuleURL	No Player V4 equivalent.
hide	No Player V4 equivalent.
layout	No Player V4 equivalent.
thruParam_dev-other	No Player V4 equivalent.

Query String Parameters

Player V3 query string parameters (`namespace`, `platform`, and `tweaks`) are not applicable in Player V4.

Player Events

Player events have changed between Player V3 and Player V4.

Deprecated Player Event. The following Player V3 event has been deprecated in Player V4, and will not work. Ooyala has removed this event.

- `SET_TARGET_BITRATE_QUALITY`

New Player Events. The following events are among those that have been added in Player V4. You can see detailed event descriptions at apidocs.ooyala.com.

- `BITRATE_INFO_AVAILABLE`
- `BITRATE_CHANGED`
- `SET_TARGET_BITRATE`
- `SET_CLOSED_CAPTIONS_LANGUAGE`
- `CLOSED_CAPTIONS_INFO_AVAILABLE`

PLAYER MIGRATION FAQ

Important: Ooyala Player V3 is deprecated and is scheduled to be *disabled* on 2018-01-31. After that date, Player V3 will no longer play your video or audio content. Customers still using Player V3 need to migrate to Player V4.

For general information, see:

- [Player V4 FAQ](#) on page 29
- [Planning Your Migration](#)

Q: How do I migrate from Player V3 to Player V4?



A: To load existing video assets with Player V4, simply create a new Player V4 and associate the Asset ID of the existing video with the new player. You can use the Backlot UI to configure many player settings (see [Configuring a Player in Backlot](#)), and you have additional customization options as well (see [Customizing the Player V4 Appearance](#) on page 116). If you use ads, you might need to update your ad configuration parameters. With the [Ad Manager Framework](#) on page 186, configuration parameters and some page level settings are different from Player V3. If you don't have any page level overrides to Backlot settings, your ads will have full backwards compatibility. See [Migrating from Player V3](#) on page 57 for full migration steps and details.

Q: How long do I have to migrate from Player V3 to Player V4?

A: Player V3 is deprecated and is scheduled to be disabled on 2018-01-31. After that date, Player V3 will no longer play your video or audio content. Customers still using Player V3 need to migrate to Player V4 before that date. Contact your Ooyala account representative for assistance.

Q: How do I enable Flash-free playback for Player V4?

A: By default, the Ooyala player uses the Flash plugin to playback HLS across all browsers. The bit_wrapper plugin for DASH and HLS (bit_wrapper.min.js) plays HLS using the HTML5 standard (and Media Source Extensions), enabling HLS on web browsers without the use of Flash. HTML5 MSE also supports playback of MPEG-DASH (clear and DRM-protected content). To enable Flash-free playback using the Ooyala player, see [Using Flash-free Playback with Player V4](#) on page 94.

Q: What happens if I have a custom Ooyala Player Framework (OPF) Flash module for Player V3?

A: OPF modules in Player V3 were developed to allow you to integrate with third-party analytics providers and make UI/UX customizations (swf files, etc.). The [Analytics Framework](#) on page 210 allows you to integrate third-party analytics with the player. You can use the Analytics Framework to re-create your integrations with third-party analytics providers.

Q: Will custom skins created for Player V3 work with Player V4?

A: Player V4 CSS customizations are made by modifying html5-skin.min.css. If you have built your own CSS for Player V3, you will need to re-create these skin customizations with html5-skin.min.css for Player V4. For details, see [Customizing the Player V4 Appearance Using CSS](#) on page 122.

Q: Do I need to reprocess my clear assets to move from Player V3 to Player V4?

A: No, not usually. Re-processing is required only if:

- You want to add DASH encoding to your processing profile.
- You want to add HLS to your processing profile and you have MP4 or HDS ONLY encodings.

Q: Do I need to reprocess my DRM-protected assets to move from Player V3 to Player V4?

A: Yes. Reprocessing is required in order to support [Apple Fairplay](#) HLS (Apple requires Apple-issued certificates), [Widevine Modular](#), and [Playready](#) DASH.

Q: Are there any changes to the Player APIs and message bus?

A: Yes, query string parameters, certain embedded parameters, and certain player events used with Player V3 are no longer supported. There are also many new parameters and events introduced with Player V4. A complete list of changes is captured in [Differences Between Player Versions](#) on page 61.

Q: How do I embed the player in third-party websites - is it the same as Player V3? Where's the hosted iframe.html?

A: The player can be loaded in any third party, fourth party site . The iFrame can actually be found in the player 'embed' tab. See [Embedding Player V4 in an HTML iframe](#) on page 95 for details.

Q: I see Player V3 and Player V4 embeds in the Backlot UI. What is the difference between them?

A: The Player V3 embeds will load Player V3 only. The Player V4 HTML5 embed will load Player V4. Player V3 embeds will continue to coexist with the Player V4 HTML5 embed while Ooyala continues to



support Player V3. The Player V4 embed allows you to simply copy and paste the page script into an HTML page.



EMBEDDING PLAYER V4 ON A WEB PAGE

These topics describe how to embed Player V4 into a web page.

PLAYER V4 PLUGINS

Player V4 is a modular player consisting of plugins, which means that you can add or remove certain player functionality depending on which plugins you load on your HTML page. Some plugins are required for your player to function and other plugins are optional, depending on what functionality you want your player to have. For details on the default location of these plugins, see [Ooyala-hosted Player V4 Resources](#) on page 77.

Note: Plugins must all be from the same Player V4 version. Mixing versions is not recommended.

Important: If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, certain plugins are automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).

REQUIRED PLUGINS

Required plugins include:

- Core player ([core.min.js](#))
- At least one video plugin:
 - Bitmovin plugin for DASH and HLS ([bit_wrapper.min.js](#))
 - HLS and MP4 Main video plugin ([main_html5.min.js](#))
 - Ooyala Player Plugin for YouTube iFrame ([youtube.min.js](#)) (Deprecated)
 - OSMF Flash plugin for HDS ([osmf_flash.min.js](#)) (Deprecated)

Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

- Akamai HD video plugin for Akamai packaged HDS ([akamaiHD_flash.min.js](#)) (Deprecated)

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

See [Flash Video Rendering](#) on page 94 for details on Flash video rendering.

OPTIONAL PLUGINS

Optional plugins include:

- HTML5 skin plugin ([html5-skin.min.js](#))
- Skin CSS ([html5-skin.min.css](#))
- Skin customization file that can be applied to the HTML5 player, Android SDK player, and iOS SDK player ([skin.json](#))
- Ad plugins
 - Ooyala Pulse ad plugin ([pulse.min.js](#))
 - Ooyala SSAI Pulse plugin for live ad insertion ([ssai_pulse.min.js](#))
 - FreeWheel ad plugin ([freewheel.min.js](#))
 - VAST and VPAID ad plugin ([ad_manager_vast.min.js](#))



- Google IMA ad plugin ([google_ima.min.js](#))
- Third-Party Analytics Plugins
 - comScore analytics plugin ([comScore Analytics Plugin](#) on page 196) (ask your account manager for access to this plugin)
 - Adobe Analytics (Omniture) plugin ([omniture.min.js](#))
 - Nice People at Work YOUBORA plugin ([YOUBORA Analytics Plugin](#) on page 207)
 - Google Analytics plugin ([Google Analytics Plugin](#) on page 199)
 - Nielsen Analytics plugin ([Nielsen Analytics Plugin](#) on page 204)
 - Conviva Analytics plugin ([Conviva Analytics Plugin](#) on page 197)
- Discovery plugin ([discovery_api.min.js](#))
- Playlists plugin ([playlists.js](#))
- SSAI Pulse plugin [ssai_pulse.js](#) (it is recommended that you host this plugin). For more information, see [Ooyala Pulse Live Stream Integration \(SSAI\)](#).

PLUGIN BEST PRACTICES

The following are best practices for using the video plugins.

- You must load the Core Player and at least one video plugin for Player V4 to function properly.
- Generally, when using the [bit_wrapper.min.js](#) plugin, you should always load it first, before the other video plugins.
- You cannot use the [bit_wrapper.min.js](#) plugin with iOS. Please note that you can still include this plugin on the page (it just won't do anything). This means that you don't have to change which plugins you load based on device.
- To support FreeWheel ad playback, you must include the [main_html5.min.js](#) plugin.
- To support VPAID ad playback, you must load the [main_html5.min.js](#) plugin first among the video plugins.
- The [osmf_flash.min.js](#) plugin does not currently support Google IMA Ad Rules.
- For DASH streams on desktop web, we recommend using the [bit_wrapper.min.js](#) plugin with Chrome.
- For MP4 streams, we recommend using the [main_html5.min.js](#) plugin.
- If you would like to use HLS streams on desktop, we recommend that you use our [bit_wrapper.min.js](#) plugin on Chrome, Firefox and Internet Explorer. For Safari and Edge we encourage you to use the [main_html5.min.js](#) plugin instead.
- The [main_html5.min.js](#) plugin does not report bitrate information. If you want to track bitrate information (see [Analytics in Player V4](#) on page 189), you must use the the Bitmovin plugin for DASH and HLS, OSMF Flash plugin for HDS (Deprecated), or the Akamai HD video plugin for Akamai packaged HDS (Deprecated).

STREAM SUPPORT PER BROWSER

Note: The OSMF and Akamai plugins (deprecated) allow backwards compatibility with existing HTTP Dynamic Streaming (HDS) and Akamai HDS (or HD2) assets. However, HDS and HDS2 use a Flash component, and browsers will soon be disabling the Flash plugin. Going forward, Ooyala strongly recommends that customers encode and use HLS and Dash instead. This may involve re-encoding existing assets where possible. Given industry changes and the increased importance of mobile playback, we expect to deprecate support for Flash-based formats (such as HDS and HD2) soon. Customers should consider these as legacy formats that are being phased out over time.

Desktop Web Stream and Browser Support

The following table shows which Ooyala video plugin to load for each stream type and browser for desktop web players.



Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

Browser	HLS and eHLS	DASH	MP4	HDS (Deprecated)
Chrome v47+	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Safari v8+	bit_wrapper.min.js or main_html5.min.js	bit_wrapper.min.js (remote and local assets supported in Player v4.3.3+)	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Firefox v43+	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Internet Explorer v11 ⁽¹⁾	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Microsoft Edge	bit_wrapper.min.js or main_html5.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)

Note: ⁽¹⁾HLS and eHLS are supported only on Windows 8+.

Mobile Web Stream and Browser Support

The following table shows which Ooyala video plugin to load for each stream type and browser for mobile web players.

Browser	HLS and eHLS	DASH	MP4
Android v4.1 and v4.3+ Chrome	<ul style="list-style-type: none"> main_html5.min.js 	<ul style="list-style-type: none"> bit_wrapper.min.js 	<ul style="list-style-type: none"> bit_wrapper.min.js or main_html5.min.js
iOS v8+ Safari	<ul style="list-style-type: none"> main_html5.min.js 	Not supported	<ul style="list-style-type: none"> main_html5.min.js



STREAM OPTIONS

Note: The HTTP Dynamic Streaming (HDS), including Akamai HDS (or HD2) streaming protocol, use Flash. Browsers have been disabling and blocking the Flash plugin. As a result, Ooyala strongly recommends that customers encode and use HLS and Dash moving forward.

Device	Supported Streams	Recommended Stream	Supported Player V4 Plugins
Desktop	MP4, HLS, eHLS, HDS (Deprecated), DASH	HLS	<ul style="list-style-type: none"> • all video plugins • all ad plugins • all analytics plugins • html5-skin.min.js • Discovery plugin
Android Web	MP4, HLS, eHLS, DASH	DASH	<ul style="list-style-type: none"> • main_html5.min.js and bit_wrapper.min.js • all ad plugins • all analytics plugins • html5-skin.min.js • Discovery plugin
iOS Web	MP4, HLS, eHLS	HLS	<ul style="list-style-type: none"> • main_html5.min.js • all ad plugins • all analytics plugins • (iPad only) html5-skin.min.js • Discovery plugin
Android App	MP4, HLS, eHLS, DASH	HLS or DASH	N/A - use the mobile SDK for Android
iOS App	HLS, eHLS, MP4	HLS	N/A - use the mobile SDK for iOS

Note: Player V4 is not supported on Xbox, Roku, Amazon Fire TV, Connected/Smart TVs, or Chromecast devices for this release.

DECIDING WHICH PLUGINS TO LOAD

The following table shows which plugins you should load given your specific use case:

If you want to...	Load the following in your HTML page...
Load a V4 player without the Ooyala skin	<ul style="list-style-type: none"> • Core player (core.min.js) • At least one video plugin <ul style="list-style-type: none"> • Bitmovin plugin for DASH and HLS (bit_wrapper.min.js) • HLS and MP4 Main video plugin (main_html5.min.js) • OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated)



If you want to...	Load the following in your HTML page...
Load a V4 player with the Ooyala skin	<ul style="list-style-type: none"> • Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated) <p>See Examples of Player V4 Web Page Embedding on page 113 for a general implementation example.</p> <p>See Flash Video Rendering on page 94 for details on Flash video rendering.</p> <ul style="list-style-type: none"> • Core player (core.min.js) • At least one video plugin <ul style="list-style-type: none"> • Bitmovin plugin for DASH and HLS (bit_wrapper.min.js) • HLS and MP4 Main video plugin (main_html5.min.js) • OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated) • Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated) • HTML5 skin plugin (html5-skin.min.js) • Skin CSS (html5-skin.min.css) • Skin customization file (skin.json) <p>See Examples of Player V4 Web Page Embedding on page 113 for a general implementation example and Customizing the Player V4 Appearance on page 116 for details on how to customize your player.</p>
Load a V4 player with the Ooyala skin and Ooyala Pulse ad functionality	<ul style="list-style-type: none"> • Core player (core.min.js) • At least one video plugin <ul style="list-style-type: none"> • Bitmovin plugin for DASH and HLS (bit_wrapper.min.js) • HLS and MP4 Main video plugin (main_html5.min.js) • OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated) • Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated) • HTML5 skin plugin (html5-skin.min.js) • Skin CSS (html5-skin.min.css) • Skin customization file (skin.json) • Ooyala Pulse ad plugin (pulse.min.js) <p>Note: If you are using the Freewheel, VAST, or Google IMA ad plugins, you can load multiple ad plugins on a page (Integrating Multiple Ad Managers on page 177). Otherwise, you can load only one ad plugin per player (at a time).</p>



If you want to...

Load a V4 player with the Ooyala skin and **FreeWheel ad functionality**

Load the following in your HTML page...

See [Integrating Ooyala Pulse Ads](#) on page 161 for an implementation example.

- Core player (core.min.js)
- At least one video plugin
 - Bitmovin plugin for DASH and HLS (bit_wrapper.min.js)
 - HLS and MP4 Main video plugin (main_html5.min.js)
 - OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated)
 - Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated)
 - HTML5 skin plugin (html5-skin.min.js)
 - Skin CSS (html5-skin.min.css)
 - Skin customization file (skin.json)
 - **FreeWheel ad plugin (freewheel.min.js)**

Note: If you are using the Freewheel, VAST, or Google IMA ad plugins, you can load multiple ad plugins on a page ([Integrating Multiple Ad Managers](#) on page 177). Otherwise, you can load only one ad plugin per player (at a time).

See [Integrating FreeWheel Ads](#) on page 143 for an implementation example.

Load a V4 player with the Ooyala skin and **VAST and/or VPAID ad functionality**

- Core player (core.min.js)
- At least one video plugin
 - HLS and MP4 Main video plugin (main_html5.min.js)
 - OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated)
 - Bitmovin plugin for DASH and HLS (bit_wrapper.min.js)
 - Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated)
 - HTML5 skin plugin (html5-skin.min.js)
 - Skin CSS (html5-skin.min.css)
 - Skin customization file (skin.json)
 - **VAST and VPAID ad plugin (ad_manager_vast.min.js)**

Note: If you are using the Freewheel, VAST, or Google IMA ad plugins, you can load multiple ad plugins on a page ([Integrating Multiple Ad Managers](#) on page 177). Otherwise, you can load only one ad plugin per player (at a time).

See [Integrating VAST and VPAID Ads](#) on page 172 for an implementation example.



If you want to...	Load the following in your HTML page...
Load a V4 player with the Ooyala skin and Google IMA ad functionality	<ul style="list-style-type: none"> Core player (core.min.js) At least one video plugin <ul style="list-style-type: none"> Bitmovin plugin for DASH and HLS (bit_wrapper.min.js) HLS and MP4 Main video plugin (main_html5.min.js) OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated) Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated) HTML5 skin plugin (html5-skin.min.js) Skin CSS (html5-skin.min.css) Skin customization file (skin.json) Google IMA ad plugin (google_ima.min.js) <p>Note: If you are using the Freewheel, VAST, or Google IMA ad plugins, you can load multiple ad plugins on a page (Integrating Multiple Ad Managers on page 177). Otherwise, you can load only one ad plugin per player (at a time).</p> <p>See Integrating Google IMA Ads on page 149 for an implementation example.</p>
Load a V4 player with the Ooyala skin and Ooyala Discovery functionality	<ul style="list-style-type: none"> Core player (core.min.js) At least one video plugin <ul style="list-style-type: none"> Bitmovin plugin for DASH and HLS (bit_wrapper.min.js) HLS and MP4 Main video plugin (main_html5.min.js) OSMF Flash plugin for HDS (osmf_flash.min.js) (Deprecated) Akamai HD video plugin for Akamai packaged HDS (akamaiHD_flash.min.js) (Deprecated) HTML5 skin plugin (html5-skin.min.js) Skin CSS (html5-skin.min.css) Skin customization file (skin.json) Discovery plugin (discovery_api.min.js) <p>See Examples of Player V4 Web Page Embedding on page 113 for a general implementation example.</p>

SPECIFYING PLUGINS FOR THE STANDARD EMBED CODE

The standard embed code (see [Configuring Player Embed Settings in Backlot](#)) provides a package of commonly used plugins so that you do not need to specify each one explicitly on a web page. However, for



further optimization, you may explicitly specify only the player plugins you want in the embed script directly. This will typically make the player payload smaller than the default. For example:

```
<script src="http://player.oyala.com/core/<playerid>?
plugins=main,bm,ima,disc"></script>
```

This script dynamically packages, minifies, and delivers just the main_html5.js, bit_wrapper.js, google_ima.js, and discovery_api.js plugins, respectively, in addition to automatically loading the minimal required plugins (core.min.js and html5-skin.min.js).

PLUGINS THAT ARE LOADED AUTOMATICALLY

The following plugins are required for the player to work and are therefore loaded automatically:

- core.min.js
- html5-skin.min.js

You do not specify these plugins on the web page.

PLUGINS YOU CAN SPECIFY

You can specify any of the following values to the `plugins` parameter. Other than the plugins that are loaded automatically, if you do not specify a plugin, it will not be loaded.

Type	Plugin	Value for plugins Param
video (at least one)	main_html5.min.js	main
	bit_wrapper.min.js	bm
	osmf_flash.min.js	osmf
	akamaiHD_flash.min.js	akmi
	youtube.min.js (deprecated)	yt
ads	freewheel.min.js	fw
	google_ima.min.js	ima
	ad_manager_vast.min.js	vast
	pulse.min.js	pulse
other	playlists.min.js	pl
	discovery_api.js	disc
	ssai_pulse.js	ssai
analytics	omniture.min.js	adobe
	Nielsen.min.js	nielsen
	conviva.min.js	conviva
	googleAnalytics.min.js	ga

LOADING SEQUENCE

Plugins are loaded in sequence:

- first by plugin type - automatic, video, then all other types (ads, other, analytics), and then
- by the sequence specified in the `plugins` parameter.



For example, if you specified the following on a web page:

```
<script src="http://player.oyala.com/core/<playerid>?
plugins=disco,main,bm,ima"></script>
```

The video plugins (main, bm) would be loaded first, then the disc plugin (other category), and then ima (ads category).

OOYALA-HOSTED PLAYER V4 RESOURCES

This topic describes the Player V4 resources (core player, plugins, and skin resources) that you can use when you embed Player V4 on a web page. If you want to host these resources yourself (instead of Ooyala), see [Hosting Player V4 Resources](#) on page 81.

OOYALA RESOURCE PATHS

The Ooyala Player is hosted, distributed, and maintained on URL paths. These paths are automatically updated with new Player V4 releases and are subject to our 99.9% uptime SLA.

production

The following path is recommended for most Ooyala customers:

```
//player.oyala.com/static/v4/production/
```

Note: Use this path only if your site does *not* have mixed content conditions (refer to this [Google article](#)) running over HTTPS.

This path is typically one release behind the latest Ooyala version. Ooyala keeps this path up to date but it does not contain the very latest player updates. This approach allows you to test your site on a staging environment with the production/latest version (see below) to verify the effect of player updates on any customizations. Example: //player.oyala.com/static/v4/production/core.js

production/latest

The following path contains the latest Ooyala player version that has been certified for release.

```
//player.oyala.com/static/v4/production/latest/
```

Note: Use this path only if your site does *not* have mixed content conditions running over HTTPS.

This path is recommended for Ooyala customers who need capabilities only available in the most recent release. Example: //player.oyala.com/static/v4/production/latest/core.js

Static Ooyala-Hosted Player Paths

The Ooyala player is also hosted on static paths.

stable/latest

The following path is required for mixed-content conditions:

```
//player.oyala.com/static/v4/stable/latest/
```

This path is set indefinitely to Player V4 Web v4.10.6, which is last player version that does not cause browsers to enforce mixed content rules with Flash playback. Example: //player.oyala.com/static/v4/stable/latest/core.js

stable/version

The Ooyala Video Player is also hosted on version-specific paths (stable/#.#.#) for each player release (refer to the [Ooyala Player V4 Release Notes](#) on page 235 for details). Using the static player paths is not recommended, as you will miss getting important player software updates automatically. If you



point to specific player versions, you will be entirely responsible for manually updating player resources to newer versions. Example: //player.oyala.com/static/v4/stable/4.13.4/core.min.js

LIST OF PLAYER V4 RESOURCES

At a minimum, you must load the Core Player and at least one video plugin for Player V4 to function properly. If you do not load at least one video plugin, your player will not play videos.

Note: Wherever you see ***url_where_hosted*** in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

Resource	Name / Location	Description
Core Player	<i>url_where_hosted</i> /core.min.js	Required.
Skin Resources		
HTML5 Skin	<i>url_where_hosted</i> /skin-plugin/html5-skin.min.js	Customizing the Player V4 Appearance on page 116 Note: To view the source in GitHub, see https://github.com/oyala/html5-skin .
Skin CSS	<i>url_where_hosted</i> /skin-plugin/html5-skin.min.css	Customizing the Player V4 Appearance Using CSS on page 122
Skin Config File	<i>url_where_hosted</i> /skin-plugin/skin.json	Customizing the Player V4 Skin with skin.json on page 120 Note: skin.json is no longer required for the Ooyala Player skin to be used. Do not specify skin.json if you want to use the Player skins in Backlot. If skin.json is included at the page level, it will overwrite all Backlot player settings in the Publish tab.
Skin iFrame	<i>url_where_hosted</i> /skin-plugin/iframe.html	Embedding Player V4 in an HTML iframe on page 95
Localization Files	<ul style="list-style-type: none">• <i>url_where_hosted</i>/skin-plugin/en.json• <i>url_where_hosted</i>/skin-plugin/es.json• <i>url_where_hosted</i>/skin-plugin/zh.json	Localizing the Player V4 UI on page 124 provides instructions for adding your own localization files.
Images and Fonts	<ul style="list-style-type: none">• <i>url_where_hosted</i>/skin-plugin/assets/images/oyala-watermark.png• <i>url_where_hosted</i>/skin-plugin/assets/images/oyala.png	Image and font resources are required for the player skin to appear properly. In general, image and font files can be found in the GitHub repository at: <ul style="list-style-type: none">• html5-skin/assets/fonts• html5-skin/assets/images
Video Recommendation	<i>url_where_hosted</i> /other-plugin/discovery_api.min.js	Discovering Content in Player V4 on page 134



Resource	Name / Location	Description
(Discovery) plugin		
Video Plugins		<p><i>Loading Video Plugins</i> on page 89</p> <p>Note: At least one video plugin is required.</p>
Bitmovin Video Plugin for DASH and HLS	Player V4 Version 4.11.13 or higher <ul style="list-style-type: none"> • <code>url_where_hosted/video-plugin/bit_wrapper.min.js</code> Player V4 Version 4.10.6 or earlier (required for mixed content conditions) <ul style="list-style-type: none"> • <code>//player.oyala.com/static/v4/stable/latest/video-plugin/bit_wrapper.min.js</code> 	<ul style="list-style-type: none"> • This plugin is required for Live DVR (HLS streams only). • For DASH video, you must use the AAC or mp4a.40.2 audio codecs. • This plugin is not used with Safari or Safari Mobile. Playback on these browsers requires the <code>main_html5.js</code> plugin. • For this release, this plugin does not support Live closed captions for HLS. <p>Note: The order of the video plugins represents the order of priority for the plugin usage. For HTML5 video, we recommend loading the <code>bit_wrapper.min.js</code> first, followed by <code>main_html5.js</code>.</p>
Main Video Plugin for HLS and MP4	<code>url_where_hosted/video-plugin/main_html5.min.js</code>	<i>Loading Video Plugins</i> on page 89
OSMF Flash Video Plugin for HDS (Deprecated)	<code>url_where_hosted/video-plugin/osmf_flash.min.js</code>	<p><i>Loading Video Plugins</i> on page 89</p> <p>Note: The Flash HDS plugin does not currently support IMA Ad Rules.</p> <p>Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the OVP Release Notes.</p>
Akamai HD Video Plugin for Akamai Packaged HDS (Deprecated)	<code>url_where_hosted/video-plugin/akamaiHD_flash.min.js</code>	<p><i>Loading Video Plugins</i> on page 89</p> <p>Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the OVP Release Notes.</p>
Ooyala Player Plugin for YouTube iFrame (Deprecated)	<code>url_where_hosted/video-plugin/youtube.min.js</code>	<i>Playing YouTube Videos in Player V4 (Deprecated)</i> on page 97
Ad Plugins		<i>Ads in Player V4</i> on page 142
Ooyala Pulse Ad Plugin	<code>url_where_hosted/ad-plugin/pulse.min.js</code>	<i>Ooyala Pulse Ad Plugin</i> on page 160



Resource	Name / Location	Description
Ooyala SSAI Pulse Plugin for Live Ad Insertion	url_where_hosted /ad-plugin/ssai_pulse.min.js	Ooyala Pulse Live Stream Integration (SSAI) Note: It is assumed that if the SSAI Pulse plugin is loaded that the stream will have server side ads inserted.
Google IMA Plugin	url_where_hosted /ad-plugin/google_ima.min.js	Google IMA Ad Plugin on page 147 To view the source in GitHub, see https://github.com/ooyala/html5-ad-plugins/blob/master/js/google_ima.js .
FreeWheel Plugin	url_where_hosted /ad-plugin/freewheel.min.js	FreeWheel Ad Plugin on page 143 To view the source in GitHub, see https://github.com/ooyala/html5-ad-plugins/blob/master/js/freewheel.js . Note: To support FreeWheel ad playback, you must also include the Main Video Plugin. See Loading Video Plugins on page 89.
VAST and VPAID Plugin	url_where_hosted /ad-plugin/ad_manager_vast.min.js	VAST and VPAID Ad Plugin on page 171 To view the source in GitHub, see https://github.com/ooyala/html5-ad-plugins/blob/master/js/ad_manager_vast.js . Note: To support VPAID ad playback, you must load the Main Video Plugin first among the video plugins.
Analytics Plugins		Analytics in Player V4 on page 189
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • url_where_hosted/analytics-plugin/VideoHeartbeat.min.js • url_where_hosted/analytics-plugin/AppMeasurement.js • url_where_hosted/analytics-plugin/VisitorAPI.js • url_where_hosted/analytics-plugin/omniture.min.js 	Adobe Analytics (Omniture) Plugin on page 190
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	comScore Analytics Plugin on page 196
Conviva Analytics plugin	<ul style="list-style-type: none"> • url_where_hosted/analytics-plugin/conviva-core-sdk.min.js • url_where_hosted/analytics-plugin/conviva.min.js 	Conviva Analytics Plugin on page 197. Note: You must specify conviva-core-sdk.min.js before conviva.min.js.
Google Analytics plugin	url_where_hosted /analytics-plugin/googleAnalytics.min.js	Google Analytics Plugin on page 199



Resource	Name / Location	Description
Nielsen Analytics plugin	<i>url_where_hosted</i> /analytics-plugin/Nielsen.min.js	<i>Nielsen Analytics Plugin</i> on page 204
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	<i>YOUTBORA Analytics Plugin</i> on page 207 Note: NPAW maintains the YOUTBORA Analytics plugin for Player V4.
Other Feature Plugins		
Playlists plugin	<i>url_where_hosted</i> /other-plugin/playlists.js	<i>Using Playlists in Player V4</i> on page 136

HOSTING PLAYER V4 RESOURCES

Ooyala hosts default Player V4 resources that are updated automatically. You have the option to host Player V4 resources yourself, including plugins, your HTML skin configuration file, images, and other resources. Advantages to hosting the plugins and files yourself include:

- You can optimize fast downloads for your personal routes.
- You can optimize even more by downloading these files and then concatenating all of your plugins and core into a single file which allows everything to download at once. This is faster and more reliable.
- If you are making any player look and feel customizations you will need to host skin configuration files anyway, so it will not be much more effort to host more files at the same endpoint.

Note: All hosted resources must be available and accessible to Player V4 users.

PLUGIN VERSIONS

Note: Plugins must all be from the same Player V4 version. Mixing versions is not recommended.

PLAYER V4 RESOURCES TO HOST

At a minimum, you must load the Core Player and at least one video plugin for Player V4 to function properly. If you do not load at least one video plugin, your player will not play videos.

If you are using the Ooyala player skin, you must also load the skin plugin (html5-skin.min.js), skin CSS (html5-skin.min.css), image resources (html5-skin/assets/images) and font resources (html5-skin/assets/fonts). You can use the default Ooyala image and font resources or load your own fonts and images.

Note: Wherever you see *url_where_hosted* in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see *Ooyala-hosted Player V4 Resources* on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see *Hosting Player V4 Resources* on page 81), be sure to check for any path dependencies within the files.

Resource	Name of File to Host
Core Player	<i>url_where_hosted</i> /core.min.js
Skin Resources	
HTML5 Skin	<i>url_where_hosted</i> /skin-plugin/html5-skin.min.js



Resource	Name of File to Host
Skin CSS	<i>url_where_hosted</i> /skin-plugin/html5-skin.min.css
Skin Config File	<i>url_where_hosted</i> /skin-plugin/skin.json Note: skin.json is no longer required for the Ooyala Player skin to be used. Do not specify skin.json if you want to use the Player skins in Backlot. If skin.json is included at the page level, it will overwrite all Backlot player settings in the Publish tab. See <i>Order of Precedence in Player V4 Settings</i> on page 118.
Skin iFrame	<i>url_where_hosted</i> /skin-plugin/iframe.html
Localization Files	<ul style="list-style-type: none"> • <i>url_where_hosted</i>/skin-plugin/en.json • <i>url_where_hosted</i>/skin-plugin/es.json • <i>url_where_hosted</i>/skin-plugin/zh.json
Images and Fonts	<ul style="list-style-type: none"> • <i>url_where_hosted</i>/skin-plugin/assets/images/ooyala-watermark.png • <i>url_where_hosted</i>/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin	<i>url_where_hosted</i> /other-plugin/discovery_api.min.js
Video Plugins	<p>Note: When you embed a player on a web page, the order of the video plugins represents the order of priority for the plugin usage. For HTML5 video, we recommend putting the bit_wrapper.js first, followed by main_html5.js.</p>
Bitmovin Video Plugin for DASH and HLS	<p>Player V4 Version 4.11.13 or higher</p> <ul style="list-style-type: none"> • <i>url_where_hosted</i>/video-plugin/bit_wrapper.min.js • <i>url_where_hosted</i>/video-plugin/bitmovinplayer-controls.min.css • <i>url_where_hosted</i>/video-plugin/bitmovinplayer-controls.min.js • <i>url_where_hosted</i>/video-plugin/bitmovinplayer-core.min.css • <i>url_where_hosted</i>/video-plugin/bitmovinplayer-core.min.js • <i>url_where_hosted</i>/video-plugin/bitmovinplayer-vr.min.js • <i>url_where_hosted</i>/video-plugin/bitmovinplayer.min.js • <i>url_where_hosted</i>/video-plugin/bitmovinplayer.swf <p>Player V4 Version 4.10.6 or earlier (required for mixed content conditions)</p> <ul style="list-style-type: none"> • https://player.ooyala.com/static/v4/stable/latest//video-plugin/bit_wrapper.min.js • https://player.ooyala.com/static/v4/stable/latest//video-plugin/bitdashplayer.min.js • https://player.ooyala.com/static/v4/stable/latest//video-plugin/bitdashplayer.swf
Main Video Plugin for HLS and MP4	<i>url_where_hosted</i> /video-plugin/main_html5.min.js
OSMF Flash Video Plugin for HDS (Deprecated)	<i>url_where_hosted</i> /video-plugin/osmf_flash.min.js



Resource	Name of File to Host
	<p>Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the OVP Release Notes.</p> <p>Note: If you host this plugin, you must also host the following file in the same location as the <code>osmf_flash.min.js</code> file: <code>http://player.ooyala.com/static/v4/stable/latest_player_version/video-plugin/osmf_flash.swf</code></p> <p>Important: Do not change the name of the <code>osmf_flash.js</code> file. If you bundle multiple video plugins into a single file, and if <code>osmf_flash.js</code> is one of the plugins in the bundle, then the file name (for the bundled file) must be <code>osmf_flash.js</code>.</p>
Akamai HD Video Plugin for Akamai Packaged HDS (Deprecated)	<code>url_where_hosted/video-plugin/akamaiHD_flash.min.js</code> <p>Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the OVP Release Notes.</p>
Ooyala Player Plugin for YouTube iFrame (Deprecated)	<code>url_where_hosted/video-plugin/youtube.min.js</code>
Ad Plugins	
Ooyala Pulse Ad Plugin	<code>url_where_hosted/ad-plugin/pulse.min.js</code>
Ooyala SSAI Pulse Plugin for Live Ad Insertion	<code>url_where_hosted/ad-plugin/ssai_pulse.min.js</code>
Google IMA Plugin	<code>url_where_hosted/ad-plugin/google_ima.min.js</code>
FreeWheel Plugin	<code>url_where_hosted/ad-plugin/freewheel.min.js</code>
VAST and VPAID Plugin	<code>url_where_hosted/ad-plugin/ad_manager_vast.min.js</code>
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • <code>url_where_hosted/analytics-plugin/VideoHeartbeat.min.js</code> • <code>url_where_hosted/analytics-plugin/AppMeasurement.js</code> • <code>url_where_hosted/analytics-plugin/VisitorAPI.js</code> • <code>url_where_hosted/analytics-plugin/omniture.min.js</code>
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • <code>url_where_hosted/analytics-plugin/conviva-core-sdk.min.js</code> • <code>url_where_hosted/analytics-plugin/conviva.min.js</code>
Google Analytics plugin	<code>url_where_hosted/analytics-plugin/googleAnalytics.min.js</code>
Nielsen Analytics plugin	<code>url_where_hosted/analytics-plugin/Nielsen.min.js</code>
YOUTORA Analytics Plugin	<code>//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js</code>
	<p>Note: Unlike other Player V4 plugins, it is not recommended that you try to host this plugin yourself.</p>
Other Feature Plugins	



Resource	Name of File to Host
Playlists plugin	<i>url_where_hosted</i> /other-plugin/playlists.js

UPDATING TO THE LATEST PLAYER VERSION

To update an Ooyala Player version from a previous version:

1. Download and host all of the new player resources (core plugin, *skin plugins, CSS, font, and image resources, video plugins, ad plugins, analytics plugins*, and so on) and update your resources to point to these new files.
2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.

TIPS FOR HOSTING PLAYER V4 RESOURCES

Hosting Player V4 resources involves the following steps:

1. Copy the resources you need from the default Ooyala location described in [Player V4 Resources to Host](#).

Note: If you are building plugins or other resources from within your own development environment (instead of pulling them from <http://player.ooyala.com/static/v4/stable/>*latest_player_version/**), refer to the applicable README.md file (for example, [README.md](#)).

2. Optionally, change any of the file(s) that you want to customize.
3. Upload each file to the server that will host the resource.

When hosting:

- You can use whatever file path / hierarchy you want. However, the location must be accessible to all player users.
- In addition to the required plug-ins (see [Player V4 Plugins](#) on page 69), be sure to include all the plugins you use for your players.
- You must meet cross-domain requirements, which are in [Cross-domain Hosting Considerations](#) below.
- You must handle any other requirements. For example, for the OSMF Flash Plugin for HDS and Bitmovin Plugin for DASH and HLS, you need to modify the Flash cross-domain policy file (see [Loading Video Plugins](#) on page 89).

CROSS-DOMAIN HOSTING CONSIDERATIONS

- If your Player V4 configuration involves resources that are hosted across multiple domains, certain resources (including JavaScript modules, skin-config settings file, language files, images) must have the appropriate CORS headers configured (see [Cross-Origin Resource Sharing \(CORS\)](#) on page 88). You do not need to add CORS headers if all Player V4 resources are hosted under a single domain, such as your own domain or Ooyala.
- To host any of the default Player V4 resources, simply download the latest version from the fixed Ooyala URL and then upload it to your server location.
- If you plan to make changes to the html5-skin repository, you'll need to set up a local build environment according to the instructions in the Developer Setup section of [README.md](#).
- If you are using the Bitmovin plugin (*bit_wrapper.min.js*), OSMF Flash plugin (*osmf_flash.min.js*, Deprecated), or Akamai HD plugin (*akamaiHD_flash.min.js*, Deprecated), you need to configure cross-domain compatibility. In your Flash cross-domain policy file, be sure to grant access to both `http://` and `https://` URLs. For more information, see [Adobe's Cross-domain Policy documentation](#).



COMBINING PLAYER V4 PLUGIN FILES

If you host plugins yourself, you can combine all of the plugins you use into a single .js file to optimize performance. It is simply faster to load one .js file than multiple .js files. At a minimum, you must include the core player plugin, at least one video plugin, and any other plugins that you reference when you embed Player V4 onto a web page.

Note: If you include the `osmf_flash.js` file, then the combined file *must* be named `osmf_flash.js`.

Ways to Combine Plugins

There are two ways to create the combined file:

- In a text editor, create a master plugin file (with a .js extension), then copy the master file.
- If you build your .js files locally, then change your build process so that it generates a single output file that aggregates the JavaScript code from all of the plugins that you use. Next, copy the .js file to the hosting server. Finally, on each web page where you have embedded Player V4 code, be sure to reference just the single .js file.

Example With Multiple Plug-ins

The following example shows the `<head>` section from a web page that references the individual plugins:

```
<head>
  <title>My Test Player V4 Web Page</title>
  <script src="url_where_hosted/core.min.js"></script>
  <script src="url_where_hosted/main_html5.min.js"></script>
  <script src="url_where_hosted/discovery_api.min.js"></script>
  <script src="url_where_hosted/html5-skin.min.js"></script>
  <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
</head>
```

Example with a Combined Plugin

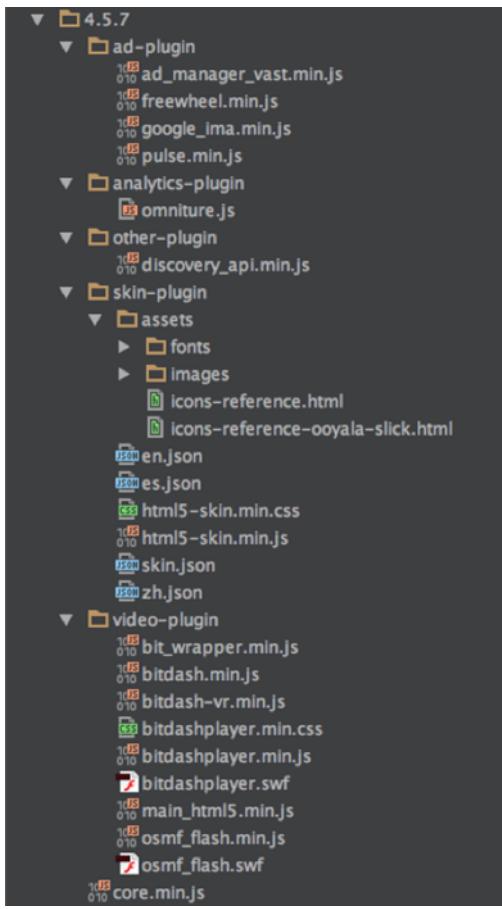
The following example references a combined plugin (named `my_master_plugin.js`), which includes `core.min.js`, `main_html5.min.js`, `discovery_api.min.js`, and `html5-skin.min.js`:

```
<head>
  <title>My Test Player V4 Web Page</title>
  <script src="url_where_hosted/my_master_plugin.js"></script>
  <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
</head>
```

BEST PRACTICES FOR SELF-HOSTING AND DIRECTORY STRUCTURE

This section describes an example of a self-hosting project structure that you can use for self-hosting JavaScript code. Note that not everything is required - it focuses on how to set up the necessary and most important files.





Refer to the following HTML page. It is assumed that the HTML page is located at the root of the project folder.

```
<!DOCTYPE html>
<html>
    <head>
        <script src="core.min.js"></script>
        <script src="video-plugin/main_html5.min.js"></script>
        <script src="video-plugin/bit_wrapper.min.js"></script>
        <script src="skin-plugin/html5-skin.min.js"></script>
        <link rel="stylesheet" href="skin-plugin/html5-skin.min.css"/>
    </head>

    <body>
        <div id='container'></div>
        <script>
            var playerParam = {
                "PCODE": "YOUR_PCODE",
                "playerBrandingId": "YOUR_PLAYER",
                "skin": {
                    "config": "skin-plugin/skin.json"
                }
            };
            OO.ready(function() {
                window.pp = OO.Player.create('container', 'YOUR_CONTENT_ID',
                    playerParam);
            });
        </script>
    </body>
</html>
```



This shows the files necessary to self-host your player. We assume that everything is located on the same project folder.

- core.min.js
- main_html5.min.js
- html5-skin.min.js
- skin.min.css
- skin.json
- Other resources needed to build self-hosting page:
 - en.json: English translation text for UI. You will also need to modify skin.json > localization > availableLanguageFile > languageFile to use skin-plugin/en.json
 - assets/ folder that contains fonts and static image resources for skin
 - If you are using bit_wrapper.min.js video plugin:
 - bitdashplayer.min.js
 - bitdashplayer.min.css
 - bitdashplayer.swf

File	Source
core.min.js	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/core.min.js
main_html5.min.js	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/video-plugin/main_html5.min.js
bit_wrapper.min.js	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/video-plugin/bit_wrapper.min.js
bitdashplayer.min.js	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/video-plugin/bitdashplayer.min.js
bitdashplayer.min.css	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/video-plugin/bitdashplayer.min.css
bitdashplayer.swf	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/video-plugin/bitdashplayer.swf
html5-skin.min.js	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/skin-plugin/html5-skin.min.js
html5-skin.min.css	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/skin-plugin/html5-skin.min.css
skin.json	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/skin-plugin/skin.json
en.json	//player.ooyala.com/static/v4/stable/RELEASE_VERSION/skin-plugin/en.json
assets/ folder	https://github.com/ooyala/html5-skin/releases

1. Download and extract a zip file.



File	Source
	2. Copy the content of the <code>assets</code> folder into <code>project/assets/</code> .

Note: All resources not hosted by Ooyala (video, audio, images, closed captions, css, js, etc.) that are used with your HTML5-based playback must have the appropriate CORS headers configured. For details, see [Cross-domain Hosting Considerations](#).

CROSS-ORIGIN RESOURCE SHARING (CORS)

Cross-Origin Resource Sharing (CORS) lets web pages make requests to a domain *other than* the one from which the page was served; it allows JavaScript to make requests across domain boundaries. Without CORS, this functionality is not available and web pages can run into cross-domain issues when trying to retrieve files, such as closed captions, from another domain.

All resources not hosted by Ooyala (video, audio, images, closed captions, css, js, etc.) that are used with your HTML5-based playback **must** have the appropriate [CORS headers](#) configured. If your resources do not follow the CORS standard, your player will not function correctly and may fail when trying to access files across domains.

You must set up CORS enablement on the CDN that you use for content serving. For information on how to enable CORS, see <http://enable-cors.org/> or talk to your hosting provider or vendor.

For example, due to the fact that the domain of a closed caption file could be different than the page and/or the video tag itself, we set `crossorigin='anonymous'` on the video tag. This causes any `streamURL` that does not properly have the CORS header enabled to fail to load.

Note: These CORS prerequisites do not apply if you are using only Ooyala-hosted content, as CORS is enabled for all Ooyala-hosted domains.

BASIC EMBEDDING INFORMATION

This topic describes the information you will need to provide when you embed Player V4 on a web page. See for details.

RETRIEVE YOUR PCODE

Player V4 requires that you provide your pcode, which you can retrieve from Backlot.

1. Log into your Backlot account.
2. Click the **ACCOUNT** tab.
3. Click the **Developers** subtab.
4. Look for your API key on the left side of the screen.
5. Copy your pcode, which is the alphanumeric string that precedes the period. For example, if the API key is xxxxxxxxxxxxxxxxxxxxxxxxx.xxxxx, then use the first 28 characters (xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx) for the pcode.

RETRIEVE A PLAYER BRANDING ID

Player V4 requires that you provide a `playerBrandingId`, which references a player defined in your Backlot account. Backlot displays this as `Player_ID`. To retrieve this from Backlot:

1. Log into your Backlot account.



2. Click the **MANAGE** tab.
3. Click the **Embed** subtab.
4. Select your player from the Custom Player dropdown.
5. Copy its `Player ID`, which you need to provide in the `playerBrandingId`.

Alternatively, you can retrieve the Player ID with the Backlot API. See the [Backlot API reference](#) for details.

You can use players that have already been defined for Player V4 (see [Migrating to Player V4](#) on page 57), or you can create an entirely new player.

There are two ways in which to create a player in Backlot:

- Backlot UI ([Creating a Player in Backlot](#))
- Backlot API (see [Creating a Player Using the Backlot REST API](#))

RETRIEVING THE ASSET ID

Player V4 requires that you provide an `asset ID` that uniquely identifies the video asset. An asset ID is the same as a `Content ID` (in the Backlot UI) and an `embed_code` (in the Player V4 JavaScript API). To retrieve this from Backlot:

1. Log into your Backlot account.
2. Click the **MANAGE** tab.
3. Click the **Embed** subtab.
4. Select your content from the list of video assets.
5. Copy its `Content ID`, which you need to provide in the `assetId`.

RETRIEVING A DEFAULT EMBED CODE

You can retrieve a default Player V4 embed code from Backlot and paste it into your web page.

1. Log into your Backlot account.
2. Click the **MANAGE** tab.
3. Click the **Embed** subtab.
4. Under **Custom Player**, select a Player V4 player.
5. In the **Generate Embed Code** section, configure any settings you want to change. For details, see [Configuring Player Embed Settings in Backlot](#).
6. Copy the embed code and paste it into your web page.

Note: The default player does not include all available plugins. You might need to add some manually after pasting the default code, including any analytics plugins (except Ooyala IQ), playlists, and some add plugins (SSAI, VAST/VPAID). See [Player V4 Plugins](#) on page 69 for a complete list.

LOADING VIDEO PLUGINS

Video plugins create video elements and video wrappers, and decode video encodings. They provide the Ooyala Player information about the encoding and encryption types they support, and the player uses that information, along with a priority order for encodings, to select which plugin to use when playing a stream. You can load any number of video plugins on your page.

Note: Plugins must all be from the same Player V4 version. Mixing versions is not recommended.

Important: If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, certain video plugins are automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).



Follow these steps to load video plugins for your web page:

- [Step 1: Load Supported Video Plugins](#) on page 91
- [Step 2: \(Optional\) Specify Video Encoding Priority](#) on page 92

See [Complete Example](#) on page 93 to view the code for the entire web page.

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

STREAM SUPPORT PER BROWSER

Note: The OSMF and Akamai plugins (deprecated) allow backwards compatibility with existing HTTP Dynamic Streaming (HDS) and Akamai HDS (or HD2) assets. However, HDS and HDS2 use a Flash component, and browsers will soon be disabling the Flash plugin. Going forward, Ooyala strongly recommends that customers encode and use HLS and Dash instead. This may involve re-encoding existing assets where possible. Given industry changes and the increased importance of mobile playback, we expect to deprecate support for Flash-based formats (such as HDS and HD2) soon. Customers should consider these as legacy formats that are being phased out over time.

Desktop Web Stream and Browser Support

The following table shows which Ooyala video plugin to load for each stream type and browser for desktop web players.

Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

Browser	HLS and eHLS	DASH	MP4	HDS (Deprecated)
Chrome v47+	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Safari v8+	bit_wrapper.min.js or main_html5.min.js	bit_wrapper.min.js (remote and local assets supported in Player v4.3.3+)	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Firefox v43+	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)
Internet Explorer v11 ⁽¹⁾	bit_wrapper.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai



Browser	HLS and eHLS	DASH	MP4	HDS (Deprecated)
Microsoft Edge	bit_wrapper.min.js or main_html5.min.js	bit_wrapper.min.js	bit_wrapper.min.js or main_html5.min.js	packaged HDS (Deprecated) osmf_flash.min.js (Deprecated) or akamaiHD_flash.min.js for Akamai packaged HDS (Deprecated)

Note: ⁽¹⁾HLS and eHLS are supported only on Windows 8+.

Mobile Web Stream and Browser Support

The following table shows which Ooyala video plugin to load for each stream type and browser for mobile web players.

Browser	HLS and eHLS	DASH	MP4
Android v4.1 and v4.3+ Chrome	• main_html5.min.js	• bit_wrapper.min.js	• bit_wrapper.min.js or • main_html5.min.js
iOS v8+ Safari	• main_html5.min.js	Not supported	• main_html5.min.js

BEST PRACTICES

The following are best practices for using the video plugins.

- You must load the Core Player and at least one video plugin for Player V4 to function properly.
- Generally, when using the [bit_wrapper.min.js](#) plugin, you should always load it first, before the other video plugins.
- You cannot use the [bit_wrapper.min.js](#) plugin with iOS. Please note that you can still include this plugin on the page (it just won't do anything). This means that you don't have to change which plugins you load based on device.
- To support FreeWheel ad playback, you must include the [main_html5.min.js](#) plugin.
- To support VPAID ad playback, you must load the [main_html5.min.js](#) plugin first among the video plugins.
- The [osmf_flash.min.js](#) plugin does not currently support Google IMA Ad Rules.
- For DASH streams on desktop web, we recommend using the [bit_wrapper.min.js](#) plugin with Chrome.
- For MP4 streams, we recommend using the [main_html5.min.js](#) plugin.
- If you would like to use HLS streams on desktop, we recommend that you use our [bit_wrapper.min.js](#) plugin on Chrome, Firefox and Internet Explorer. For Safari and Edge we encourage you to use the [main_html5.min.js](#) plugin instead.
- The [main_html5.min.js](#) plugin does not report bitrate information. If you want to track bitrate information (see [Analytics in Player V4](#) on page 189), you must use the Bitmovin plugin for DASH and HLS, OSMF Flash plugin for HDS (Deprecated), or the Akamai HD video plugin for Akamai packaged HDS (Deprecated).

STEP 1: LOAD SUPPORTED VIDEO PLUGINS

Note:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the Bitmovin plugin for HLS and DASH (`bit_wrapper.min.js`) and Main video plugin for HLS and MP4 (`main_html5.min.js`) are



automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#). You can also override the default plugins by specifying only the plugins you want to load (see [Specifying Plugins for the Standard Embed Code](#) on page 75).

- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add these plugins your web page so that they load before the player is created.

To load supported video plugins, create a `<script>` tag for each video plugin within the `head` element for your page. The order in which you load the video plugins helps determine which plugin will be used to decode video encodings.

See [Ooyala-hosted Player V4 Resources](#) on page 77 for a list of currently supported video plugins. The video plugins support HLS and MP4, Flash HDS, and HLS and DASH (with DASH you can only use audio codec formats AAC or mp4a.40.2 audio codecs for this release).

Warning: You must load the Core Player and at least one video plugin for Player V4 to function properly. If you do not load at least one video plugin, your player will not play videos.

Each tag must specify a `src` attribute that makes the request to load the plugin. The plugins are loaded immediately, so be sure to load the core player first.

Note: The order of the video plugins represents the order of priority for the plugin usage. For HTML5 video, we recommend loading the `bit_wrapper.min.js` first, followed by `main_html5.js`.

Here is the portion of the web page code that loads a plugin (`main_html5.min.js` in this example).

```
<head>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
        Advertising need to be loaded separately -->
    <script src="url\_where\_hosted/core.min.js"></script>
    <!-- A Video Plugin is required. This example shows the Main Video
        Plugin -->
    <script src="url\_where\_hosted/main_html5.min.js"></script>
    <!-- Change these html5-skin.min.css and html5-skin.js to your local
        build if necessary -->
    <script src="url\_where\_hosted/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url\_where\_hosted/html5-skin.min.css"/>
</head>
```

STEP 2: (OPTIONAL) SPECIFY VIDEO ENCODING PRIORITY

Having loaded the supported video plugins, you may optionally specify the encoding priority. As you learned in [Basic Tutorial for Player V4](#) on page 53, you can include embedded parameters. The `encodingPriority` embedded parameter may be used to specify the priority for video encoding.

The following are best practices for setting encoding priority and using video streams:

- The default encoding priority is `["dash_drm", "hls_drm", "dash", "hls", "mp4", "hds"]`. We strongly recommend that you use the default encoding unless you have other streaming needs.
- With DASH, you should always prioritize an alternate stream (do not set the encoding priority to only DASH).
- For DASH video, you must use the AAC or mp4a.40.2 audio codecs.

See an example of setting the encoding priority below:

```
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
```



```

        "encodingPriority": ["hls", "dash", "mp4", "hds"],
        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>
</body>

```

COMPLETE EXAMPLE

You now have a complete working example of a web page that loads a supported Ooyala V4 plugin, specifies the encoding priority, and creates the player. In the example below, items in bold must be modified for the example to play a video.

Wherever you see **url_where_hosted** in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

```

<!DOCTYPE html>
<html>
    <head>
        <script src="url_where_hosted/core.min.js"></script>

        <script src="url_where_hosted/main_html5.min.js"></script>

        <script src="url_where_hosted/html5-skin.min.js"></script>
        <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
    </head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "encodingPriority": ["hls", "dash", "mp4", "hds"],
            "skin": {
                // Config contains the configuration setting for player
                skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            }
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
        });
    </script>
</body>
</html>

```



FLASH VIDEO RENDERING

The Bitmovin plugin ([bit_wrapper.min.js](#)), OSMF Flash plugin ([osmf_flash.min.js](#), Deprecated), and Akamai HD plugin ([akamaiHD_flash.min.js](#), Deprecated) can decode and render video in a Flash-based element.

Considerations

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Note: All of the plugins, ad logic, UI, Player APIs, etc. of the player are always in HTML5. The only Flash component is video decoding and rendering if the selected encoding is not supported in HTML5 on a certain browser where Flash is supported. The encoding is selected irrespective of the technology used to render it (the encoding is selected first, then the plugins check to see which renderer to use).

Note: For cross-domain compatibility, in your Flash cross-domain policy file, be sure to grant access to both `http://` and `https://` URLs. For more information, see [Adobe's Cross-domain Policy documentation](#).

Note: For this release, the only way you can render ads with Flash is using VPAID 1.0 Google IMA ads and the `google_ima.min.js` plugin or VAST HLS ads on Chrome, FF, and IE or VAST DASH ads on Safari.

Bitmovin Plugin for DASH and HLS

You can use the Bitmovin plugin to decode and render DASH and HLS videos. This plugin will render in HTML5, if possible. If HTML5 rendering is not available on the given browser, the video will be rendered using Flash. Safari and Edge will play HLS natively. There are also some restrictions with the Bitmovin plugin, which are noted in [Ooyala-hosted Player V4 Resources](#) on page 77.

OSMF Flash Plugin for HDS (Deprecated)

You can use the OSMF Flash plugin to decode and render HDS videos. To enable Flash decoding and rendering of HDS videos with the OSMF plugin, you must do the following:

1. Load the [OSMF Flash plugin](#) on your page.
2. Set the `encodingPriority` so that `hds` is the highest available priority. If the encoding priority is not set, lower priority video encodings will be rendered.

Akamai HD Plugin for HDS (Deprecated)

You can use the Akamai HD plugin to decode and render Akamai packaged HDS videos. To enable Flash decoding and rendering of Akamai HDS videos with the Akamai HD plugin, you must do the following:

1. Load the [Akamai HD plugin](#) on your page.
2. For live streams, set the `encodingPriority` so that `akamai_hd2_hds` is the highest available priority. For VOD streams, set the `encodingPriority` so that `akamai_hd2_vod_hds` is the highest available priority. If the encoding priority is not set, lower priority video encodings will be rendered.

USING FLASH-FREE PLAYBACK WITH PLAYER V4

The `bit_wrapper` plugin for DASH and HLS ([bit_wrapper.min.js](#)) plays HLS using the HTML5 standard (and [Media Source Extensions](#)), enabling HLS on web browsers without the use of Flash. HTML5 MSE also supports playback of MPEG-DASH (clear and DRM-protected content).

If you use the `bit_wrapper` plugin, HTML5 playback is enabled by default. This plugin will render in HTML5 if possible. If HTML5 rendering is not available on the given browser, the video will be rendered using Flash.



IF YOUR SITE SERVES CONTENT VIA HTTP

If you serve your pages using *only* HTTP, you simply load this plugin on your page.

IF YOUR SITE SERVES CONTENT VIA HTTPS

If you serve *any* of your pages via HTTPS/SSL, you need to complete the following steps to ensure that all pages are served using HTTPS.

1. Contact Ooyala Technical Support to ensure that your environment is ready to support HTML5 Media Source Extensions (MSE)-based streaming.
2. For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, be sure to resolve all mixed content conditions. For example, if you are using remote streams (remote assets) with HLS, make sure that all of your remote streams (manifest/m3u8 and fragment/ts) are served via HTTPS.
3. If you are using any CDN other than Ooyala, make sure that the CORS setting is enabled from that CDN. For details, see [Cross-Origin Resource Sharing \(CORS\)](#) on page 88.

RENDERING WITH FLASH

If you prefer to lead with Flash-based playback, you need to add the following page-level parameter to `playerParam: { "platform": "flash" }`. For details, see the [platform](#) page-level parameter.

EMBEDDING PLAYER V4 IN AN HTML IFRA

You can use the HTML `<iframe>` ("inline frame") tag to embed content in a separately defined "container" of a web page. You can embed Player V4 in an `<iframe>`.

Note: Wherever you see `url1_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

GETTING THE IFRA

In Backlot, you can select the **V4 HTML5 Player iFrame** embed code type to copy and paste an automatically generated iFrame embed for your player on a web page. For details, see [Configuring Player Embed Settings in Backlot](#).

Note: This is the easiest way to insert a player in an iFrame on a web page. The rest of this topic provides background information in case you want to further customize your embedded player inside an iframe on a web page.

PLAYER V4 IFRA.HTML FILE

Ooyala provides a basic sample `<iframe>` page (`iframe.html`) that shows an example of using the basic player and skin in an `<iframe>`. It takes the values you specify in the `<iframe>` attributes and invokes the Ooyala Player in the inline frame. For the default location of this file, see [Ooyala-hosted Player V4 Resources](#) on page 77.

Note: You should not use this example out-of-the-box. Instead, customize the example with plugins you host and specify additional parameters.



EXAMPLE PLAYER V4 EMBEDDING FOR AN IFRA

The following example shows how you can embed the player in an iframe on a web page.

Note: The following example has been formatted for readability. In actual use, everything should be on a single line.

```
<iframe width="480" height="320" src="url_where_hosted/iframe.html?
  &ec=YOUR_ASSET_ID
  &pbid=YOUR_PLAYER_ID
  &pcode=YOUR_PCODE
  &optional_parameter_1=value
  &optional_parameter_2=value
  &optional_parameter_N=value"
  frameborder="0" allowfullscreen>
</iframe>
```

EXPLANATION OF <IFRA> ATTRIBUTES

- `width` and `height`: Modify these as desired to fit your web page, although the values shown are optimal.
- The value for `src`: Leave this as is, except for the protocol (`http` or `https`) and the values for `ec`, `pbid`, and `docUrl` (detailed below).
- Depending on the security of the web page where you will put this iframe, set the protocol to either `http` or `https`. Insecure iframes (served by `http`) in a secure web page (served by `https`) cause alarming error messages about security mismatches in a user's browser.

For more information about `ec`, `pbid`, and `pcode`, see [Basic Embedding Information](#) on page 88.

PLAYER PARAMETERS FOR <IFRA>

Parameter	Description of Value	Required?
<code>ec</code> or <code>embedCode</code>	The asset ID for the desired video. An asset ID is the same as a <code>ContentID</code> (in the Backlot UI) and an <code>embed_code</code> (in the Player V4 JavaScript API).	Yes
<code>pbid</code>	A "player branding ID" (or simply "player ID") for a player you have defined in Ooyala Backlot.	Yes
<code>pcode</code>	Your <code>pcode</code> , which is the alphanumeric string that precedes the period in your API key.	Yes
<code>docUrl</code>	This parameter can be used to specify the analytics URL of a video (must be URL encoded). This causes analytics events for the video to be grouped under the URL specified (rather than on the URL on which the player is placed).	No

PASSING EMBEDDED PARAMETERS TO THE PLAYER IN AN <IFRA>

Standard Ooyala Player syntax allows you to specify certain runtime arguments as the third argument on the player invocation, such as `autoplay:true`, `initialTime:122`, and other options detailed at [Embedded Parameters](#).

With Ooyala's `<iframe>` you can pass the same embedded parameters as part of the query. You simply preface each parameter name with an ampersand (&). You can add attributes and values as many times as you like. The only limit is the maximum length of a URL.



The following code example shows how to include the `autoplay` and `initialTime` parameters as part of the query. Be sure to URL-encode values that require it, such as URLs.

```
<iframe width="480" height="320" src="url_where_hosted/iframe.html?
  ec=YOUR_ASSET_ID
  &pbid=YOUR_PLAYER_ID
  &pcode=YOUR_PCODE
  &autoplay=true
  &initialTime=122"
  frameborder="0" allowfullscreen>
</iframe>
```

EXAMPLE: PASSING GOOGLE IMA AD TAGS

In the example below, the embedded parameters to set the Google Interactive Media Ads (IMA) `tagUrl` and other parameters are specified in the query for the iframe (see [Google IMA Ad Parameters](#) on page 157). Note that the iframe must be URL encoded.

```
<iframe width="480" height="320" src="url_where_hosted/iframe.html?
  ec=YOUR_ASSET_ID
  &pbid=YOUR_PLAYER_ID
  &pcode=YOUR_PCODE
  &autoplay=true
  &google-ima-ads-manager.all_ads=%22tag_url%22:%22someUrlThatIsUrlEncoded
  %22,
  %22position_type%22:%22t%22,
  %22position%22:%220%22}">
</iframe>
```

PLAYING YOUTUBE VIDEOS IN PLAYER V4 (DEPRECATED)

Note: Due to rapid changes that YouTube makes to its API and other functionality, maintaining this plugin at an acceptable quality level has not been achievable. Ooyala is therefore deprecating Player V4 playback of YouTube-hosted videos. If you have specific needs around YouTube videos with the Ooyala player, please contact Ooyala [Tech Support](#).

The Ooyala Player Plugin for YouTube iFrame enables you to play YouTube-hosted videos using Player V4 and the Player V4 skin. This allows you to syndicate video content from YouTube into your web application. Integration involves adding an asset in Backlot that points to the video hosted on YouTube. In Backlot, the asset type is "youtube", so in our documentation, we refer to these as *YouTube assets*. You can customize video metadata (video title and description) in Backlot so that it appears during playback.

SUPPORTED PLAYERS

The Ooyala Player Plugin for YouTube iFrame is supported in the Ooyala HTML5 web player versions 4.9.8 through 4.10.6 (but not later).

USAGE CONSIDERATIONS

- The Ooyala Player Plugin for YouTube iFrame uses YouTube's IFrame player API, which supports HTML5 playback only.
- The Ooyala Player Plugin for YouTube iFrame supports pre-recorded videos. Although live streams will work, the player experience does not reflect that the content is live.



AVAILABLE PLAYER V4 CONTROLS

You can use the following Player controls with Player V4 and YouTube-hosted videos:

- Play
- Pause
- Volume
- Social Sharing
- Discovery (if enabled in Backlot)
- Bitrate controls

In addition, playback events associated with a YouTube asset (represented by an embed_code in Backlot) are tracked using any of the Player V4 analytics plugins that implement the [Analytics Framework](#) on page 210.

REQUIREMENTS AND LIMITATIONS

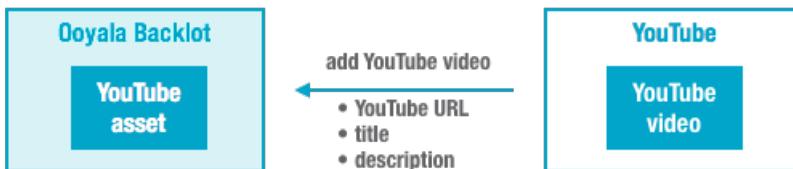
Because YouTube (not Ooyala) hosts the video content, playback in Player V4 is subject to the following requirements:

- Playing YouTube content is supported for HTML5 only. There is no Flash support.
- YouTube content cannot be played on iOS devices.
- Due to YouTube licensing requirements:
 - You cannot remove the YouTube watermark branding.
 - You cannot remove YouTube video ads.
 - To use Player V4 ad plugins (see [Ads in Player V4](#) on page 142) with YouTube-hosted videos, you must have written permission from YouTube, per their [Terms of Service](#). If you have this permission, contact Ooyala Technical Support for information on how to use Ooyala's Ad Manager Framework (AMF) with YouTube-hosted videos and Player V4.
- YouTube playlists are not supported within Player V4.
- YouTube preview images and thumbnails are not supported in Player V4.
- YouTube closed captions are not supported in Player V4.

INTEGRATION STEPS

Step 1: Add the YouTube Video in Backlot

Begin by adding the YouTube video to your Backlot account.



Note:

- When you add a YouTube video to Backlot, you simply create an asset that points to the video hosted on YouTube (its YouTube URL contains its YouTube ID). The video is not actually ingested into your Backlot account.
- There is a 1:1 correspondence between YouTube assets in Backlot and YouTube IDs. A YouTube ID is uniquely associated with one YouTube asset in Backlot. No two YouTube assets in your account can have the same YouTube ID.

For details, see:



- **Backlot GUI:** [Adding a YouTube Video](#)
- **Backlot API:** [Uploading a YouTube Video](#)
- **Video Tutorial:** [Adding New Content from YouTube](#)

Step 2: Associate the YouTube Video with the Player

Next, in Backlot, associate the embed code of the YouTube video with the Player. For details, see [Managing YouTube Asset Details](#).

Step 3: Specify the Plugin on the Web Page

Add the Ooyala Player Plugin for YouTube iFrame In the <head> section, load the Ooyala Player Plugin for YouTube iFrame.

```
<head>
  <title>My Test Player V4 Web Page</title>

  <script src="url_where_hosted/core.min.js"></script>

  <script src="url_where_hosted/html5-skin.min.js"></script>
  <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>

  <script src="url_where_hosted/video-plugin/youtube.min.js"></script>
</head>
```

Step 4: Embed the Player and YouTube Asset in an HTML iFrame

See [Embedding Player V4 in an HTML iframe](#) on page 95.

PAGE-LEVEL PARAMETERS FOR PLAYER V4

You can pass embedded parameters to the `OO.Player.create()` method. These parameters include CSS style settings such as width and height, and other parameters such as tags from your ad server or ad network account used for advanced ad tracking and targeting.

Required Parameters

- `pcode`
- `playerBrandingId`

Optional Parameters

- `adManagerLoadTimeout`
- `autoplay`

Important: Apple announced that it will start blocking unmuted autoplay of videos on its Safari 11 desktop browser. This could impact your desktop playback experience. See the following topics for details:

- [Notification: Autoplay for Safari 11](#)
- Example script to add pop-up message asking viewers to add your site to the allowed list for unmuted autoplay. [Autoplay for Safari 11: Script Example](#)
- `autoPlayUpNextVideosOnly`
- `encodingPriority`
- `initialBitrate`
- `initialTime`
- `initialVolume`
- `iosPlayMode`
- `location`



- *locationBaseUrl*
- *loop*
- *muteFirstPlay*
- *onCreate*
- *platform*
- *playerControlsOverAds*
- *playlistsPlugin*
- *preload*
- *skin.config*
- *skin.inline*
- *useFirstVideoFromPlaylist*

Note: To specify ad embedded parameters for Ooyala Pulse, Google IMA, VAST, VPAID, and FreeWheel, see [Configuring Ad Parameters](#) on page 180.

As you can see in the example below, you will add embedded parameters at the page level when you create a player.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Test Player V4 Web Page</title>
    <script src="//player.ooyala.com/core/YOUR_PLAYER_ID"></script>
  </head>
  <body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
      var playerParam = {
        // Add Optional Embedded Parameters Here
      };
      OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
      });
    </script>
  </body>
</html>
```

SYNTAX FOR EMBEDDED PARAMETERS

- Enclose parameter names in double quotes (""). Example: "autoplay"
- Enclose string values in double quotes. Example: "PCODE": "YOUR_PCODE"
- Omit quotes for Boolean values and numbers. Example: "autoplay": false
- For non-string parameter values, refer to the parameter example. Example: "initialTime": 10

REQUIRED PARAMETERS

The following are required embedded parameters, represented as key value pairs, that you must use when creating a V4 player.

pcode

The pcode is your account identifier. This is an alphanumeric string that precedes the period in your API key. You can get your [pcode](#) from your API keys. If you do not include your pcode, the player will not load.

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
```



```

<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>

```

playerBrandingId

The player branding ID is a reference to your player. You can get your player branding ID (referred to as the Player ID in Backlot) by going to the **MANAGE tab > Embed subtab** in Backlot. If you do not include your player branding ID, the player will not load.

Example:

```

<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>

```

OPTIONAL PARAMETERS

Note: Make sure that page-level parameters are added at the base level of the player params and not somewhere deeper in your code.

The following are optional embedded parameters, represented as key value pairs, that you may use with the `OO.Player.create()` method.

`adManagerLoadTimeout`

Specify the timeout (in seconds) for the loading of the Ad Manager module. The default is 3 seconds. To assist with ad-fill rate issues related to timeout settings, you can increase this value (for example, 4 or 5 seconds, depending on your page setup and load time).

Type: number

Example:

```

<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {

```



```

    "PCODE": "YOUR_PCODE",
    "playerBrandingId": "YOUR_PLAYER_ID",
    "adManagerLoadTimeout": 5,
    "skin": {
        // Config contains the configuration setting for player skin.
        Change to your local config when necessary.
        "config": "url_where_hosted/skin.json"
    }
};

OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>

```

autoplay

Enables the automatic playing of an asset (video or audio) upon loading. This is useful for UIs that do not have play/pause controls or conditions where you want the content to play immediately.

- 'autoplay': false (default) - disables automatic playback of the main video.
- 'autoplay': true - enables automatic playback for the main video.

Important: Autoplay on mobile browsers requires that the audio be muted. Therefore, if you set the player to autoplay on mobile browsers (Chrome and Safari Mobile), the video will play in a muted state until the user interacts with the video. Additionally, for Safari Mobile, autoplay requires setting an additional parameter ([iosPlayMode](#) to 'inline') to set playback to inline mode (the default is full-screen).

Note: Facebook does not allow autoplay of videos by third party players inside the Facebook feed. Therefore, autoplay settings will be ignored when the player is embedded on Facebook.

Type: Boolean

Example:

```

<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "autoplay": true,
        "skin": {
            // Config contains the configuration setting for player skin.
            Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
};
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>

```

autoPlayUpNextVideosOnly

Enables (default) or disables the automatic playing of Up Next or Discovery videos after the main video has played. Discovery will require users to initiate play events (no autoplay) if `playerParams.autoPlay` is false and `playerParams.autoPlayUpNextVideosOnly` is defined and is false. See [Discovering Content in Player V4](#) on page 134.

- 'autoPlayUpNextVideosOnly': false - disables automatic playback of Up Next or Discovery videos.



- 'autoPlayUpNextVideosOnly': true (default) - enables automatic playback. Up Next or Discovery videos will be played automatically after the main video has played - even if the main video does not automatically play ('autoplay': false).

Type: Boolean

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "autoplay": false,
        "autoPlayUpNextVideosOnly": false,
        "skin": {
            // Config contains the configuration setting for player skin.
            Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
    });
</script>
```

dynamicFilters

One or more predefined filter names for generating a dynamic manifest. See [Dynamic Manifests](#).

Type: String

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "autoplay": false,
        "dynamicFilters": "filterA,filterB,filterC",
        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container",
        "YOUR_ASSET_ID", playerParam);
    });
</script>
```

encodingPriority

Use this parameter to define the video encoding priority in a series of encodings separated by commas. The highest priority encoding that is available and can be decoded by the player will be selected. Any encoding that you do not specify will be appended to the end of the array in pseudo-random order.

Note: Use akamai_hd2_hds for Live streams and akamai_hd2_hds for VOD content.

The following are best practices for setting encoding priority and using video streams:



- The default encoding priority is ["dash_drm", "hls_drm", "dash", "hls", "mp4", "hds"]. We strongly recommend that you use the default encoding unless you have other streaming needs.
- With DASH, you should always prioritize an alternate stream (do not set the encoding priority to only DASH).
- For DASH video, you must use the AAC or mp4a.40.2 audio codecs.

Type: string

Valid Values:

- "hls"
- "dash"
- "mp4"
- "hds"
- "ima"
- "akamai_hd2_hds"
- "akamai_hd2_vod_hds"
- "dash_drm"
- "dash_hls"
- "AKAMAI_HD2_VOD_HLS"

Default: ["dash_drm", "hls_drm", "hls", "dash", "mp4", "hds"]

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "encodingPriority": ["hls", "dash", "mp4", "hds"],
        "skin": {
            // Config contains the configuration setting for player skin.
            // Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
    });
</script>
```

initialBitrate

Use this parameter to set the initial minimum bitrate level (immediately after video playback) and to sustain that level for a specific period of time. Once the duration is reached, the bitrate level changes to the video plugin's automatic bitrate level.

Settings:

You can specify two child parameters:

Name	Type	Valid Values	Description
level	string	One of the following: <ul style="list-style-type: none"> "auto" (default) number between 	<ul style="list-style-type: none"> Specify "auto" to defer to the default bitrate set by the video plug-in's ABR logic. The behavior is equivalent to omitting initialBitrate entirely. Specify a number if you want to set a particular level between zero and 1, inclusive. <p>If you specify a number:</p>



Name	Type	Valid Values	Description
	0 (zero) and 1, inclusive		<ul style="list-style-type: none"> Specify 0 (zero) to select the lowest bitrate first for the associated video plugin Specify 1 to select the highest bitrate (<code>MAX_BITRATE</code>) for the associated video plug-in. Specify a number between 0 and 1 to indicate which stream to use. From the array of available bitrates, the player selects the maximum value that is <i>less than or equal to</i> the preferred bitrate factor. For example, if your available bitrates were 0.1, 0.75, and 1.0Mbps, then a value of 0.5 would select the lowest bitrate (0.1), a value of 0.8 would select the middle bitrate (0.75), and so on. <p>To determine the level, consider such factors as the approximate network speed of the end user, the type of device (desktop or mobile), and so on.</p>
duration	number	a number greater than zero	Represents the number of seconds to sustain the initial bitrate level.

Note: The following APIs also override the `initialBitRate` and automatic bitrate, starting from when they are called and for the remainder of the playback session:

- `SET_TARGET_BITRATE`
- `setTargetBitrate()`

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
  var playerParam = {
    "PCODE": "YOUR_PCODE",
    "playerBrandingId": "YOUR_PLAYER_ID",
    "skin": {
      // Config contains the configuration setting for player skin.
      // Change to your local config when necessary.
      "config": "url_where_hosted/skin.json"
    },
    // Set the initial bitrate to 80% of the maximum bitrate for 30
    // seconds
    "initialBitrate": {"level": 0.8, "duration": 30}
  };
  OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
    playerParam);
  });
</script>
```

initialTime

Use this parameter to set an initial time in seconds to start playing content at a specific point. This parameter can be used to enable seeking for iOS-based devices.

Type: integer

Valid Values: time in seconds

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
  var playerParam = {
    "PCODE": "YOUR_PCODE",
```



```

    "playerBrandingId": "YOUR_PLAYER_ID",
    "skin": {
        // Config contains the configuration setting for player skin.
        Change to your local config when necessary.
        "config": "url_where_hosted/skin.json"
    },
    "initialTime": 10
};
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>

```

initialVolume

Use this parameter to set an initial volume for a video.

Type: number

Valid Values: 0-1

Example:

```

<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
            // Config contains the configuration setting for player skin.
            Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        },
        "initialVolume": 1.0
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>

```

iosPlayMode

Use this parameter to specify the initial playback mode for Safari Mobile on iOS devices (iOS 10 and later). By default, playback is initially in full screen mode ("iosPlayMode": "fullscreen"). You can change this to inline playback (playing videos inline on the page rather than in full screen mode) by specifying "iosPlayMode": "inline" on the web page.

Note: Inline playback is required for skippable Google IMA ads. In addition, autoplay on mobile browsers requires that the audio be muted. See the note under [autoplay](#) for Safari mobile.

location

Use this parameter (within a "bit-wrapper" object) to specify individual, custom paths for Bitmovin HTML5, CSS, and Flash files if any of those resources are stored in a location *other than* where the [bit_wrapper.min.js](#) plugin is stored.

Note: Use this parameter only if you are using the bit_wrapper.min.js plugin and *only* if you do any of the following:

- You self-host player resources *and* you want to serve one or more of the associated resource files from a location that differs from where the bit_wrapper.min.js plugin is served.



- You self-host player resources and you concatenate all of the Player V4 Javascript files into a single bundle.
- You load the Player V4 files using a module loader library (most notably SystemJS) that bundles Javascript files. For SystemJS, this is true regardless of where the bitmovin resource files are located (custom or default).

Otherwise, this parameter is not needed.

Note: This parameter is identical to Bitmovin's `Location` parameter, which is described in [Bitmovin's documentation](#). Player V4 currently uses only the `html5`, `css`, and `flash` options. Other options (`vr`, `ctrls`, or `ctrls_css`) are currently not used.

Type: object

Valid Values:

The following options are supported.

Status	Description
html5	Location of <code>bitmovinplayer-core.min.js</code> . Path (fixed or relative) plus filename.
css	Location of <code>bitmovinplayer-core.min.css</code> . Path (fixed or relative) plus filename.
flash	Location of <code>bitmovinplayer.swf</code> . Path (fixed or relative) plus filename.

If not specified, then the default location is used (the location of the `bit_wrapper.min.js` plugin) or, if provided, the location specified by the `locationBaseUrl` parameter (described below).

Notes:

- You can specify URLs using HTTP or HTTPS. However, to avoid mixed content issues, Ooyala recommends using the double slash instead (//).
- When using both `locationBaseUrl` and `location` at the same time, `location` has priority over `locationBaseUrl`. Any values specified in `location` will override those generated with `locationBaseUrl`. Any files omitted from `location` will use the location generated with `locationBaseUrl`.

Example

```
{
  "bit-wrapper": {
    "locationBaseUrl": "//test.com",
    "location": {
      "html5": "//my.custom.url/my-custom-file.js"
    }
  }
}
```

The above settings would resolve to the following configuration:

```
"location": {
  "html5": "//my.custom.url/my-custom-file.js",
  "css": "//test.com/bitmovinplayer-core.min.css",
  "flash": "//test.com/bitmovinplayer.swf",
  "vr": "//test.com/bitmovinplayer-vr.min.js",
  "ctrls": "//test.com/bitmovinplayer-controls.min.js",
  "ctrls_css": "//test.com/bitmovinplayer-controls.min.css"
}
```



Important: Player releases are associated with specific Bitmovin versions and are compatible only with those versions. When self-hosting, remember to download the Bitmovin files for the matching version using the exact static path for that version.

locationBaseUrl

Use this parameter (within a "bit-wrapper" object) to specify a custom path for the resources (Bitmovin HTML5, CSS, and Flash files) associated with the [bit_wrapper.min.js](#) plugin if they are not stored in the same location as the bit_wrapper.min.js plugin. Whereas the location parameter lets you specify an individual path for each resource file, the locationBaseUrl parameter allows you to specify a single path for all resource files (Bitmovin HTML5, CSS, and Flash files) at once.

Note: Use this parameter only if you are using the bit_wrapper.min.js plugin and *only* if you do any of the following:

- You self-host player resources *and* you want to serve one or more of the associated resource files from a location that differs from where the [bit_wrapper.min.js](#) plugin is served.
- You self-host player resources and you concatenate all of the Player V4 Javascript files into a single bundle.
- You load the Player V4 files using a module loader library (most notably SystemJS) that bundles Javascript files. For SystemJS, this is true regardless of where the bitmovin resource files are located (custom or default).

Otherwise, this parameter is not needed.

Notes:

- You can specify a fixed or relative path.
- The url should be the full path without the filename and without a trailing slash.
- The url can be set with either http or https, but Ooyala recommends using double slash instead (//).
- If you specify this parameter, default file names will be used unless overridden with the location parameter.

Type: string

Example

```
{ "bit-wrapper": {  
    "locationBaseUrl": "//test.com"  
}
```

The above settings would resolve to the following configuration:

```
{  
    "html5": "//test.com/bitmovinplayer-core.min.js",  
    "css": "//test.com/bitmovinplayer-core.min.css",  
    "flash": "//test.com/bitmovinplayer.swf",  
    "vr": "//test.com/bitmovinplayer-vr.min.js",  
    "ctrls": "//test.com/bitmovinplayer-controls.min.js",  
    "ctrls_css": "//test.com/bitmovinplayer-controls.min.css"  
}
```

Important: Player releases are associated with specific Bitmovin versions and are compatible only with those versions. When self-hosting, remember to download the Bitmovin files for the matching version using the exact static path for that version.

loop

Use this parameter to enable continuous play. With loop set to true, once the playback has started it continues until the user stops playback or closes the browser. Also the behavior is the same when the ASSET_ID is set using setEmbedCode. As soon as the ASSET_ID is set, if autoplay is true, the



playback starts immediately regardless of the previous state of the player (video playing/paused/stopped). If `autoplay` and `loop` parameters are not passed in through `setEmbedCode`, the existing values are used (which may have been set via a previous call to `setEmbedCode`).

Type: Boolean

Example:

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
            // Config contains the configuration setting for player skin.
            // Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        },
        "loop": true
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
    });
</script>
```

`muteFirstPlay`

Use this parameter to control audio when a video starts. Behavior can be affected by the `autoplay` setting:

- `"muteFirstPlay":true` causes the video to start muted.
- `"muteFirstPlay":false` causes the video to start unmuted.

Note: If `"autoplay":true` and the browser requires muted autoplay, then the video will play muted even if `"muteFirstPlay":false`.

- `"muteFirstPlay":true` and `"autoplay":true` causes the video to autoplay muted

Type: Boolean

`onCreate`

Use this parameter to listen to an event message and perform an action. This parameter enables you to subscribe to event messages from the message bus before the player is fully created. It allows you to modify the player prior to its complete creation.

When called, `onCreate: function(player)`:

- Checks for any additional modules (custom, 3rd party or other type).
- Enables these additional modules to connect to the message bus.
- Sends a message to the message bus signaling each module to start up.

You must call `onCreate` before anything can happen; otherwise, the existing and additional or third-party modules are not connected to the message bus and are not initialized.

Example

```
<div id="container" style="width:640px; height:360px;"></div>
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
```



```

        // Config contains the configuration setting for player skin.
        Change to your local config when necessary.
        "config": "url_where_hosted/skin.json"
    },
    "onCreate": function(player) {
        player.mb.subscribe("*", "myPage", function(eventName) {});
    }
};
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>

```

onCreate and the Player Callback

To handle events in Player V4, you provide an `onCreate` function to the `OO.Player.create()` call, and then register for all the messages.

`platform`

This parameter applies only to the [bit_wrapper.min.js](#) plugin for DASH and HLS (`bit_wrapper.min.js`). As of Player v4.13.4, HTML5 playback is used as the default. However, if you prefer to lead with Flash-based playback, you need to add the following page-level parameter to `playerParam`: `{"platform": "flash"}`.

Note: If you specify `"platform: "flash"` to make Flash the priority, and Flash is disabled in a user's browser, then playback might fail for that user.

Example

```

<script>
var playerParam = {
    "PCODE": "YOUR_PCODE",
    "playerBrandingId": "YOUR_PLAYER_ID",
    "platform": "flash",
        // Add Optional Embedded Parameters Here
    };
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID", playerParam);
});
</script>

```

`playerControlsOverAds`

Use this parameter to enable (`true`) the Ooyala Player control bar during ads (via the new `playerControlsOverAds` page-level parameter). Defaults to `false`.

Caution: This capability should be used *only* for when VPAID ads have no interactivity. If this option is used in combination with ads that include user interaction (navigation links within the ad), they may conflict with the player control and prevent user from navigating the video.

Type: Boolean

Example:

```
{"playerControlsOverAds":true}
```

`playlistsPlugin`

Use this parameter to hide the Playlist plugin but still have it drive the autoplay a sequence of videos. By default, the Playlist plugin is displayed. To hide it, set `hideUi` to `true`.

Type: Boolean



Example:

```
{  
    "playlistsPlugin": {  
        "hideUi": true  
    }  
}
```

preload

Use this parameter to specify whether to start loading the video after the player is loaded and before the user starts playback (for non-autoplay settings) (`true`), or to wait until the user starts playback (`false`, the default). One of the following values:

- `'preload': false` (default)
- `'preload': true` - start preloading the video as soon as the player is loaded. Typically, a video will preload the stream up to filling the buffer, the size of which will vary by browser.

Preloading can significantly speed up the time-to-first-frame experience because the video has the opportunity to buffer before the viewer plays it.

Considerations:

- Preloading is disabled if the `initialTime` parameter is used.
- When using Google IMA with prerolls, preloading will be triggered at the start of the fourth quartile of the preroll ad or the last ad in a podded preroll. If ads are skipped before the fourth quartile of the ad is played, preloading of content will not occur.
- Preloading is supported for the `main_html5` and `bit_wrapper` video plugins.
- Preloading is not yet supported for the Pulse or Freewheel ad managers.
- Enabling content preloading can increase your overall video stream consumption because, for any given player embed, videos are preloaded even when playback is not initiated.

Type: Boolean

Example:

```
<div id="container" style="width:640px; height:360px;"></div>  
<script>  
    var playerParam = {  
        "PCODE": "YOUR_PCODE",  
        "playerBrandingId": "YOUR_PLAYER_ID",  
        "preload": true  
    };  
    OO.ready(function() {  
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",  
        playerParam);  
    });  
</script>
```

skin.config

The skin parameter references the `skin.json` config file. This loads the player skin. Use it if you want to use your own skin settings.

Example:

```
<div id="container" style="width:640px; height:360px;"></div>  
<script>  
    var playerParam = {  
        "PCODE": "YOUR_PCODE",  
        "playerBrandingId": "YOUR_PLAYER_ID",  
    };  
</script>
```



```

        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>

```

`skin.inline`

Use this parameter to specify inline skin modifications. `skin.inline` will overwrite any settings in the `skin.json` config file. The JSON object within `inline` must have the same structure as `skin.json` (all parent objects going all the way back to the root object, as described in the [Player V4 JSON Schema](#)). For example, if you want to overwrite the start screen play button color using inline, you need to include the start screen object, `playIconStyle` object, and `color` (as shown in the following example).

Example:

```

<div id="container" style="width:640px; height:360px;"></div>
<script>
var playerParam = {
    "PCODE": "YOUR_PCODE",
    "playerBrandingId": "YOUR_PLAYER_ID",
    "skin": {
        "config": "url_where_hosted/skin.json",
        "inline": {
            // Put your player customizations here
            // to override settings in skin.json.
            // The JSON object must match the structure of
            skin.json.
            "startScreen": {"showDescription": false,
"playIconStyle": {"color": "blue"}}
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
    });
</script>

```

For additional information, see [Customizing the Player V4 Skin with skin.json](#) on page 120 and [Embed a Simple Player with Inline Player Skin Modifications](#).

`useFirstVideoFromPlaylist`

Use this parameter to specify whether the player uses the first video from the playlist `playlist` (`true`) or not (`false`, the default). One of the following values:

- `'useFirstVideoFromPlaylist': false` (default)
- `'useFirstVideoFromPlaylist': true` - let the player set the first asset specified on the first item in a playlist.

Note: Before you set `'useFirstVideoFromPlaylist': true`, contact Ooyala Tech Support to have your player plugin registered. If the plugin is not registered, playback will fail.

See [Using Playlists in Player V4](#) on page 136 for details.

Type: Boolean



EXAMPLES OF PLAYER V4 WEB PAGE EMBEDDING

This topic provides examples of how you embed Player V4 on a web page. Embedding involves decisions on which plugins you want to load, along with the degree to which you want to customize the skin and CSS.

Note: Wherever you see *url_where_hosted* in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

EMBED A PLAYER WITH A VIDEO

To create a player with a video, add the content ID for the video. You can get the content ID from your Backlot account. For more information, see [Publishing a Video](#).

```
<!DOCTYPE html>
<html>
    <head>
        <title>My Test Player V4 Web Page</title>
        <script src="//player.ooyala.com/core/YOUR_PLAYER_ID"></script>
    </head>
    <body>
        <div id="container" style="width:640px; height:360px;"></div>
        <script>
            OO.ready(function() {
                window.pp = OO.Player.create("container", "YOUR_ASSET_ID");
            });
        </script>
    </body>
</html>
```

This uses the Backlot configuration of the player associated with **YOUR_ASSET_ID**. It includes the following plugins:

- skin-plugin/
 - html5-skin.min.css
 - html5-skin.min.js
- video-plugin/
 - main_html5.min.js
 - bit_wrapper.min.js
- ad-plugin/
 - pulse.min.js
 - google_ima.min.js
 - freewheel.min.js
 - ad_manager_vast.min.js
- other-plugin/
 - discovery_api.min.js



EMBED A SIMPLE PLAYER WITH INLINE PLAYER SKIN MODIFICATIONS

The following example shows how to use the inline parameter to make inline player skin modifications. skin.inline will overwrite any settings in the skin.json config file. The JSON object within the `skin.inline` page-level parameter must have the same structure as skin.json (all parent objects going all the way back to the root object). For example, if you want to overwrite the start screen play button color using inline, you need to include the start screen object, playIconStyle object, and color, as shown in the following code.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Test Player V4 Web Page</title>
    <script src="//player.oyala.com/core/YOUR_PLAYER_ID"></script>
  </head>
  <body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
      var playerParam = {
        "skin": {
          "inline": {
            //Put your player customizations here to override
            settings in skin.json. The JSON object must match the structure of
            skin.json
            "startScreen": {"showDescription": false,
              "playIconStyle": {"color": "blue"}}
            }
          }
        };
        OO.ready(function() {
          window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
          playerParam);
        });
      </script>
    </body>
  </html>
```

For additional information, see [Customizing the Player V4 Skin with skin.json](#) on page 120.

EMBED A PLAYER WITH OPTIONAL EMBEDDED PARAMETERS

For a more advanced use of the Player embed style, you can create a player with optional embedded parameters. Add parameters within the `playerParam` variable. See [Page-level Parameters for Player V4](#) on page 99 for a full list of embedded parameters.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Test Player V4 Web Page</title>
    <script src="//player.oyala.com/core/YOUR_PLAYER_ID"></script>
  </head>
  <body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
      var playerParam = {
        // Add Optional Embedded Parameters Here
      };
      OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
      });
    </script>
  </body>
</html>
```



```
</script>
</body>
</html>
```

EMBED A PLAYER WITH EXTERNAL IDENTIFIERS

For a more advanced use of the Player embed style, you can create a player using external identifiers. The following example creates a player using an external Id.

```
<script>
    var playerParam = {
        "skin": {
            // Config contains the configuration setting for player
            skin. Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        }
    };
    OO.ready(function() {
        window.pp =
    OO.Player.create("container", "extId:<name:numeric_id>", "YOUR_ASSET_ID",
    playerParam);
    });
</script>
```



CUSTOMIZING THE PLAYER V4 APPEARANCE

CUSTOMIZABLE COMPONENTS

The following web, mobile web, and mobile SDK player UX components are customizable:

Player Skin Color Accent Highlights	Controls Closed Captions Localization	Accessibility Show/Hide Custom icons Metadata on UI
Branding Logos Watermarks	State Screens Start Pause End	Engagement Discovery Social Sharing

WAYS TO CUSTOMIZE

You can customize the UX of your player in the following ways:

- **Backlot.** Configure Player V4 using the Backlot GUI. For details, see [Configuring a Player in Backlot](#).
- **Modify the JSON config file (HTML5, iOS SDK, and Android SDK):** Modify the `skin.json` config file to make basic UX customizations, such as player controls, state screens, branding, etc. Host your own customized version of `skin.json` and reference it when you load your player. You can use the same `skin.json` file across your HTML5, Android SDK, and iOS SDK players. For the default location of this file, see [Ooyala-hosted Player V4 Resources](#) on page 77. Go to [Skin Schema](#) and click **properties** to view the JSON schema reference documentation for `skin.json`. See [Customizing the Player V4 Skin with skin.json](#) on page 120 for details on where to look in the API documentation for each player component you want to customize.

If you are customizing the player through the `skin.json` config file, all you need to understand is HTML and CSS.

- **Modify CSS Settings (HTML5):** Make advanced interface customizations by modifying the CSS (such as the `html5-skin.min.css` file). For details, see [Customizing the Player V4 Appearance Using CSS](#) on page 122.
- **Use embedded parameters on the page (HTML5 only):** Override skin settings when you create the player through embedded parameters on the page level. For example, if you wanted to hide the description text and render the play button blue on the start screen of your HTML5 player, you would add the following parameters to your page:

```
var playerParam = {
  "skin": {
    "config": "//url_where_hosted/skin.json",
    "inline": {
```



```
        "startScreen": { "showDescription": false, "playIconStyle": { "color": "blue" } }
    }
}
};
```

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

See the section "Create a Simple Player with Inline Player Skin Modifications" in [Examples of Player V4 Web Page Embedding](#) on page 113 for more details on how to make inline customizations.

- **Fork and modify the source code. (HTML5, iOS SDK, and Android SDK):** Fork and modify the source code to make advanced customizations such as adding custom fonts and images.

Warning: Ooyala will support any pre-defined and valid modifications to the skin.json file. If you fork the skin-config repository and make non-valid modifications, or if you fork the ios-skin or html5-skin repositories and make changes, we will not offer technical support for these changes.

- For HTML5, advanced customization is readily available by modifying JS files. Follow the section [Developer Setup](#) of the README file on GitHub to create a local repository and to run a build script. Built files are available inside the `build` folder. You are welcome to host your built skin JavaScript to be run with your player.
- For iOS, follow the section [How to Perform Complex Modifications to the Sample Application](#) of the README on GitHub.
- For Android, follow the section [How to Perform Complex Modifications to the Sample Application](#) of the README on GitHub.

If you are forking the skin code and making more complex customizations, you need to understand HTML, CSS, JavaScript, and the Ooyala approach to using [React Native](#). For web players we use the standard React Native implementation. For iOS and Android players, we have implemented custom Ooyala React components.

For more information on Ooyala and React, see [Player V4 Skin and the React Application Framework](#) on page 125.

ORDER OF PRECEDENCE

Where the same configuration setting (for example, player background color) is controlled by different mechanisms, one mechanism prevails and appears to the viewer during playback. See [Order of Precedence in Player V4 Settings](#) on page 118 for more information.

END USER PERSONALIZATION

End users can personalize their players by selecting settings for the following features.

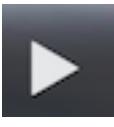
- Set closed captions to on or off.
- Set autoplay to on or off.
- Select a bitrate.

These settings will persist for the user after they make their selections the first time. The user information is tracked via cookies and/or device storage.



BEST PRACTICES

For the optimal end user experience with your brand, we recommend, at a minimum, that you update the following components of the default player. For more best practices to follow when customizing your player, see <http://community.oyala.com/t5/Ooyala-Player-V3-Knowledge-Base/Best-Practices-for-Customizing-Player-V4/ta-p/8440>.

Player Component	Default Value	How to Customize
Brand colors	white 	You can change the title font color on the different player screens by changing the color values in the titleFont>color elements in skin.json. You can change the color of the icons in the control bar by updating the color values in the controlBar >iconStyle element in skin.json. To change the color of the play button on the startScreen, update the color value in the startScreen>playIconStyle element in skin.json.
Logo	Ooyala logo 	Image resources are located in html5-skin/assets/images. You can specify which image is used as the watermark image on the player by updating the imageResource path specified in the watermark element in skin.json.

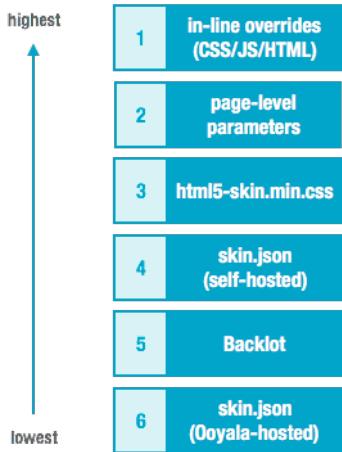
HOW TO CUSTOMIZE

ORDER OF PRECEDENCE IN PLAYER V4 SETTINGS

Player V4 provides many options for customizing the player experience. Where the same configuration setting (for example, player background color) is controlled by different mechanisms, it is important to understand which mechanism prevails and appears to the viewer during playback. See [Customizing the Player V4 Appearance](#) on page 116 for background.

The following figure shows which settings take precedence in priority order (1 = highest priority).





CONFIGURATION OPTIONS

You can configure playback experience using the following mechanisms.

Mechanism	More Information
Player Skin (<code>skin.json</code>)	<ul style="list-style-type: none"> Customizing the Player V4 Skin with <code>skin.json</code> on page 120 Localizing the Player V4 UI on page 124 Adding a Custom Branding Overlay to Player V4 on page 132
Backlot	<ul style="list-style-type: none"> Configuring a Player in Backlot Discovering Content in Player V4 on page 134
CSS	<ul style="list-style-type: none"> Customizing the Player V4 Appearance Using CSS on page 122 Customizing Player V4 Controls with Graphics on page 124
Page-level parameters	<ul style="list-style-type: none"> Page-level Parameters for Player V4 on page 99 Configuring Ad Parameters on page 180 Analytics in Player V4 on page 189 Using Playlists in Player V4 on page 136
Inline overrides on the web page (CSS/JS/HTML)	Skin and CSS overrides

DEFAULT SETTINGS AND OVERRIDES

Every mechanism has default settings that you can override by changing their default values.

PRECEDENCE DEPENDS ON WHERE SKIN RESOURCES ARE HOSTED

Where player settings overlap across mechanisms, precedence is conditional based on where the player skin is hosted.

- If Ooyala hosts your `skin.json` file, then the Backlot setting has priority over the settings in `skin.json`. See [Ooyala-hosted Player V4 Resources on page 77](#).
- If you host and modify your `skin.json` file, then the `skin.json` file has priority. See [Hosting Player V4 Resources on page 81](#).



For example, if the default player color is orange in Backlot and blue in skin.json, the prevailing color (during playback) depends on where skin.json is hosted.

- If Ooyala hosts your skin.json file, then the viewer will see orange.
- If you host your skin.json file, then the viewer will see blue.
- If you have no settings anywhere, the core default will set a color.

CUSTOMIZING THE PLAYER V4 SKIN WITH SKIN.JSON

In addition to using the Backlot UI, you can make customizations to the Player V4 and the Android and iOS skin UX by modifying the key/value pairs in the JSON config file (skin.json). You can use the same skin.json file for your HTML5, Android SDK, and iOS SDK players. To learn more about customization approaches, see [Order of Precedence in Player V4 Settings](#) on page 118.

Note: For Player V4 Web, rather than customize skin.json directly, you can simply override skin settings on the web page using the `skin.inline` page-level parameter. For details, see [Page-level Parameters for Player V4](#) on page 99 and [Embed a Simple Player with Inline Player Skin Modifications](#).

CUSTOMIZATION STEPS

To use skin.json to customize the player look and feel:

1. Download the latest version of the skin.json config file:
 - desktop and mobile web: <https://github.com/ooyala/html5-skin/releases>
 - iOS and Android mobile SDKs: <https://github.com/ooyala/native-skin/releases>
2. *Host* the skin.json config file on your server.
3. *Make modifications* to the JSON attributes to apply look and feel changes to your player.
4. Reference the absolute path to your skin.json file when you *load your player*.

Note: Customize skin.json only if you want to change configuration settings in the file. Alternatively, you can simply use the default settings. Alternatively, if you want to override some skin settings, you can use the `skin.inline` parameter when you embed Player V4 on a web page. See [Page-level Parameters for Player V4](#) on page 99 for details.

SETTINGS TO CUSTOMIZE IN THE SKIN.JSON FILE

To view the reference documentation for the skin.json file, go to [Player V4 JSON Schema](#). The following table shows the path to the documentation for each customizable player component. See [User Interface for the Player V4 Skin](#) on page 11 for a description of the player components.

Player Component	Path to Documentation	Notes
General	properties > General	Watermark, loading image, and accent color.
Responsive Design	properties > responsive	For videos with an aspect ratio where the height of the video is greater than the width or the width is much greater than the height, we recommend that you use our default responsive design setting (aspectRatio:auto).
Start Screen	properties > startScreen	Start screen settings.
Pause Screen	properties > pauseScreen	Pause screen settings.



Player Component	Path to Documentation	Notes
End Screen	properties > endScreen	End screen settings.
Up Next Screen	properties > upNext	Up Next screen settings.
Ad Screen	properties > adScreen	You can also use the ads embedded parameters described in Configuring Ad Parameters on page 180 to further customize the ad screen.
Social Sharing Options	properties > shareScreen	Share screen settings.
Player Watermark	properties > general > watermark	Used to specify the watermark image.
More Options Screen	properties > moreOptionsScreen	More Options screen settings.
Discovery Screen	properties > discoveryScreen	<p>Any Discovery settings you make in Backlot will be overwritten by the settings specified in skin.json.</p> <p>Note: You must have Discovery enabled in Backlot for Discovery recommendations to appear. If Discovery is not enabled, the Discovery icon will not appear in the control bar or more options menu.</p> <p>Depending on the type of embed code you use (see Configuring Player Embed Settings in Backlot):</p> <ul style="list-style-type: none"> • If you are using the V4 HTML5 Player Embed Code (Advanced), you must load the Discovery module (<code><script src="url_where_hosted/discovery_api.min.js"></script></code>) in the <code><head></code> of the page where you load your player. • If you are using the V4 HTML5 Standard Player Embed Code, the plugin is automatically loaded by default.
Closed Caption Options	properties > closedCaptionOptions	You can configure closed captions for VOD (all streams) and Live (HLS on Safari) content. By default, the player will show a closed caption button for all assets with closed captions. To enable closed captions for a video, simply upload your closed



Player Component	Path to Documentation	Notes
Player Buttons	properties > buttons	caption file for your asset. See <i>Uploading a Closed Caption File in Backlot</i> and <i>Ingesting Closed Caption Files</i> .
Player Icons	properties > icons	Icon settings.
Player Control Bar	properties > controlBar	Control bar settings.
Player Localizable Strings	properties > localization	For this release, Ooyala offers localized player controls and error messages into Spanish and Simplified Chinese.

CUSTOMIZING THE PLAYER V4 APPEARANCE USING CSS

You can use Cascading Style Sheets (CSS) to customize the appearance of your Ooyala Player V4 for HTML5.

WHEN TO USE CSS TO CUSTOMIZE THE PLAYER

Player V4 provides a default CSS file (`html5-skin.min.css`) with detailed settings that define the Player V4 user interface. These CSS settings are in addition to the `skin.json` file (see *Customizing the Player V4 Skin with skin.json* on page 120) that you can use to change mostly Player V4 behavioral settings. The JSON settings do not overlap the CSS settings.

WHAT YOU CAN CUSTOMIZE

The CSS allows you to configure Player V4 display settings (such as fonts, colors, icons, and measurements) for:

- player elements (control bar, scrubber bar, buttons, spinner, slider, video quality)
- state screens (start, pause, share, error)
- ads (ad screen, overlays, panel)
- discovery (screen, up next panel, countdown clock)
- closed captions (screen, options)
- responsive design (width, minimum width, height)
- other settings (base settings, plug-ins, and so on)

Note: Before you make CSS changes, you should have expertise in working with CSS files.

WHERE TO LOOK

Ooyala generates the `html5-skin.min.css` file using the Sass (<http://sass-lang.com/>) (SCSS) for CSS preprocessor and modular SCSS files stored in the source repository (<https://github.com/ooyala/html5-skin/tree/stable/scss>). This folder includes source files for:

- base elements (including typography rules, fonts, icons, colors, measurements, enable/disable flags, etc.)



- Player V4 components (buttons, forms, screens, other components)
- Player V4 skin (default values pulled from /skin-plugin/config/skin.json)

Refer to the README (<https://github.com/ooyala/html5-skin/blob/stable/README.md>) for the latest description of the source files. For example, if you wanted to learn which CSS settings are associated with the scrubber bar, you could look at https://github.com/ooyala/html5-skin/blob/stable/scss/components/_scrubber-bar.scss

WAYS TO CONFIGURE CSS SETTINGS

You can approach CSS configuration in several ways.

Make a Copy the Default File

You can copy the default html5-skin.min.css file (see Hosting Ooyala Player V4 for HTML5 Resources), make changes, host it, and the point to it from the web page on which you initialize the Player V4. For example:

```
<link rel="stylesheet" href="url_where_hosted/my_custom.css"/>
```

Append Your Changes Using a Separate CSS

You can also make a separate CSS file that overrides or appends settings. For example:

```
<link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
<link rel="stylesheet" href="url_where_hosted/my_custom_overrides.css"/>
```

The order on the page determines the winner when identical elements have different values. Specify the CSS file with the overrides after the default CSS on the HTML page. Where the elements are the same, the settings in the latter file prevail. For example:

html5-skin.min.css	my_custom_overrides.css
<pre>.innerWrapper .oo-control-bar { -moz-user-select: none; background: rgba(0, 0, 0, 0.3) none repeat scroll 0 0; font-size: 18px; height: 60px; list-style: outside none none; margin: 0; position: absolute; transition: bottom 0.5s ease 0s, height 0.25s ease 0s; width: 100%; z-index: 12500; }</pre>	<pre>.innerWrapper .oo-control-bar { background: rgba(0, 0, 0, 0.9) none repeat scroll 0 0; height: 50px; }</pre>

Embed Inline Settings on the HTML Page

You can override CSS settings on an HTML page from which you launch Player V4. To specify in-line CSS settings in your code, add a `<style>` tag immediately below the `<link>` tag that points to the CSS file. For example, to change the color of the start screen text:

```
<link rel="stylesheet" href="//myserver.com/html5-skin.min.css"/>

<style type='text/css'>
    .oo-player_skin .oo-startScreen .oo-state-screen-title
        Unknown macro: { color}
</style>
```



Modify the Source html5-skin SCSS Files

Although more involved, you might prefer to build your own CSS using a local copy of <https://github.com/oyala/html5-skin/blob/stable/README.md>, modifying the source SCSS files, building the html5-skin.min.css file, hosting the built file, and point to it from the web page on which you initialize the Player V4.

CUSTOMIZING PLAYER V4 CONTROLS WITH GRAPHICS

You can replace the default playback controls (buttons) with custom graphics by modifying the player CSS.

- Modify `_type.scss` to include a graphic rather than use a font file.

For example, you would modify

```
.icon-play:before  
{ content: "h"; }
```

to look like the following:

```
.icon-play:before  
{ content: url("myPlayIcon.png"); }
```

- To adjust the size of the graphic, style psuedo elements by inserting whitespace as content and setting a background image and style instead.

For example:

```
.icon-play:before  
{ content: " "; background: url("./assets/images/icon-play.png");  
background-size: 30px; background-size: contain; width: 30px; height:  
30px; display: inline-block; }
```

This method works for flattened images, or svg. The important piece is content: " ".

LOCALIZING THE PLAYER V4 UI

You can localize the Player V4 (HTML5) and Android and iOS player skins. The default `skin.json` file included in the example on Github references default localization files for the player.

```
"localization": {  
  "defaultLanguage": "en",  
  "availableLanguageFile": [  
    {  
      "language": "en",  
      "languageFile": "",  
      "androidResource": "skin-config/languageFiles/en.json",  
      "iosResource": "en"  
    },  
    {  
      "language": "es",  
      "languageFile": "",  
      "androidResource": "skin-config/languageFiles/es.json",  
      "iosResource": "es"  
    },  
    {  
      "language": "zh",  
      "languageFile": "",  
      "androidResource": "skin-config/languageFiles/zh.json",  
      "iosResource": "zh"  
    }  
  ]  
}
```



```

        "androidResource": "skin-config/languageFiles/zh.json",
        "iosResource": "zh"
    }
    {
        "language": "ja",
        "languageFile": "",
        "androidResource": "skin-config/languageFiles/ja.json",
        "iosResource": "ja"
    }
]

```

The localization files contain a list of translated phrases. For example, [es.json](#) contains the following JSON data:

```
{
    "Learn More" : "Más información",
    "CLOSED CAPTION PREVIEW": "VISTA PRELIMINAR DE SUBTÍTULOS",
    "Sample Text": "Texto de muestra",
    ...
}
```

If you [host](#) your own customized copy of the `skin.json` config file, you can also host your own localization files and reference them from `skin.json` as shown above. See [Customizing the Player V4 Skin with skin.json](#) on page 120 for details about skin customization.

PLAYER V4 SKIN AND THE REACT APPLICATION FRAMEWORK

If you plan to fork the skin code and make more complex customizations, you need to understand how Ooyala integrates with Facebook's React application framework.

REACT AND WEB APPLICATIONS

Ooyala Player V4 for HTML5 uses the standard [React](#) implementation for building dynamic, single-page web applications. Player V4 implements React's code structure with the controller and components. The controller, which provides the main logic, does all network calls, state management, and grabbing of items from the database. The controller passes on the state to other relevant components. Components are abstracted elements that have high degrees of functionality. They consist of a UI element and relevant logic. Examples include the HTML player controller bar, start screen, and widgets. For an introduction, see React's [Hello World](#) documentation.

REACT NATIVE AND MOBILE APPLICATIONS

Ooyala Player Skin for iOS and Ooyala Player Skin for Android use [React Native](#) for mobile native applications. For an introduction, see [Getting Started](#) with React Native.

CLOSED CAPTIONS IN PLAYER V4

Closed captions are additional text shown within a video during playback. The captions typically contain an audio transcript of the video. For an overview, see [Managing Closed Captions](#)



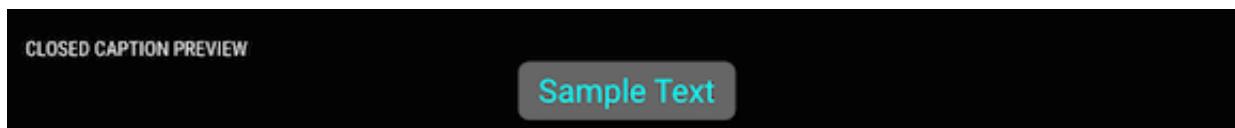
USER EXPERIENCE WITH CLOSED CAPTIONS

If closed captions are available for a video, end users can click the CC button on the Player V4 control bar to open the closed captions screen for all browsers with web and mobile platforms. In the closed captions screen end users can enable or disable closed captions in the language of their choice and apply color, opacity, and font styling settings.

In Player V4, closed captions consist of the text, background, and window. The window is a rectangle that contains the background, which contains the text.



From the closed captions screen end users can preview what the closed captions will look like on the player before applying settings. Note that if closed captions are off, when you make a selection in the GUI, for example, selecting a language, closed captions will automatically turn on with that setting applied.



CONFIGURATION OPTIONS IN PLAYER V4

By default, if closed caption files are available for a video, they will be enabled and shown in English. You can configure the default behavior and language of closed captions for your player in skin.json by modifying the defaultEnabled and defaultLanguage attributes of the closedCaptionOptions object.

SUPPORTED INPUT FORMATS

Closed caption files should be uploaded to Backlot along with your video files. For details on how to upload closed caption files to Backlot, see [Ingesting Closed Caption Files](#) and [Uploading a Closed Caption File in Backlot](#).

The following input formats are supported for Player V4 closed captions:

- DFXP
- VTT
- CEA-608 (Ooyala Live only)
- SCC

Note: Any styling settings specified in the closed caption file will be ignored.

FCC COMPLIANCE

Player V4 is now Federal Communications Commission (FCC) compliant and supports FCC closed caption styling per the Twenty-First Century Communications and Video Accessibility Act ([CVAA](#)) requirements. FCC compliant closed caption styling allows the end user to change how the closed captions appear on the player, including:



- Multi-language support
- Font
- Text size
- Text, background, and window color
- Text, background, and window opacity
- Text styling

If closed captions are available for a video, the end user has the ability to apply the styling mentioned above.

Note: FCC closed caption styling is only supported on the web Player versions 4.5+ for desktop web and mobile web. The native SDKs do not have any styling settings within the Ooyala Player. For Android, the Android Skin SDK uses the operating system's closed caption styling information to render closed captions. As of June 23, 2016, the iOS Skin SDK does not support FCC closed caption styling. We are working on adding this functionality. The Android Skin SDK supports FCC closed caption styling.

STORING CLOSED CAPTION SETTINGS

With the closed caption screen UI controls, end users can make changes to closed caption styles on the fly. Player V4 saves closed caption settings automatically in the browser. If a user changes a setting (for example, chooses a different font color), the change remains in effect when they open a new browser tab or window, or launch a new browser session. The information is saved using local storage key "ooyala_player_settings". The OO.EVENTS.SAVE_PLAYER_SETTINGS event is triggered when a change is saved.

ENABLING OR DISABLING CLOSED CAPTIONS

To turn closed captions on, the end user clicks **CC button > On**. The player displays closed captions immediately after the end user clicks **CC button > On** in the default closed caption language. The default closed caption language is English, unless otherwise specified in skin.json.



To turn closed captions off, the end user clicks **CC button > Off**.

CONFIGURING LANGUAGE AND CAPTION SETTINGS

In the popover, click **Caption Options** to change the language and other settings for closed captions.

SELECTING A LANGUAGE

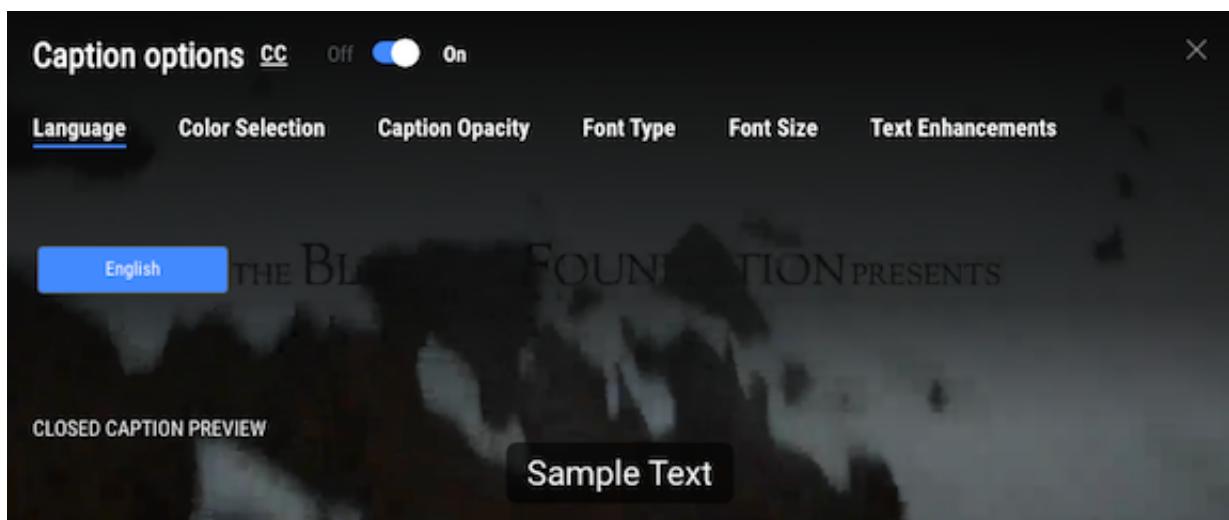
If you have associated closed captions with your video, end users will be able to select a caption language from the languages you provided.

To customize the closed caption language, the end user clicks **Language** and selects the desired closed caption language.

Note: The entry for each language selection in the closed caption screen is localized in its own language. For example, Italian is listed as Italiano.

The default language is English, unless otherwise specified with the defaultLanguage attribute of the closedCaptionOptions object in skin.json.



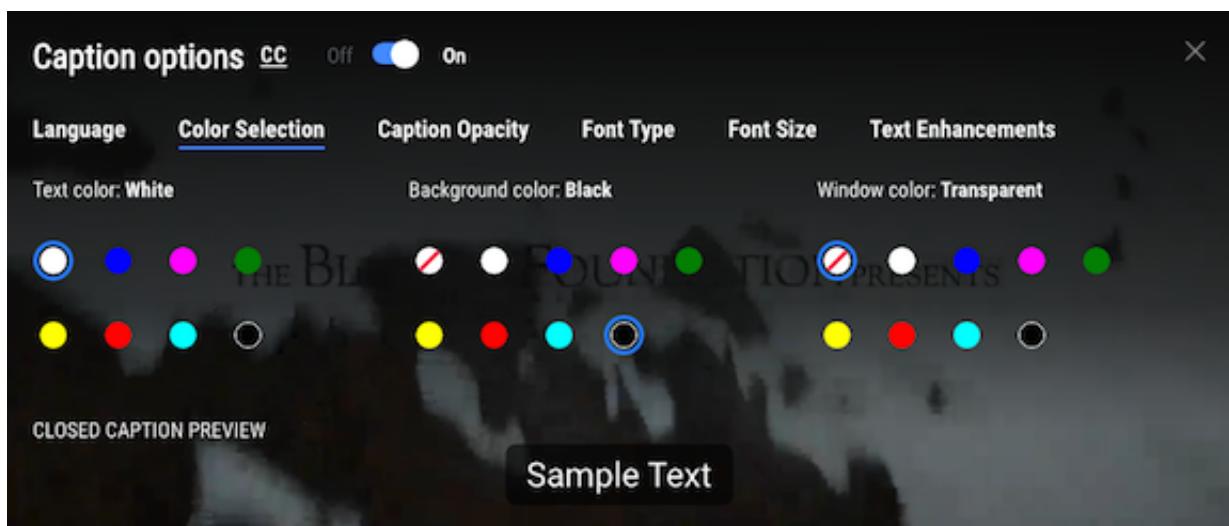


SELECTING TEXT, BACKGROUND, AND WINDOW COLORS

For the closed caption text, end users can select a color option to apply.

For the closed caption background and window, end users can select a color option or transparent option to apply.

To customize colors, end users click **Color Selection** and select a color. The default color settings are white text with a transparent window and a black background.

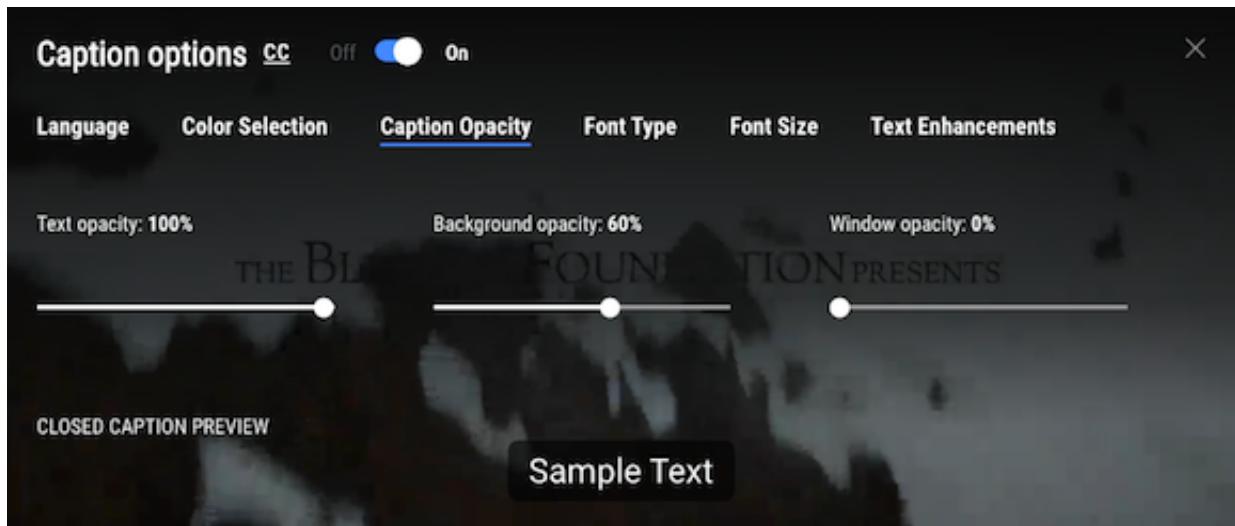


SELECTING TEXT, BACKGROUND, AND WINDOW OPACITY

For the closed caption text, background, and window, end users can use the slider bars to select an opacity. The default opacity settings are 100% opacity for text, 0% opacity for the window, and 60% opacity for the background.

To view opacity options, end users click **Caption Opacity**.

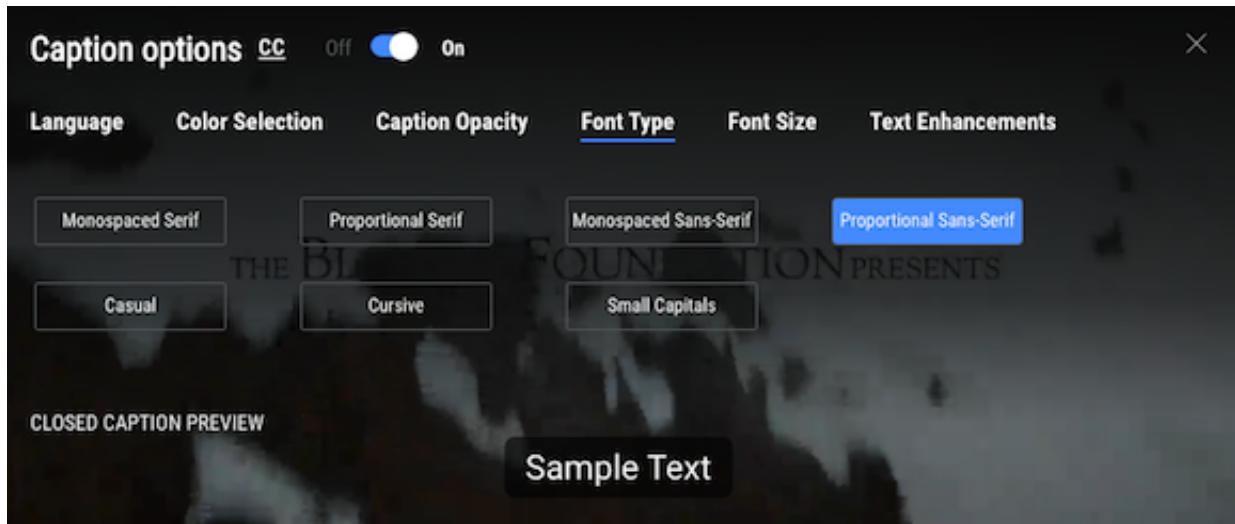




SELECTING A FONT TYPE

End users can choose from a selection of font types. All Font types shown are web safe fonts that browsers can render automatically.

To customize the font type, end users click **Font Type** and select a font. The default font is Proportional Sans-Serif.

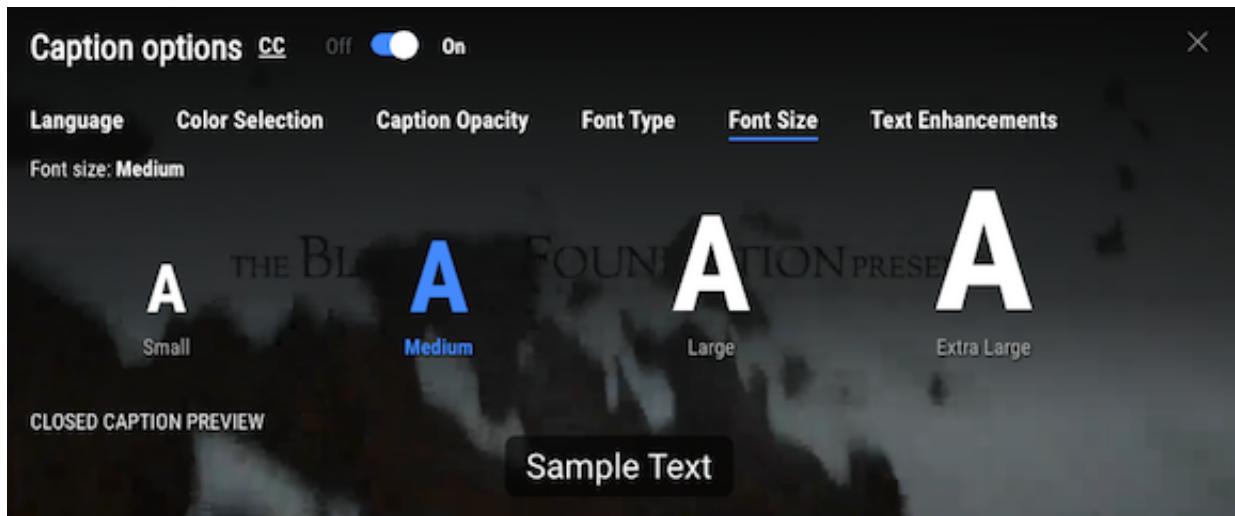


SELECTING A FONT SIZE

End users can select from small, medium, large, and extra large font sizes.

To customize the font size, end users click **Font Size** and select a size. The default size is the second text size icon (medium).



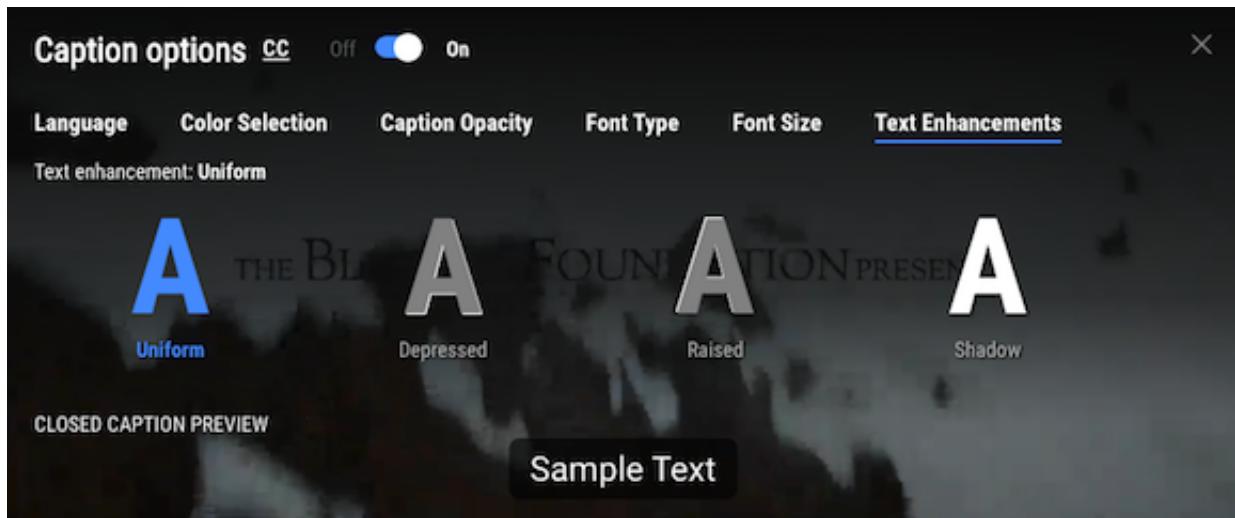


APPLYING TEXT ENHANCEMENTS

End users can apply the following text enhancements to closed captions. Only one enhancement can be applied at a time.

- Uniform
- Depressed
- Raised
- Shadow

To apply the text enhancements, end users click **Text Enhancements** and select an option.



KNOWN ISSUES

When using HLS video assets with Mac devices and the Safari browser, the closed caption button always displays even if closed caption files are not associated with the video asset. This is due to an intrinsic problem with Safari.



CUSTOMIZING VR 360 PLAYER CONTROLS

You can configure several parameters in the skin.json file to customize Ooyala player controls that are used to navigate VR 360 videos. For more information, see [Customizing the Player V4 Skin with skin.json](#) on page 120 and [Player V4 JSON Schema](#).

ON-SCREEN CAMERA ANGLE CONTROL

The on-screen Camera Angle Control allows the viewer to navigate in VR 360 using a mouse to click on the arrows to direct movement.



You can customize this control using the following settings in skin.json.

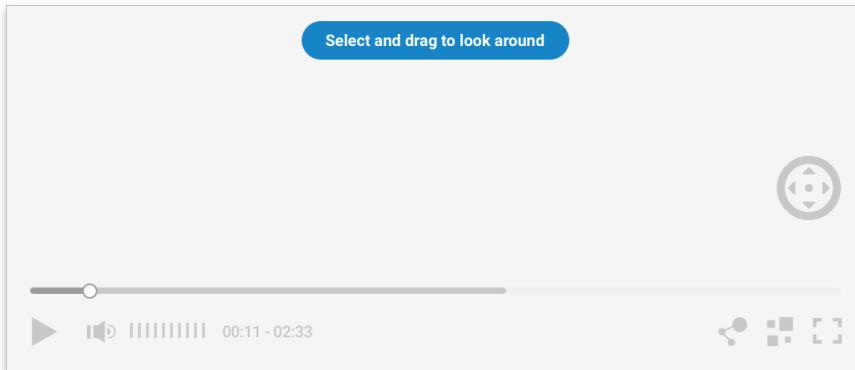
```
{ "name": "arrowsBlack", "location": "mainView", "whenDoesNotFit": "keep",  
  "minWidth": 45 }
```

Considerations:

- This control is available only on desktop browsers (where mouse devices are used). It is disabled on mobile devices (including mobile browsers) and touch-screens, even if it is specified in skin.json. For example, the "arrowsBlack" control, if specified in skin.json, is ignored on mobile platforms and touch screens.
- If you want to exclude this control from Player V4 web during VR 360 playback, add an override inline to remove this element from the skin.json.

VR 360 VIDEO NAVIGATION HELPER

The following message appears briefly upon starting playback and advises the viewer to navigate during VR 360 video by using either a mouse to click and drag it or touch and drag for touchscreens.



Note: This control is available only on desktop browsers (where mouse devices are used). It is disabled on mobile devices (including mobile browsers) and touch-screens, even if it is specified in skin.json.

To show the message and/or icon, specify the following settings in json.config

```
"isVrAnimationEnabled": {  
    "vrNotification": true,  
    "vrIcon": false  
}
```



Both (message and icon) appear with fade in / fade out animation. The length of animation can be managed through the additional settings:

```
"animationDurations": {  
    "vrNotification": 5,  
    "vrIcon": 4  
}
```

To display the VR 360 icons and control, you must specify these settings in the skin.json file. To hide these controls, simply remove these lines from skin.json. Also, if in stereoscopic mode, ads will be displayed, but if the ads are flat, the viewer will not be able to view the VR 360 video properly. Ads are typically flat currently.

STEREOSCOPIC ICON

The Stereoscopic icon allows the viewer to switch between Monoscopic and Stereoscopic modes.



You can customize this control using the following settings in skin.json.

```
{ "name": "stereoscopic", "location": "controlBar", "whenDoesNotFit": "keep",  
  "minWidth": 45 }
```

Considerations:

- This should be left as disabled if you do not want to allow stereoscopic viewing of the VR 360 content using binocular head-mounted displays (HMDs), such as Google Cardboard.
- This option is appropriate only for screens that can be used with a binocular head-mounted display. Do not use this toggle option if the viewer will view the VR 360 content with a monocular HMD or on a desktop or mobile device without binocular lenses.
- The Stereoscopic icon is disabled for Web desktop and for tablets. Even if the "stereoscopic" parameter is specified in skin.json, these controls won't be visible on Web desktop and for tablets (only appropriate for devices that can be used with head-mounted displays). However, note that this means the user may not realize the content is VR 360. If this icon is disabled, we recommend that the thumbnail make it clear that the video is VR 360.

ADDING A CUSTOM BRANDING OVERLAY TO PLAYER V4

You can configure branding overlays on stream (formerly known as "watermarks") in Backlot (PUBLISH > Player Branding > Branding), as described in [Configuring a Player in Backlot](#). You can also override Backlot settings (location, transparency, click URL, and scaling) using CSS and skin.json settings, as described below. Images used for branding overlays must be in the PNG, JPG, or JPEG format.

Player V4 supports showing custom branding overlays during playback, where the image URL is specified in skin.json and inline page parameter configurations. To add a custom branding overlay in Player V4:

1. Configure the following values in skin.json, which must be locally hosted to configure your own .PNG file:

```
"watermark": {  
    "imageResource": {  
        "url": "watermark.png",  
        "androidResource": "logo",  
        "iosResource": "logo"  
    },  
    "clickUrl": "http://www.oyala.com/" ,
```



```
    "width": 177,  
    "height": 44  
},
```

2. To add the branding overlay as a new div in the playing screen, you must customize the `html5-skin.json` file. Follow the setup instructions at <https://github.com/ooyala/html5-skin>. Once this is complete your `sample.html` will be running at <http://0.0.0.0:4444/>.
3. Add the following code to `playingScreen.js`, which customizes the appearance of the screen while the asset is playing:

```
// Under the required variables, add this line:  
classNames = require('classnames')  
  
// Add this clickthrough handler:  
handleWatermarkClick: function() {  
    var watermarkClickUrl =  
        this.props.skinConfig.controlBar.watermark.clickUrl;  
    if (watermarkClickUrl) {  
        window.open(watermarkClickUrl, '_blank');  
    }  
}
```

4. In the `render` function, add the following variable to properly add the class to the branding overlay div and to the click URL configured in `skin.json`:

```
var watermarkClass = classNames({  
    "oo-playeScreen-watermark": true,  
    "oo-non-clickable-watermark": !  
this.props.skinConfig.controlBar.watermark.clickUrl  
});
```

5. Look for this div:

```
<div className="oo-state-screen-selectable"  
      onMouseUp={this.handlePlayerMouseUp}  
      onTouchEnd={this.handleTouchEnd}  
></div>
```

Right after that div, add the new branding overlay div:

```
<div className={watermarkClass}>  
<img src={watermarkUrl} onClick={this.handleWatermarkClick}/></div>
```

6. After saving, the `gulp` system should detect the changes and build a new `html5.skin.js` under the build folder.



DISCOVERING CONTENT IN PLAYER V4

Discovery enables you to keep users engaged by directing them to related and relevant content on your site. Discovery learns continuously from video impressions and plays to provide users with a dynamic, personalized experience.

Using Discovery, you can recommend:

- Videos based on content popularity or velocity for the current hour, day, week, or month.
- Videos with usage patterns that are similar to a selected video.

ENABLING DISCOVERY FOR YOUR ACCOUNT

Discovery must be explicitly enabled for your Ooyala Backlot account. If it is not already enabled, contact your Ooyala account representative to activate Discovery features for your site.

CONFIGURING DISCOVERY

You configure discovery settings using the Discover tab in Backlot. For instructions, see [Configuring Discovery in the Backlot UI](#).

ADDING THE DISCOVERY PLUGIN TO AN HTML PAGE

Note: Discovery and Playlists (created using ThemeBuilder) each control end screen behavior, so you cannot use them together. When you embed a player on a web page, you can use *either* Discovery or Playlists, but not both.

To use Discovery with Player V4:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the Discovery plugin is automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).
- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add the Discovery plugin ([discovery_api.min.js](#)) to your web page so that the Discovery plugin loads before the player is created.

Discovery functionality requires the `playerBrandingId` you specify on the page.

CUSTOMIZING THE DISCOVERY USER INTERFACE

In addition to using Backlot (see [Configuring Discovery in the Backlot UI](#)), you can customize the Discovery interface by modifying settings in the `skin.json` config file or in CSS properties.

Discovery settings in skin.json

You can change Discovery interface settings in `skin.json`:

- To customize visual elements in the Discovery screen (such as the title), change settings under `discoveryScreen`.
- To display the Discovery screen when playback ends, under the `endScreen` settings, specify `"screenToShowOnEnd": "discovery"`.
- To display the Discovery screen when playback pauses, under the `pauseScreen` settings, specify `"screenToShowOnPause": "discovery"`.
- To configure Discovery buttons, under `buttons`, change the `discovery` button settings for `desktopContent` and `mobileContent`.



- To configure the Discovery icon, under `icons`, change the discovery icon settings.

For more information, see [Customizing the Player V4 Skin with skin.json](#) on page 120.

Discovery settings in CSS

You can customize the Discovery interface by modifying settings under the `discoveryScreen` element. See [Customizing the Player V4 Appearance Using CSS](#) on page 122 for instructions.

CUSTOMIZING VIDEO RECOMMENDATIONS

You can change settings that Ooyala's video recommendations engine uses to generate recommendations for consumers. See [Configuring Discovery in the Backlot UI](#) for instructions.

DISCOVERY AND AUTOPLAY

- Beginning with Player V4 version 4.12.6, Discovery defaults to autoplay, even when the main video is not set to autoplay on the page. To disable the automatic playing of Up Next or Discovery videos after the main video has played, set `playerParams.autoPlay` to false and specify `playerParams.autoPlayUpNextVideosOnly` as false. See [Page-level Parameters for Player V4](#) on page 99.
- For Player V4 version 4.11.14 and earlier, Discovery defaulted to autoplay only if the embedded player was set to autoplay.

FOR MORE INFORMATION

To learn more about Discovery, see:

- [Discovery REST APIs](#)
- [Discovery FAQ](#)



USING PLAYLISTS IN PLAYER V4

You can create playlists for Player V4 using Ooyala Theme Builder. Some differences exist in the way you configure and embed playlists for Player V4 and Player V3. For detailed information about Theme Builder, see [Playlists and the Theme Builder User Interface](#).

Note: Playlist pods are not supported in this release.

SUPPORTED PLAYER VERSIONS

Playlist integration with Ooyala Theme Builder is supported in Player V4 version 4.10.4 or later.

SUPPORTED BROWSERS

Playlist integration runs on all supported Player V4 browsers (see [Supported Browsers and Operating Systems for Player V4](#) on page 22).

SUPPORTED THEME BUILDER UI SETTINGS IN PLAYER V4

The following table describes which settings in the Theme Builder user interface are supported in Player V4.

Theme Setting Tab	Player V4	Player V3	Notes
Playlist tab	✓	✓	Fully Supported
Players tab			
Choice of player	✓	✓	Required. This selection defines the <code>PLAYER_ID</code> specified in the embed code.
Setting subtab		✓	Not supported. Configure Player V4 skin settings instead. See Customizing the Player V4 Skin with skin.json on page 120.
Controls subtab		✓	Not supported. Configure Player V4 skin settings instead.
Screens subtab		✓	Not supported. Configure Player V4 skin settings instead.
Pods subtab	✓	✓	Supported. See the following settings.
Type	✓	✓	Supported. Corresponds to the <code>orientation</code> page-level parameter.
Overflow	✓	✓	Supported. Corresponds to the <code>podType</code> page-level parameter.
Position	✓	✓	Supported. Corresponds to the <code>position</code> page-level parameter.
			Note: If specified as a page-level parameter, this setting overrides the orientation setting.
Thumbnail Size	✓	✓	Supported. Corresponds to the <code>thumbnailsSize</code> page-level parameter.
Thumbnail Spacing	✓	✓	Supported. Corresponds to the <code>thumbnailsSpacing</code> page-level parameter.



Theme Setting Tab	Player V4	Player V3	Notes
Embed tab	Font size	✓	Supported. Corresponds to the <code>wrapperFontSize</code> page-level parameter.
	Playlist elements	✓	Supported. Corresponds to the <code>caption</code> page-level parameter.
	Caption position	✓	Supported. Corresponds to the <code>captionPosition</code> page-level parameter.
Settings			
Auto Play	✓	✓	Supported. Corresponds to the <code>autoplay</code> page-level parameter. See Page-level Parameters for Player V4 on page 99.
Auto Repeat	✓	✓	Supported. Corresponds to the <code>loop</code> page-level parameter. See Page-level Parameters for Player V4 on page 99.
Player Size		✓	Not supported. Configure Player V4 skin settings instead.
Width		✓	Not supported. Configure Player V4 skin settings instead.
Generate Embed Code	✓	✓	You will use portions of the generated V3 embed code when you embed Player V4 format on a web page.

INTEGRATING A PLAYLIST WITH PLAYER V4

1. Create and configure the playlist in Theme Builder. See [Working with Playlists in Theme Builder](#) and [Working with Players and Pods in Theme Builder](#) (refer to the table above for supported settings).
2. In Theme Builder, generate the embed code (Player V3 version). See [Working with Embeds in Theme Builder](#) (refer to the table above for supported settings).
3. On the web page where you load your player, add the Playlists plugin (`playlists.js`) to the `<head>` section, somewhere after you specify `core.min.js`. See [Ooyala-hosted Player V4 Resources](#) on page 77.

Note: Discovery and Playlists (created using ThemeBuilder) each control end screen behavior, so you cannot use them together. When you embed a player on a web page, you can use *either* Discovery or Playlists, but not both.

```

<html>
  <head>
    <title>My Test Player V4 Web Page</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
        Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <!-- Change these html5-skin.min.css and html5-skin.js to your
        local build if necessary -->
    <script src="url_where_hosted/html5-skin.min.js"></script>
  
```



```

        <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css" />
        <!-- A Video Plugin is required. This example shows the Main Video
    Plugin -->
        <script src="url_where_hosted/main_html5.min.js"></script>
        <!-- Playlists Module -->
        <script src="url_where_hosted/playlists.js"></script>
    </head>
    ...
<html>
```

- On the web page where you load your player, use specific portions of the generated embed code for Player V4 embedding. For example, if the generated Player V3 embed code is:

```

<div id="playerContainer"></div>
<script type="text/javascript" src="//player.ooyala.com/v3/[PLAYER_ID]"></script>
<script type="text/javascript">
var ooyalaPlayer;
OO.ready(function() {
    var playerConfiguration = {
        playlistsPlugin: {"data": "[PLAYLIST_ID]"}, 
        autoplay: false,
        loop: false,
        height: 357,
        width: 800,
        useFirstVideoFromPlaylist: true
    };
});
```

You would use the following portions in your Player V4 page embed script:

```

var playerConfiguration = {
    "playlistsPlugin": {"data": "[PLAYLIST_ID]"}, 
    "autoplay": false,
    "loop": false,
};
```

You must also use the `PLAYER_ID` that is assigned here. A working playlist must have both the proper `PLAYER_ID` and its matching `PLAYLIST_ID` assigned or you will get an error for 'metadata' not found. The `PLAYLIST_ID` assigns the list of videos, but the layout, size, and look of the playlist is attached to the `PLAYER_ID`. The `PLAYER_ID` must match the setting for the `playerBrandingId` specified on the web page.

Note: Ignore everything else in the generated Player V3 embed code.

- If you want, customize the playlist further by configuring the associated skin settings. See [Customizing the Player V4 Skin with skin.json](#) on page 120.

PLAYLIST PAGE-LEVEL PARAMETERS

Settings specified in Theme Builder override the playlist default settings. Settings specified in the following page-level parameters override settings specified in Theme Builder.

Note: Page-level parameter names are case sensitive.

Required Parameters

Description	Default
<code>parentParameter</code> for playlists.	Object
<code>PLAYLIST_ID</code> from the generated embed code for this playlist.	string



Optional Parameters

Name	Description	Type	Default
caption	<p>Contents of the caption for a video. You can specify any combination of title, description, and duration, or specify "none" to omit captions. Examples:</p> <ul style="list-style-type: none"> • "caption": "title,description,duration" • "caption": "title,duration" • "caption": "none" (disables the default and displays no captions) <p>For each video, the values for the specified caption options come from Backlot. The values will be placed with each thumbnail.</p>	string	"title"
captionPosition	<p>Position of the caption. One of the following values:</p> <ul style="list-style-type: none"> • "inside" places the thumbnail information over the thumbnail. • "outside" places the thumbnail information in a separate box to the side, above, or below the thumbnail. 	string	"inside"
orientation	<p>Playlist orientation. One of the following values:</p> <ul style="list-style-type: none"> • "vertical" • "horizontal" 	string	"horizontal"
podType	<p>Pod type. If orientation: "horizontal", determines how the player handles clicking the right and left arrows. One of the following values:</p> <ul style="list-style-type: none"> • "scrolling" moves one video at a time • "paging" moves one screen full of thumbnails ("page") 	string	"scrolling"



Name	Description	Type	Default
position	This setting has no effect if orientation: "vertical". Playlist position. One of the following values: <ul style="list-style-type: none">• "left"• "right"• "top"• "bottom"	string	"bottom"
thumbnailsSize	Size (in pixels) of the thumbnail image. Must be smaller than the size of the video or the playlist will fail.	number	150
thumbnailsSpacing	Spacing (in pixels) between thumbnail images.	number	3
useFirstVideoFromPlaylist	<ul style="list-style-type: none"> • If true, allows a playlist to be loaded and have its first video set as the page's initial embedCode, rather than the actual embedCode placed on the page. • If false (the default), the actual embedCode placed on the page is used. <p>Note: If set to false, and if the first video specified (on player creation) is not in the playlist, then the player will play that video first, and then start the playlist. In this case, the viewer will not be able to go back to play the initial video.</p> <p>It's optional and is either true or false. For example, if you had a playlist loaded into an already existing page, you could use this parameter to allow the playlist to immediately replace the currently loaded / playing video. If a user is browsing collections of videos and</p>	Boolean	false



Name	Description	Type	Default
	chooses one of them, they would expect the first video of the selected collection to start playing (or at least be loaded right away) without performing additional clicks.		
wrapperFontSize	Size of the wrapper font.	number	14



ADS IN PLAYER V4

Note: You can use only one ad plugin per Ooyala player, including My Ads. Serving ads from multiple ad plugins within the same Ooyala player is not a supported player configuration.

To deliver ads during playback, use any of the ad plugins that integrate with Player V4. To deliver My Ads from Backlot, see [Delivering My Ads](#) on page 184 instead.

AD PLUGINS

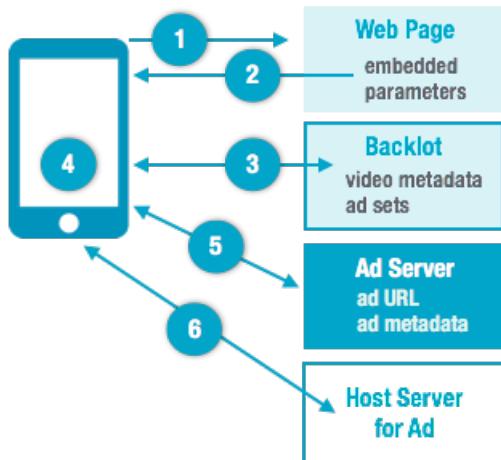
You can use the following ad plugins to deliver ads in Player V4.

- [FreeWheel Ad Plugin](#) on page 143
- [Google IMA Ad Plugin](#) on page 147
- [Ooyala Pulse Ad Plugin](#) on page 160
- [VAST and VPAID Ad Plugin](#) on page 171

These plugins were developed using the [Ad Manager Framework](#) on page 186. For more information, see [About Ads](#) on page 186 and [Integrating Ads](#) on page 176.

Note: If you are migrating from Player V3 to Player V4, see [Migrating to Player V4](#) on page 57.

RUN-TIME PLAYBACK AD FLOW FOR AD PLUGINS



1. When the HTML page is opened, the Ooyala player is created and loads.
2. The player reads in all embedded parameters on the page.
3. The player obtains from Backlot any metadata about the video and any ad sets, if defined.
4. The player starts playing the video.
5. During playback, if a cue point is reached (for example, for a pre-roll, mid-roll, or post-roll ad), the player obtains information about the ad or overlay (such as the URL where it's hosted, and other metadata) from the applicable ad server.
6. The player pulls the ad or overlay from the specified URL, and then plays the ad.

ADS AND FIREFOX PRIVATE BROWSING

For consumers using [Private Browsing windows](#) in Firefox, some ads may fail to load because privacy mode blocks certain kinds of tracking URLs.



FREEWHEEL AD PLUGIN

Use the `freewheel.js` ad plugin to load FreeWheel ads in Player V4. FreeWheel is an ad server and ad management platform. Using FreeWheel, publishers are able to segment their ad inventory, define ad targeting rules, and take advantage of the FreeWheel MRM network capabilities. The Ooyala Ad Manager Controller (part of the [Ad Manager Framework](#) on page 186) handles when ads play and how video ads are rendered for FreeWheel ads. FreeWheel controls overlay ads.

SUPPORTED PLATFORMS

Ooyala supports the FreeWheel ad manager on the following platforms for Player V4 for this release:

- HTML5 web
- iOS mobile SDK
- Android mobile SDK

SUPPORTED AD POSITIONS

Ads served through the FreeWheel ad server (VAST XML) are supported on V4 HTML5 players and the iOS and Android mobile SDKs for the following ad positions:

- Pre-Roll
- Mid-Roll
- Post-Roll
- Skippable
- Clickthrough
- Podded
- Overlay

ADS AND FIREFOX PRIVATE BROWSING

For consumers using [Private Browsing windows](#) in Firefox, some ads may fail to load because privacy mode blocks certain kinds of tracking URLs.

Integrating FreeWheel Ads

You can integrate FreeWheel ads with Player V4 using the [FreeWheel Ad Plugin](#) on page 143 (`freewheel.min.js`).

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

ENABLING THE FREEWHEEL AD SOURCE FOR YOUR ACCOUNT

Before you can use Freewheel ad integration:

1. Log into the Ooyala [Customer Portal](#).
2. Submit a ticket requesting that the Freewheel ad source be added to your Ooyala account.



OPTIONS FOR ASSOCIATING AD TAGS WITH YOUR VIDEO ASSETS

Once the FreeWheel ad source is enabled for your account, you can associate FreeWheel ad tags with your videos using:

- Player Embedded Parameters. Specify embedded parameters at the page level of the player.
- Backlot ad sets. Create ad sets with the Backlot UI or API.

CUSTOMIZING THE AD CONTROL BAR AND MARQUEE FUNCTIONALITY

You can modify [skin.json](#) to customize ad control bar and ad marquee functionality for the FreeWheel ad plugin.

LOADING THE FREEWHEEL AD PLUGIN ON A PAGE

Note:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the Freewheel plugin is automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).
- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add the Freewheel plugin to your web page so that the Freewheel plugin loads before the player is created.

For every page on which you want to use the Freewheel plugin with a player (regardless of how you associated Freewheel ads with your videos):

1. Add the `freewheel.min.js` script to the page where you are loading the player. You must load this plugin *after* you load `core.min.js`.

```
<script language="javascript" src="url_where_hosted/freewheel.min.js"></script>
```

2. Associate the player with the Ooyala Freewheel ad plugin by passing in `freewheel-ads-manager` as one of the player parameters during the player creation (`OO.Player.create`).
3. Pass any global parameters. See [Configuring Ad Parameters](#) on page 180.

OPTION: INTEGRATE WITH FREEWHEEL VIA PLAYER EMBEDDED PARAMETERS

Pass FreeWheel ad tags to the Ooyala player using the `freewheel-ads-manager` parameter and its child parameters. See [FreeWheel Ad Parameters](#) on page 146 for details.

Note: When you use the FreeWheel ad manager, you **must** provide an AdServer URL for HTML5, either via a page-level override "`html5_ad_server`" or via `third_party_module_parameters` with a name "`html5_ad_server`".

The following example shows how to use FreeWheel page-level parameters with Player V4.

Note: The values in the following example are only used to illustrate how to use the `freewheel-ads-manager` parameter. You need to replace them with your own profiles, ids, and URLs.

```
<!DOCTYPE html>
<html>
    <head>
        <title>FreeWheel Example</title>
        <!-- V4 JS core is required. Plugins such as skin, discovery and
        Advertising need to be loaded separately -->
        <script src="url_where_hosted/core.min.js"></script>
```



```

        <script src="url_where_hosted/other-plugin/discovery_api.min.js"></script>
        <!-- Change these html5-skin.min.css and html5-skin.js to your local build if necessary -->
        <script src="url_where_hosted/html5-skin/build/html5-skin.min.js"></script>
        <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
        <!-- A Video Plugin is required. This example shows the Main Video Plugin -->
        <script src="url_where_hosted/video-plugin/main_html5.min.js"></script>
        <!-- Ad module -->
        <script src="url_where_hosted/freewheel.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            "freewheel-ads-manager": {
                // HTML5-specific
                "fw_ad_module_js": "//adm.fwmrm.net/p/exampleprovider_live/AdManager.js",
                "html5_ad_server": "//lb656.v.fwmrm.net/ad/g/1",
                "html5_ssl_ad_server": "//lb656.v.fwmrm.net/ad/g/1",
                "html5_player_profile": "123400:exampleprovider_live_html5",
                // Parameters that should be set on each page on the actual site
                "fw_site_section_id": "EXAMPLEVIDEO_EXAMPLEVIDEO_SHOWS",
                "fw_video_asset_id": 1234567,
                "fw_player_profile": "123400:exampleprovider_live",
                "fw_mrm_network_id": 123400
            }
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID", playerParam);
        });
    </script>
</body>
</html>

```

OPTION: INTEGRATE WITH FREEWHEEL VIA BACKLOT AD SETS

1. Create a FreeWheel ad set using:

- **Backlot UI:** To create an ad set using the Backlot UI, see [Creating Ad Sets for Integrating with Ad Sources](#).
- **Backlot API:** To create an ad set programmatically, see [Ad Sets](#).

Note: For the Ad Source, select Freewheel. Disregard any other ad source with "Freewheel" in the name.

The following table describes the **required ad set fields** for creating a FreeWheel ad set:



Field	Description
Ad Set name	You will enter a name for the ad set.
MRM network ID	Provided by your FreeWheel account.
Player Profile	Provided by your FreeWheel account.
Video Asset network ID	Provided by your FreeWheel account.
Site Section ID	Provided by your FreeWheel account.

Note: The Tracking Pixel URL field is not relevant to the FreeWheel ad manager and should be left blank.

2. Assign an ad set to an asset or multiple assets using:

- **Backlot UI:** To assign your FreeWheel ad set to a single asset, see [Managing Monetization](#). For instructions on how to assign your FreeWheel ad set to multiple assets, see [Bulk Applying Settings](#).
- **Player API:** With the Player API, you can associate an ad set only with an asset on your web page. To associate an ad set with an asset on multiple players you must replicate the code for each player. To associate an ad set with an asset using the Player API, see [Assigning Ad Sets Dynamically](#).
- **Backlot API:** With the Backlot API, you can associate an ad set with an asset more concretely. When you associate an asset with an ad set using the Backlot API, the asset and the ad set will be paired together on any player and page on which you play the asset. To associate an asset with an ad set using the Backlot API, see [Associate Ad Set with Asset](#).

FreeWheel Ad Parameters

Use the following parameters to configure FreeWheel ads in Player V4 using the [FreeWheel Ad Plugin](#) on page 143. See [Integrating FreeWheel Ads](#) on page 143 for details.

Note: These parameters are specific to FreeWheel ads. You can also set global parameters for FreeWheel ads. See [Configuring Ad Parameters](#) on page 180 for details.

Description	Required?
Parent parameter for FreeWheel ads embedded parameters. ads-manager	Yes
Specifies the initial bitrate (user bandwidth, in kbps) for Freewheel ads. If Freewheel ads are playing in low quality, set this to override the initial bitrate selected by Freewheel for rendering ads. See Freewheel's PARAMETER_DESIRED_BITRATE setting.	Number (in kbps)
<pre>"freewheel-ads-manager": {"bitrateOverride":20000}</pre>	

location: the location from which the player will request ads. The value assigned to this key must be HTTP. This is a **string** HTML5 player-specific override, used when the site is loaded over HTTP.

location_ssl: the location from which the player will request ads. The value assigned to this key must be HTTPS. This is a **string** HTML5 player-specific override, used when the site is loaded over HTTPS.

location_framed: the location from which the player will load the FreeWheel AdManager.js. The value assigned to this key must be HTTP. This is a **string** HTML5 player-specific override, used when the site is loaded over HTTP.



Description	Required?
<u>Unique identifier assigned by FreeWheel that identifies a company. Clients are assigned their own Network ID during integration, and the player passes this value to the AdManager before making an ad request.</u>	string
<u>Set of player configurations that define what ads will be playable given the player environment and framework. For example, profiles ensure that the Ad Server does not try to return Silverlight ads for a Flash player. This is set up in the back end during integration, and the correct profile for the player to set will be provided by FreeWheel.</u>	string
<u>Site section tag is a string that identifies the site section, similar to the custom video ID for a video asset. This is the site section tag as specified by you in the Network Module. While the player does not directly interact with BVI, this is the system by which the Custom Asset ID that it will supply is entered into the FreeWheel system. Consequently, the player will need access to at least some of the same CMS information so it can provide this identifier.</u>	string
<u>Type of Site Section ID</u> <u>Id_type</u>	string
<u>Unique identifier of site section network to segment ad retrieval.</u>	string
<u>URL for tracking pixels</u> to monitor the delivery of your ad assets.	string
<u>Piece of content a user</u> is viewing. The player will need to identify the asset to the AdManager for targeting purposes, using the custom asset ID provided via FreeWheel's Bulk Video Ingestion (BVI). This field can be set on the page-level or by movie metadata (configured via Backlot UI or Backlot APIs). If neither are set, the value for this field will fall back to an external ID or an embed code (whichever is used to embed our player).	string

GOOGLE IMA AD PLUGIN

Use the `google_ima.js` plugin to load Google IMA ads in Player V4. The way the `google_ima.js` plugin works, ads will be loaded dynamically before the ad position. The plugin was developed using the [Ad Manager Framework](#) on page 186.

GOOGLE IMA V3.0

Google IMA v3.0 is an ad server and ad management platform. The Google Interactive Media Ads (IMA) SDK enables publishers to display linear, non-linear, and companion ads in interactive media content such as videos and games. Publishers can use the IMA SDK to request and display ads served by DoubleClick for Publishers (DFP), the Google AdSense network, or any VAST-compliant ad server. Common scenarios for using the IMA SDK include displaying pre-rolls or overlays in video content.

Note: Google IMA's HTML5 SDK is not supported on all browsers. For a list of browsers supported by Google IMA, go to <https://developers.google.com/interactive-media-ads/compatibility#platforms>.

Per Google, the Google IMA SDK is intended for the following publishers:

- **DoubleClick for Publishers (DFP) users:** Publishers can use DFP to traffic, target, and serve their directly-sold ads. The IMA SDK offers advanced video ad serving features.
- **AdSense publishers:** Publishers can also use the Google AdSense Network (AdSense for Video (AFV) and AdSense for Games (AFG)) to monetize their inventory. If you only want to display AFV/AFG ads, DFP is not required. However, AFV/AFG ads still require the IMA SDK.
- **Publishers using VAST ads** from a third-party ad server: Publishers can book their VAST ads in DFP and use the IMA SDK to request and display them. VAST ad responses are parsed prior to ad playback. Refer to the VAST guidelines for details.



- **AdExchange for Video:** Publishers can make use of remnant ad inventory from AdExchange for Video via the Google AdSense Network.

SUPPORTED PLATFORMS

Ooyala supports the Google IMA ads served by DFP or AdSense generated in VAST 2.0 and 3.0 and VPAID 1.0 and 2.0 format on the following platforms for this release of Player V4:

- HTML5
- iOS mobile SDK
- Android mobile SDK

AD RULES

Ad Rules define how ads display with video content. Publishers can create Ad Rules on their DFP Premium account to determine when ads play in a video, for how long, and as a result of which triggers, such as content targeting parameters. They can also target Ad Rules so that they apply to:

- Specific videos, such as all episodes of a particular TV show.
- Video metadata, such as videos in the "sports" genre.
- Users' geography.
- Users' browsers.
- Custom targeting keys and values that you define.

For more information about the supported platforms for Ad Rules, go to https://developers.google.com/interactive-media-ads/video_features_and_sdk_versions#html5_footnote

Note: Google IMA's AdRules feature supports ad-breaks without additional code. Do not use AdRules with other types of ads. We highly discourage you from mixing other adTagUrls with an AdRule adTag as it may lead to unexpected behavior.

For ad rule ads, Google handles all ad management and placement of Google IMA ads. For ad rule ads, the Ooyala plugin only tells Google what point the video is at on the video's timeline. One ad tag is used to represent the group of ad rule ads. By default, Ooyala supports Google IMA Ad Rules on iOS Safari (HTML5) with Player V4. See Google's [Supported video player platforms](#) for more information.

For ads without ad rules, Google plays and renders the ad, but the Ad Manager Controller determines when to play the ad. One ad tag is used for each ad.

SUPPORTED AD POSITIONS

For this release, ads are supported on Player V4 HTML5 players and the mobile SDKs for iOS and Android for the following ad positions:

- Pre-Roll
- Mid-Roll
- Post-Roll
- Podded
- Skippable
- Clickthrough
- Overlay (not supported with Ad Rules ads)
- Companion



ADS AND FIREFOX PRIVATE BROWSING

For consumers using [Private Browsing windows](#) in Firefox, some ads may fail to load because privacy mode blocks certain kinds of tracking URLs.

Integrating Google IMA Ads

You can integrate Google IMA ads with Player V4 using the [Google IMA Ad Plugin](#) on page 147 (`google_ima.min.js`).

ENABLING THE GOOGLE IMA AD SOURCE FOR YOUR ACCOUNT

Before you can use Google IMA ad integration:

1. Log into the Ooyala [Customer Portal](#).
2. Submit a ticket requesting that the Google IMA ad source be added to your Ooyala account.

OPTIONS FOR ASSOCIATING AD TAGS WITH YOUR VIDEO ASSETS

Once the Google IMA ad source is enabled for your account, you can associate Google IMA ad tags with your videos using:

- Backlot Ad Sets. Create ad sets with the Backlot UI or API.
- Player Embedded Parameters. Specify embedded parameters at the page level of the player.

NOTES ABOUT THE GOOGLE IMA PLUGIN

- Ads that use Backlot settings will be loaded dynamically before the ad position (as determined in Backlot). The ad will now be requested and loaded at the specified ad position (via Backlot settings or Ad Rules). The video will be paused at the pre-determined ad position and an ad request will be made. The loading icon will be displayed until a successful ad response is returned, or the video playback will resume if an ad error is detected. These behavior changes were made because publishers can be very specific about when ad requests are made.
- For the Google IMA ad plugin, the ad marquee and ad control bar configurations in `skin.json` are ignored (and are forcibly set to off) to avoid blocking ad interactivity.

LOADING THE GOOGLE IMA AD PLUGIN ON A PAGE

Note:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the Google IMA plugin is automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).
- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add the Google IMA plugin to your web page so that the Google IMA plugin loads before the player is created.

For every page on which you want to use the Google IMA plugin with a player (regardless of how you associated Google IMA ads with your videos):

1. Add the `google_ima.min.js` script to the page where you are loading the player. You must load this plugin *after* you load `core.min.js`.

```
<script language="javascript" src="url_where_hosted/google_ima.min.js"></script>
```



- Associate the player with the Ooyala Google IMA ad plugin by passing in `google-ima-ads-manager` as one of the player parameters during the player creation (`OO.Player.create`).

OPTION: INTEGRATE WITH GOOGLE IMA VIA BACKLOT AD SETS

- Create a Google IMA ad set using the Backlot UI or Backlot API:

- Backlot UI:** To create an ad set using the Backlot UI, see [Creating Ad Sets for Integrating with Ad Sources](#). The HTML5 player will respect the following Ad Set settings from Backlot:
 - Ad Tag URL
 - Tracking Pixel URL
 - Ad positions
 - Overlay ads

Ad rules and page-level settings take precedence over Backlot settings.

Note:

- Choose the **Google IMA V3** option for the ad source when creating this ad set. Disregard any other options.
- If the Ad Tag is an Ad Rule, then you must select **Ad Rules** in the Ad Position drop-down.

The **required fields** for creating a Google IMA ad set are:

Field	Description
Ad Set name	Enter a name for the ad set.
Ad Tag	Enter your Google IMA tag here. This is available from your DFP, Adxchange, or Adsense account. See below for an example.

Example Ad Tag:

```
http://123fakeads.g.doubleclick.net/gampad/ads?sz=400x300&iu=%2F6062%2Fiab_vast_samples&ciu_szs=300x250%2C728x90&impl=s&gdfp_req=1&env=vp&output=3Dlinear
```

- Backlot API:** To create an ad set programmatically with the Backlot API, see [Ad Sets](#)
- Assign an ad set to an asset or multiple assets using:
 - Backlot UI:** To assign your Google IMA ad set to a single asset, see the Support Center topic [Managing Monetization](#). To assign your Google IMA ad set to multiple assets, see the Support Center topic [Bulk Applying Settings](#).
 - Player API:** With the Player API, you can associate an ad set only with an asset on your web page. To associate an ad set with an asset on multiple players, you must replicate the code for each player. To associate an ad set with an asset using the Player API, see [Assigning Ad Sets Dynamically](#) on page 182.
 - Backlot API:** With the Backlot API, you can associate an ad set with an asset more concretely. When you associate an asset with an ad set using the Backlot API, the asset and the ad set will be paired together on any player and page on which you play the asset. To associate an asset with an ad set using the Backlot API, see [Associate Ad Set with Asset](#).

OPTION: INTEGRATE WITH GOOGLE IMA VIA PLAYER EMBEDDED PARAMETERS

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a



separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

1. Add the `google_ima.min.js` script to the page where you are loading the player. You must load this plugin *after* you load `core.min.js`.

```
<script language="javascript" src="url_where_hosted/google_ima.min.js"></script>
```

2. Use the Player V4 `OO.Player.create` function to create the player.
3. Pass any global parameters. See [Configuring Ad Parameters](#) on page 180.
4. Pass Google IMA ad tags to the Ooyala player using the `google-ima-ads-manager` parameter and its child parameters. See [Google IMA Ad Parameters](#) on page 157.

Note: Ad sets that are loaded at the page level will override the ad set associated with the asset in Backlot.

Here is an example of how to override an ad set in the Google IMA ad manager on your web page. If you want to replace the Ad Set attached to your video (from Backlot), or if it does not have an IMA Ad Set already associated with it, you will replace "`yourAdTagUrl`" with the actual Google IMA ad tag containing the response. Otherwise, you may leave it out.

```
<script>
  var playerParam = {
    "PCODE": "YOUR_PCODE",
    "playerBrandingId": "YOUR_PLAYER_ID",
    "skin": {
      // Config contains the configuration setting for player skin. Change
      to your local config when necessary.
      "config": "url_where_hosted/skin.json"
    },
    "google-ima-ads-manager": {
      "all_ads": [
        {
          "position_type": "r",
          "tag_url": "http://pubads.g.doubleclick.net/1234567/ads?
sz=640x480&iu=/12345/
pb_preroll_ad&ciu_szs&impl=s&cmsid=123&vid=1234567abcdefg&gdfp_req=1&env=vp&output=xml_v
          }
        ]
      }
    };
  OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
    playerParam);
  });
<script>
```

Here is a more complete example of Google IMA integration that will work for HTML5. The player branding_id of a player and the embed code of an asset can be found in the Embed tab on the MANAGE page of Backlot.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Google IMA Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
    Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/other-plugin/discovery_api.min.js"></
    script>
```



```

<!-- Change these html5-skin.min.css and html5-skin.js to your local
build if necessary -->
<script src="url_where_hosted/html5-skin.min.js"></script>
<link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
<!-- A Video Plugin is required. This example shows the Main Video
Plugin -->
<script src="url_where_hosted/main_html5.min.js"></script>
<!-- Ad module -->
<script src="url_where_hosted/google_ima.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player
                skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            "google-ima-ads-manager": {
                "all_ads": [
                    {
                        "position_type": "r",
                        "tag_url": "//pubads.g.doubleclick.net/1234567/ads?
sz=640x480&iu=/12345/
pb_preroll_ad&ciu_szs&impl=s&cmsid=123&vid=1234567abcdefg&gdfp_req=1&env=vp&output=xml_v
                    }
                ]
            }
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
        });
    </script>
</body>
</html>

```

SPECIAL CASE: GOOGLE DFP PREMIUM

This section applies only if you are using DFP.

Before trafficking ads, **if you are using DFP Premium** you must first map your video content and all related custom metadata to Google's platform. To learn how to do this, see the Support Center topic [Monetizing your Ooyala Content with DFP](#).

Custom Metadata

If your video content has been successfully ingested into DFP, all custom metadata key/value pairs should be visible in DFP's Content tab. These values may be used to target ads against particular types of



content. The following screenshot, for example, is from an individual video asset from Ooyala's test DFP

Targetable metadata [?](#)

keys	values
channel	channel_1
	<input type="text" value="Add a new value"/> <input type="button" value="Add"/>
category	comedy
	<input type="text" value="Add a new value"/> <input type="button" value="Add"/>
cmssource	ad ops account 75708
	<input type="text" value="Add a new value"/> <input type="button" value="Add"/>
sponsor	ford
	<input type="text" value="Add a new value"/> <input type="button" value="Add"/>
<input type="text" value="Add a new content key"/> <input type="button" value="Add"/>	

Save **Archive** [Cancel](#)

account.

Mapping Custom Metadata to DFP Keys

You have the option of creating custom targeting keys on DFP Premium. These keys are then mapped to the key/values ingested from Backlot. To create these keys, in DFP, go to **Inventory > Custom targeting > + New key**, as shown below.

Custom targeting [?](#)

<input type="checkbox"/>	Targeting key ↑	Display name ?	Type	Values
<input type="checkbox"/>	category	Category	Free-form	comedy, drama, family, horror
<input type="checkbox"/>	channel	Channel	Free-form	channel_1, channel_2, channel_3, channel_4
<input type="checkbox"/>	cmssource		Predefined	ad ops account 75708, dfp - ooyala demo, ooyala test feed, se account
<input type="checkbox"/>	sponsor	Sponsorship	Free-form	coca_cola, ford, kellogg, microsoft, united_airlines

Go to page: Show rows: 1 - 4 of 4

For more information, go to the [DFP help article](#) on custom targeting.

Ad Rules

Ad rules can define when ads are inserted, how long they should run for, what format of ads are run, and what to use as the ad source.

Publishers can set up two types of ad rules in DFP in addition to Default Ad Rules that are already available in DFP:

- Standard Ad Rules, which apply to a single stream of content.
- Session Ad Rules, which applies to a visitor's entire visit to customer's pages. They can be applied across multiple content streams and across multiple sites. Session ad rules can only be applied to pre-roll ads.

To enable ad rules for an ad:

1. Specify your ad rules in DFP. For information on implementing DFP ad rules, go to https://support.google.com/dfp_premium/answer/2553686?hl=en.
2. To enable your DFP ad rules to correctly render for your Google IMA V3 ad with the Backlot UI, click **MONETIZE > Ad Sets**, and set the ad position to "ad rules" for the desired ad.

OR



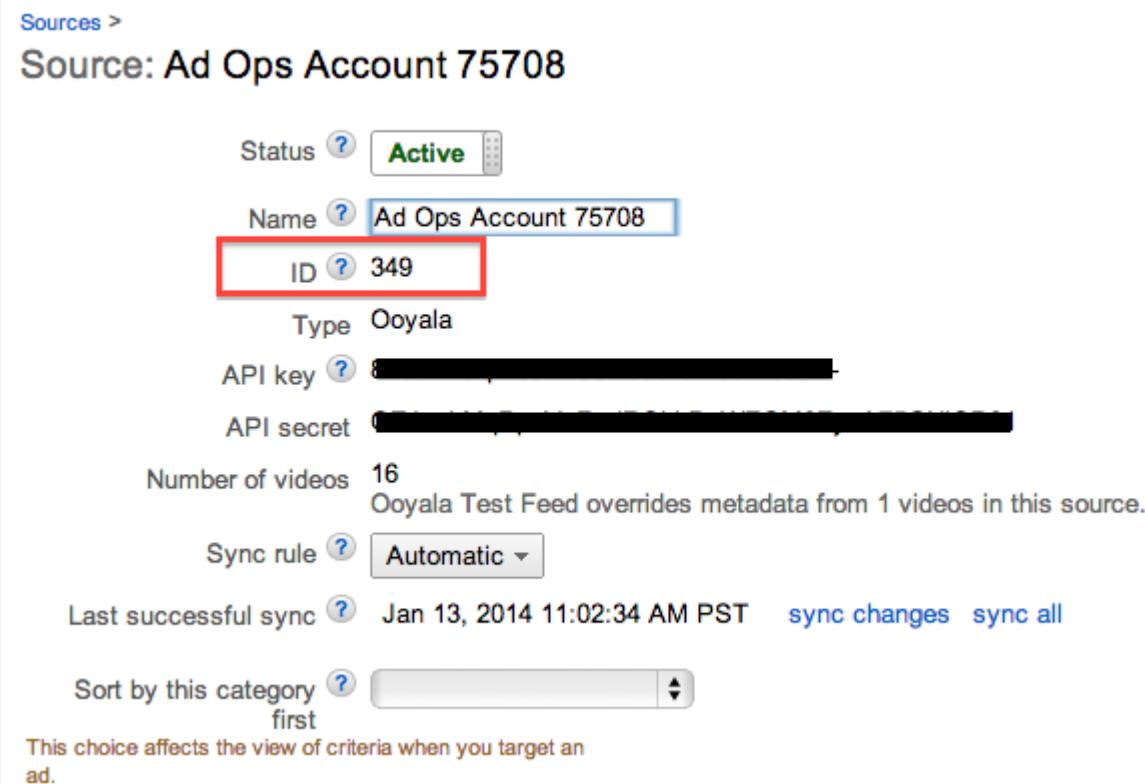
To enable your DFP ad rules to correctly render for your Google IMA V3 ad with the Backlot API, set "ad_type": "rules" for the desired ad. For more information see [Ad Sets](#).

Note: Settings applied at the page level with the ad tag url will override Backlot settings. However, the position type (ad rule or non ad rule) must match on the page level and in Backlot for ads and ad rules to properly render.

Ad Targeting

To enable targeting against content metadata values, Google DFP requires two values to be included on your IMA tag: **cmsid** and **vid**. Once these values are included when making ad requests, the IMA ad manager "knows" which video asset is making the request. As a result, it returns whatever ad response has been defined by the publisher's ad operations team.

cmsid: A unique value assigned automatically by Google to each content source. To locate it within the DFP Premium platform, click on the Video tab (on the upper right), navigate to Sources and click on the source in question. The value is "ID," as shown below:



Sources >
Source: Ad Ops Account 75708

Status ? **Active**

Name ? Ad Ops Account 75708

ID ? **349**

Type Ooyala

API key ? [REDACTED]

API secret [REDACTED]

Number of videos 16
Ooyala Test Feed overrides metadata from 1 videos in this source.

Sync rule ? Automatic ▾

Last successful sync ? Jan 13, 2014 11:02:34 AM PST [sync changes](#) [sync all](#)

Sort by this category ? first

This choice affects the view of criteria when you target an ad.

vid: A unique value for each video asset. The Ooyala-IMA integration uses Ooyala's Content ID.

When creating the IMA Ad Manager adset, the publisher will need to append the IMA ad tag with the macro [oo_embedcode]. Here's an example IMA Ad Manager tag with both cmsid and vid:

[http://googleads.g.doubleclick.net/pagead/ads?client=ca-video-pub-9498212586027311&slotname=8286566767&cmsid=176&vid=\[oo_embedcode\]](http://googleads.g.doubleclick.net/pagead/ads?client=ca-video-pub-9498212586027311&slotname=8286566767&cmsid=176&vid=[oo_embedcode])

You can also pass vid and cmsid programmatically on your web page, as shown in the following example.

```
"google-ima-ads-manager": {  
  "all_ads": [  
    {  
      "position_type": "r",  
      "tag_url": "//pubads.g.doubleclick.net/gampad/ads?  
sz=640x480&iu=/12345/
```



```

pb_preroll_ad&amp;ciu_szs&amp;impl=s&amp;cmsid=123&amp;vid=1234567abcdefg&amp;gdfp_req=1
    }
],
"additionalAdTagParameters": {
    "vid": "embed code",
    "cmsid": "349"
}
}

```

IMA AD RULES

Ad rules define how ads display with video content. You can create ad rules on their DFP Premium account to determine when ads play in a video, for how long, and as a result of which triggers, such as content targeting parameters. You can also target ad rules so that they apply to:

- Specific videos, such as all episodes of a particular TV show.
- Video metadata, such as videos in the "sports" genre.
- Users' geography.
- Users' browsers.
- Custom targeting keys and values that you define.

Example 1: You want to show two 60-second ad breaks during an episode of a popular TV show. The TV show has specific cue points that DFP ingested from your content management system (CMS). You can set up ad rules to specify when the ads should appear (either at the pre-defined cue points or after a certain number of minutes), what types of ads appear, and how many ads should appear during each ad break.

Example 2: For all sports videos on your website, you want to show pre-roll ads followed by a house ad (also called a bumper). You can set up an ad rule to specify what types of ads can show and how long they should run, then target the ad rule to videos with the correct metadata.

SPECIAL CASE: GOOGLE IMA V3 COMPANION ADS

Google IMA V3 companion ads don't use the standard Player WILL_SHOW_COMPANION_ADS event. This is because Google IMA itself is responsible for parsing and generating the companion ad. Google documents how to use the Google Publisher Tag (GPT) at <https://developers.google.com/interactive-media-ads/docs/sdk/html5/companions-gpt>. To use companion ads:

1. Include the following code in your webpage:

```

<script src="//www.googletagservices.com/tag/js/gpt.js">
</script>

```

2. Create the companion tag.
3. Call the right Google API to place the companion ad. For example:

```

<div id="cad" style="width:300px;height:250px;background-color:red">
</div>
<br>
<script type="text/javascript">
    // Add a command to the command queue
    googletag.cmd.push(function() {           // Define the unit

        var adSlot1 = googletag.defineSlot(          "/5129/News/QLD", // Ad
        Unit Name, obtained by the provider
            [300, 250],
        "cad");
        adSlot1.addService(googletag.companionAds());
    });
</script>

```



```

        googletag.enableServices();
        // Immediately signal to show it.

        googletag.display("cad");
    });
</script>

```

Google documents using Companion Ads with the Ad API at <https://developers.google.com/interactive-media-ads/docs/sdk/html5/companions-api>. To display companion ads through the WILL_SHOW_COMPANION_ADS event, slot sizes need to be defined as a page level setting while creating the player as follows:

```

<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "onCreate": function(player) {
            player.mb.subscribe("*", "test", function(event, params) {
                if (event.match(/willShowCompanionAds/)) {
                    console.log(params); // We get the companion ads from params
                }
            });
        },
        "skin": {
            // Config contains the configuration setting for player skin. Change
            to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        },
        companionAd: {
            slots: [{width: 300, height: 250}, {width: 300, height: 60}]
        }
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
    });
</script>

```

If there are any companion ads in the XML with the defined slot sizes, you can retrieve them by listening to the WILL_SHOW_COMPANION_ADS event as demonstrated on the onCreate function code above.

This will get the ads as follows, which is the already formatted companion ad as HTML and the corresponding size, so you can identify the ads and place them anywhere on the page.

```
{
  "ads": [
    {
      "size": "300x250",
      "ad": "<Companion Ad as HTML>"
    },
    {
      "size": "300x60",
      "ad": "<Companion Ad as HTML>"
    }
  ]
}
```

SPECIAL CASE: GOOGLE IMA AND MOBILE

Ooyala supports Google IMA for both desktop and mobile. For information about working with Google IMA for mobile devices, see the following mobile SDK for iOS topics.

- Android - [Integration with Google IMA on Android](#) and [IMA Sample App in Action on Android](#).



- iOS - [Integration with Google IMA on iOS](#) and [IMA Sample App in Action on iOS](#).
- Supported video player platforms: [Google IMA SDK for HTML5](#)

Google IMA Ad Parameters

Use the following parameters to configure Google IMA ads in Player V4 using the [Google IMA Ad Plugin](#) on page 147. See [Integrating Google IMA Ads](#) on page 149 for details.

Note: These parameters are specific to Google IMA ads. You can also set global parameters for Google IMA ads. See [Configuring Ad Parameters](#) on page 180 for details.

Scroll to the right to see whether a parameter is required or optional.

Description	Required?
ads-object The object key used to pass Google IMA ad server tags or network tags to the Ooyala player. This key has effect only with the values described below. ads- manager	Yes
additionalAdTagParameters Additional ad tag parameters get added to the ad tag URL and are automatically populated (if they are macros) before making the ad request. The additional ad tag url parameters need to be in the form of key-value pairs. When you use these parameters, we append them to the end of the ad tag. Appended parameters do not override anything in the URL from Backlot.	No

Valid Values:

If you want the Ooyala code to fill in the ad tag url data using a macro, the following macros are supported:

- [random] or [cache_buster]: These insert a random number
- [timestamp]: This inserts the time at which the request is run
- [referrer_url]: This inserts the URL of the page
- [oo_embedcode]: This inserts the Ooyala embed code for the current video

If you do not use additionalAdTagParameters, you can manually add macros to your ad tag. This circumvents Ooyala's code and is processed directly by the Google iMA SDK. According to Google, the following macros are supported (as of 02/04/2016):

- url=[referrer_url]
- correlator=[timestamp]

Google IMA documentation lists macros at : https://support.google.com/dfp_premium/answer/1181016?hl=en&ref_topic=2480647. You can read Google IMA's documentation on how to add custom parameters to ad tags at https://support.google.com/dfp_premium/answer/1080597?hl=en&ref_topic=2480647

Example:

Here is an example of using Google IMA's macros:

```
http://myAdTagUrl.com?url=[referrer_url]&correlator=[timestamp]
```

Here is an example of how to add parameters through the Ooyala Google IMA plugin:

```
"google-ima-ads-manager": {
    "additionalAdTagParameters": {
        "timestamp": "[timestamp]",
        "video": "[oo_embedcode]",
        "myVar": "value"
    }
}
```



Description	Required?
-------------	-----------

This would make your ad tag change from

```
http://myadtagurl.com
```

to

```
http://myadtagurl.com?
timestamp=[timestamp]&video=[oo_embedcode]&myVar=value
```

Note that [timestamp] and [oo_embedcode] would be filled in with the values specified above.

Array of all the ads to be played during the video. Example:

```
"google-ima-ads-manager": {
  "all_ads": [
    {
      "position": "10000",
      "position_type": "t",
      "tag_url": "http://myadtagurl"
    },
    {
      "position": "5000",
      "position_type": "t",
      "tag_url": "http://myadtagurl"
    }
  ]
}
```

Object
if you do not specify this setting in Backlot; must be specified in either page-level parameters or Backlot.

String Google IMA ad tag. If you want to override the tag attached to the video through Backlot, supply ~~any~~ string value here.

```
ads-
manager.all_ads.tag_url
```

String
if you are specifying the parameter all_ads

Meaning depends on the value of all_ads.position_type:

- **ima-ads-** If position_type is set to "t", then position specifies the time at which the ad should play, in milliseconds:
manager.all_ads.position

- **pre-roll ad:** a value of 0.
- **mid-roll ad:** any value greater than 0 and less than the total time of the video.
- **post-roll ad:** any value greater than the total time of the video.
- If position_type is set to "p", then position specifies the percentage of the total time of the video:
 - **pre-roll ad:** a value of 0.
 - **mid-roll ad:** any value greater than 0 and less than 100.
 - **post-roll ad:** a value of 100 or more.
- If position_type is set to "r" (Ad Rules), then position is ignored.

Manager
for ad overrides without ad rules



Description	Required?
Specifies how to interpret the position value. Valid values: <ul style="list-style-type: none"> • "t" - indicates that position is the time at which the ad should play, in milliseconds. • "p" - indicates that position is the percentage of the total time of the video. • "r" - indicates that Ad Rules determine the position. The all_ads.position setting is ignored. 	String
If set to true, skippable ads can be served to Safari Mobile on iOS 10 (or later) devices if the player is playing inline (not full screen) on the page. This parameter requires that the <code>adsPlayMode</code> : "inline" page-level parameter is specified on the web page. Default is false. Example: <code>enableIosSkippableAds</code>	Boolean
<pre>"google-ima-ads-manager": { "enableIosSkippableAds":true }</pre>	
Note: If playback is in full screen mode when a skippable ad starts playing, the browser window will change to inline playback mode. The user will need to manually return to full screen mode after the ad finishes playing.	
Overrides the zindex of the Google IMA iframe. Set this parameter to 0 if you are using clickable VPAID ads with VPAID INSECURE mode on desktop devices to avoid covering up the ad. You can specify an integer greater than or equal to 0. Default is 10004. Warning: Using this parameter may cause undesirable interactions with the player skin and the video. VPAID INSECURE mode allows UI elements to be placed outside of the iframe. Due to this behavior, the UI elements may not interact well with the Ooyala video player. The Player has zindices in the 1000s range. Elements that are below the player cannot be interacted with through the player. One example is that UI elements that control play, pause, and clickthrough of the ad could be placed outside the iframe with a zindex of 10 (which would place the elements below the player). We recommend that you work with your ad creators or ad providers to encourage use of VPAID secure mode for ads played in the Ooyala player to avoid these side effects.	Integer
If set to true, displays Ooyala's ad control bar while an IMA ad is playing. The IMA ad is displayed above our ad control bar. This feature is disabled by default.	Boolean
Note: In order for this setting to work, the <code>showControlBar</code> setting (under <code>adScreen</code> in <code>skin.json</code>) must also be set to true. For example (in <code>skin.json</code>):	
<pre>"adScreen": { "showAdMarquee": true, "showAdCountDown": true, "showControlBar": true ...}</pre>	
If set to true, ads will be displayed with the Google Ad UI, including the ad countdown, learn more button, and more. Default is false.	Boolean
Note: This parameter can be used only for desktop web. This parameter will not work for mobile web. Note: Do not enable both <code>useGoogleAdUI</code> and <code>useGoogleCountdown</code> at the same time. Doing so might result in unexpected behavior.	
If set to true, the Google IMA countdown timer will be displayed (while still enabling clickthroughs to be triggered by clicks on the video). Default is false.	Boolean
Note: This parameter can be used only for desktop web. This parameter will not work for mobile web. Note: Do not enable both <code>useGoogleAdUI</code> and <code>useGoogleCountdown</code> at the same time. Doing so might result in unexpected behavior.	



Description

Required?

Note: The Google IMA documentation states that: "These settings may be ignored for AdSense and ADX ads."

Example:

```
"google-ima-ads-manager":  
  { "useGoogleCountdown":true }
```

If set to "insecure", the Google IMA SDK will be notified to use the INSECURE VPAID mode. The INSECURE VPAID mode allows you to use "insecure" VPAID creatives.

ads-

manager.vpaidMode

Strong

OYALA PULSE AD PLUGIN

Use the Ooyala Pulse ad plugin to load Ooyala Pulse ads in Player V4. Ooyala Pulse is an ad server and ad management platform, where publisher are able to take full control over their ad inventory, use targeting based on audience and video content, and raise effective CPM and sell-out rates with realtime simulations, dynamic ad delivery and alternative ad sources. The Ooyala Ad Manager Controller (part of the [Ad Manager Framework](#) on page 186) handles when ads play and how video ads are rendered for Ooyala Pulse ads.

SUPPORTED PLATFORMS

The Ooyala Pulse ad manager is available for Player V4 on the following platforms:

- HTML5 web
- iOS
- Android

SUPPORTED AD POSITIONS

The following ad positions and other features are supported for ads served from the Ooyala Pulse ad server on V4 HTML5 players and the iOS and Android mobile SDKs:

- Pre-Roll
- Mid-Roll
- Post-Roll
- Overlays (HTML5 only)
- Skippable
- Clickthrough

SUPPORTED FEATURES

In addition to supported ad positions, Ooyala Pulse ad plugin supports the following features:

- VPAID 1.0 and 2.0 (*), with these considerations:
 - VPAID 1.0 is supported only on web browsers that have Flash support.
 - VPAID 2.0 is supported only on web browsers, even in web browsers used on Android and iOS. For iOS, however, only iPad devices support layering, which means that they can display interactive elements part of a VPAID 2.0 ad. iPhone devices do not support layering and do not display the interactive elements, but they will display the other parts of a VPAID 2.0 ad.



- Sponsor ads
- Multiple ads for each ad break (pre-, mid-, post-roll)

Note: This is not the same as podded ads in a VAST ticket, but the concept is similar to being able to serve multiple ads in a break.

- Passbacks
- Session Extension: mainly used to insert extra ad breaks in live streams

USAGE CONSIDERATIONS

- Muted autoplay is not supported currently with the Pulse plugin. Autoplay should not be used in combination with ads served using the Pulse plugin.

ADS AND FIREFOX PRIVATE BROWSING

For consumers using *Private Browsing windows* in Firefox, some ads may fail to load because privacy mode blocks certain kinds of tracking URLs.

Integrating Ooyala Pulse Ads

You can integrate Ooyala Pulse ads with Player V4 using the *Ooyala Pulse Ad Plugin* on page 160 (`pulse.min.js`).

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see *Ooyala-hosted Player V4 Resources* on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see *Hosting Player V4 Resources* on page 81), be sure to check for any path dependencies within the files.

USAGE CONSIDERATIONS

- Muted autoplay is not supported currently with the Pulse plugin. Autoplay should not be used in combination with ads served using the Pulse plugin.

USAGE CONSIDERATIONS

- Muted autoplay is not supported currently with the Pulse plugin. Autoplay should not be used in combination with ads served using the Pulse plugin.

ENABLING THE OOYALA PULSE AD SOURCE FOR YOUR ACCOUNT

Before you can use Ooyala's Pulse ad integration:

1. Log into the Ooyala *Customer Portal*.
2. Submit a ticket requesting that the Ooyala Pulse ad source be added to your Ooyala account.

OPTIONS FOR ASSOCIATING AD TAGS WITH YOUR VIDEO ASSETS

Once the Ooyala Pulse ad source is enabled for your account, you can associate Ooyala Pulse ads with your videos using:

- *Player Embedded Parameters*. Specify embedded parameters at the page level of the player.
- *Custom Metadata in Backlot*. Specify custom metadata on individual videos with the Backlot UI or API.
- *Backlot Ad Sets*. Create ad sets with the Backlot UI or API.



These ways of setting parameters complement each other, which means that it is possible to set the same parameter in different locations. Values for parameters are read from their base location first, if present, and then read from any subsequent location where they can be set. In general, parameters set in the player override those set as custom metadata, which in turn override those set in the ad set. For a list of where each parameter can be set and where the base values are located, see [Ooyala Pulse Ad Parameters](#) on page 164.

LOADING THE OYALA PULSE PLUGIN ON THE PAGE

Note:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the Ooyala Pulse plugin is automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).
- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add the Ooyala Pulse plugin to your web page so that the Ooyala Pulse plugin loads before the player is created.

For every page on which you want to use the Ooyala Pulse plugin with a player (regardless of how you associated Ooyala Pulse ads with your videos):

1. Add the `pulse.min.js` script to the page where you are loading the player. You must load this plugin *after* you load `core.min.js`.

```
<script language="javascript" src="url_where_hosted/pulse.min.js"></script>
```

2. Associate the player with the Ooyala Pulse ad plugin by passing in `videoplaza-ads-manager` as one of the player parameters during the player creation (`OO.Player.create`).
3. Pass any global parameters. See [Configuring Ad Parameters](#) on page 180.

OPTION: INTEGRATE WITH OYALA PULSE USING PLAYER EMBEDDED PARAMETERS

Pass Ooyala Pulse ad tags to the Ooyala player using the `videoplaza-ads-manager` parameter and its child parameters, as described in [Ooyala Pulse Ad Parameters](#) on page 164.

The following example shows how to use Ooyala Pulse page-level parameters with Player V4.

Note: The values in the following example are used only to illustrate how to use the `videoplaza-ads-manager` parameter. You need to replace them with your own profiles, IDs, URLs, and other values.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ooyala Pulse Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
        Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/other-plugin/discovery_api.min.js"></script>
    <!-- Change these html5-skin.min.css and html5-skin.js to your local
        build if necessary -->
    <script src="url_where_hosted/html5-skin/build/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
    <!-- A Video Plugin is required. This example shows the Main Video
        Plugin -->
    <script src="url_where_hosted/video-plugin/main_html5.min.js"></script>
    <!-- Ad module -->
    <script src="url_where_hosted/pulse.min.js"></script>
```



```

</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player skin.
                // Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            "videoplaza-ads-manager": {
                // Pulse settings
                "pulse_host": "//pulse-host.videoplaza.tv",
                "pulse_device_container": "deviceContainer",
                "pulse_persistent_id": "persistenId",

                // Values mapped to the request settings
                "pulse_height": 640,
                "pulse_width": 360,
                "pulse_max_bitrate": 400,
                "pulse_linear_cuepoints": "5,23,34",
                "pulse_non_linear_cuepoints": "12,14,50",
                "pulse_insertion_point_filter": "onBeforeContent,playbackPosition",
                "pulse_referrer_url": "//www.oyala.com",
                "pulse_linear_slot_size": 3,
                "pulse_max_linear_break_duration": 20,

                // Values mapped to the content metadata
                "pulse_category": "myCategory",
                "pulse_content_form": "longForm", //Can be 'longForm' or
                'shortForm'
                "pulse_content_id": "myCustomId",
                "pulse_content_partner": "myContentPartner",
                "pulse_duration": 52,
                // The flags will be merged with the ones coming from the Backlot
                Ad Set, if any.
                // If you want to override them completely, set
                pulse_override_metadata to true.
                "pulse_flags": "flag1,flag2",
                // The tags will be merged with the ones coming from the Backlot
                Ad Set, if any.
                // If you want to override them completely, set
                pulse_override_metadata to true.
                "pulse_tags": "tag1,tag2",
                "pulse_custom_parameters": {
                    "key1": value1,
                    "key2": value2
                },
                // Integration settings
                "pulse_override_metadata": true, // If set, all the Backlot
                metadata (ad set or custom) will be ignored.
                "pulse_show_ad_title": false
            }
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
            playerParam);
        });
    </script>
</body>

```



```
</html>
```

OPTION: INTEGRATE WITH OYALA PULSE THROUGH BACKLOT CUSTOM METADATA

Add Ooyala Pulse parameters as custom metadata to individual video assets using the Backlot UI or Backlot API:

- **Backlot UI:** For instructions on how to set custom metadata on an asset using the Backlot UI, see [Adding Custom Metadata](#).
- **Backlot API:** For instructions on how to set custome metadata on an asset using the Backlot API, see [Custom Metadata](#).

OPTION: INTEGRATE WITH OYALA PULSE THROUGH BACKLOT AD SETS

1. Create an Ooyala Pulse ad set using:

- **Backlot UI:** To create an ad set using the Backlot UI, see [Creating Ad Sets for Integrating with Ad Sources](#).
- **Backlot API:** To create an ad set programmatically, see [Ad Sets](#). For the Ad Source, select the **Videoplaza Module**.

The **required ad set field** for creating an Ooyala Pulse ad set is:

vpDomain: The host for your Ooyala Pulse account, which is derived from the "sub-domain" found in the Pulse UI. Here is how it is formulated:

```
//[sub-domain].videoplaza.tv
```

2. Assign an ad set to an asset or multiple assets using:

- **Backlot UI:** For instructions on how to assign your Ooyala Pulse ad set to
 - a single asset, see [Managing Monetization](#)
 - multiple assets, see [Bulk Applying Settings](#)
- **Player API:** With the Player API, you can only associate an ad set with an asset on your web page. To associate an ad set with an asset on multiple players you must replicate the code for each player. To associate an ad set with an asset using the Player API, see [Assigning Ad Sets Dynamically](#).
- **Backlot API:** With the Backlot API, you can associate an ad set with an asset more concretely. When you associate an asset with an ad set using the Backlot API, the asset and the ad set will be paired together on any player and page on which you play the asset. For details, see [Associate Ad Set with Asset](#).

Oyala Pulse Ad Parameters

Use the following parameters to configure Ooyala Pulse ads in Player V4 using the [Oyala Pulse Ad Plugin](#) on page 160. See [Integrating Oyala Pulse Ads](#) on page 161 for details.

Note: These parameters are specific to Ooyala Pulse ads. You can also set global parameters for Pulse ads. See [Configuring Ad Parameters](#) on page 180 for details.

OOYALA PULSE-SPECIFIC EMBEDDED PARAMETERS

Name	Description	Type	Required?
videoplaza-ads-manager	The parent parameter for Ooyala Pulse ads embedded parameters.		Yes



Name	Description	Type	Required?
	Required: Yes		
pulse_host	<p>host URL: When the SDK is initialised, the host for the Ooyala Pulse account must be provided. This is used to identify which Pulse account to request ads from. The host is derived from the "sub-domain" found in the Pulse UI and is formulated like this:</p> <pre>http://[sub-domain].videoplaza.tv</pre> <p>Note: The Pulse host must be set in the integration with this parameter, or as Custom Metadata on the video asset in Backlot, or as vpDomain in the Backlot Ad Set associated with the video asset.</p>	string	See Note
pulse_callbacks	<p>Allow clients to provide callbacks which can be used to provide internal instances not otherwise available from the plugin.</p> <p>An array of callback functions. The following functions are available:</p> <ul style="list-style-type: none"> • <code>onSessionCreated</code>: argument is the session and the function is called whenever a Pulse session is created. This way you get access to the session object, and can extend the session when needed. See HTML5 Pulse SDK for an example of session extension and further links to the SDK documentation. • <code>onAdPlayerCreated</code>: argument is the ad player and the function is called whenever a Pulse ad player is created. <p>Example:</p> <pre>pulse_callbacks: { onSessionCreated: function(session){ console.log('> Session created:', session); } , onAdPlayerCreated: function(adPlayer){ console.log('> Ad player created:', adPlayer); } }</pre>	array	No
pulse_category	category: The category is a string that is used to associate content with a category that has been defined for the client's account in the Ooyala Pulse UI. The value of the category can be the Pulse native ID of the category found in the Pulse UI or a human readable "alias" that has been associated with the category in the Pulse UI. Categories are used for reporting and targeting purposes in Pulse UI. It is also possible to apply individual ad policies to each category, controlling how many and what type of ads are returned.	string	No
pulse_content_form	content form: A piece of video content can be defined as either long form or short form content. Long form content can be the entire episode of a program, a full-length feature	string	No



Name	Description	Type	Required?
	movie, or just a piece of content that is more than 10 minutes long. It is up to the client to define the threshold between short and long form. In the Pulse UI it is possible to configure individual ad policies for long and short form, adding an additional ad policy dimension to a category. Valid Values: "longForm", "shortForm"		
pulse_content_id	content ID: Content ID is used for forwarding the ID of the main content to third party trackers and to Pulse for reporting. This value can be added to request URLs for third party ad requests or additional tracking URLs through a placeholder macro in the Pulse UI.	string	No
pulse_content_partner	content partner: The content partner is used to add an additional reporting dimension to the ads that are displayed. The content partner value can use the Pulse native ID of the content partner found in the Pulse UI or a human readable "alias" that has been associated with the content partner in the Pulse UI.	string	No
pulse_custom_params	Any custom parameters you need to set in the integration. An array of key value pairs. Example: {key1: value1, key2: value2}	array	No
pulse_device_container	device container: The device container in Ooyala Pulse is used for targeting and reporting purposes. This device container attribute is only used if you want to override the Pulse device detection algorithm on the Pulse ad server. This should only be set if normal device detection does not work and only after consulting Ooyala personnel. An incorrect device container value can result in no ads being served or incorrect ad delivery and reports.	string	No
pulse_duration	duration: This is the total duration of the main content.	integerNo	
pulse_enforce_cachebusting	cachebusting: Enable or disable the addition of the [CACHEBUSTING] macro to tracking URLs in third party VAST 2.0 tickets. The macro adds a randomised number to the tracking URL each time the associated event is triggered to ensure tracking. Without the randomised number, the URL may be cached and not handled properly. By default, cache busting is enabled.	Boolean	No
pulse_flags	flags: The flags are used to apply special rules and conditions to an ad request. For example, a piece of premium content has been sponsored and no pre-rolls should be displayed. Adding the flag "noprerolls" will prevent pre-rolls from being returned but mid-rolls and post-rolls will be available as normal. Supported flags are: <ul style="list-style-type: none">• nocom: No ads are returned at all• noprerolls: No pre-rolls are returned• nomidrolls: No mid-rolls are returned	string	No



Name	Description	Type	Required?
	<ul style="list-style-type: none"> • nopostrolls: No post-rolls are returned • nooverlays: No overlays are returned <p>A string containing a comma-separated list of values.</p>		
pulse_height	height: The height of the video player. This is used to determine the best size of the delivered media files.	integer	No
pulse_insertion_point	insertion point type: This is the point, in relation to the main content, where the ad spot should be inserted. This will determine what type of ads are requested by the SDK. The insertion point types are defined as constants in the SDKs and are as follows: <ul style="list-style-type: none"> • On before content: Linear ads that are played before the main content starts (Pre-rolls). • On content end: Linear ads that are played after the main content has ended (Post-rolls). • On playback position: Linear ads that are to be displayed at a certain point on the main content's timeline (Mid-rolls). This value requires that you also set the playback position (see playback position). • On pause content: Linear ads that are played when the content is paused. • On playback time: This is for non-linear ads that should be displayed after the user has viewed the main content for X seconds (Overlays). 	string	No
	<p>A string containing a comma-separated list of valid values.</p> <p>Valid Values: "onBeforeContent", "playbackPosition", "onContentEnd", "onPause", "playbackTime"</p>		
pulse_linear_cuepoint	playback position: This is the point on the main content timeline, in seconds, that the "on playback position" (Mid-roll) ad slot should be displayed. A warning may be triggered if this value is higher than the duration of the main content.	string	No
	A string containing a comma-separated list of positions, in seconds. Example: "10, 20, 40"		
pulse_linear_slot_size	Number of linear ads per slot. This value overrides the value set in Ooyala Pulse.	integer	No
pulse_max_bitrate	max bandwidth: The maximum bandwidth in Kbps that your device has access to.	integer	No
pulse_max_linear_break	maxbreakduration: Set the maximum duration for a linear ad break in seconds. For more information about this feature and its limitations, refer to Time Based Breaks .	integer	No
pulse_non_linear_cuepoint	onplayback time: This is for non-linear ads that should be displayed after the user has viewed the main content for X seconds (Overlays).	string	No



Name	Description	Type	Required?
	A string containing a comma-separated list of positions, in seconds. Example: "10,20,40"		
pulse_override_metadata	Set this parameter to true if you want the <code>pulse_flags</code> and <code>pulse_tags</code> to completely override any values coming from the Backlot Ad Set and Backlot Custom Data associated with the video asset. If not set, then all values are merged.	Boolean	No
pulse_persistent_id	persistent identifier: The persistent identifier is used to identify the end user and is the basis for frequency capping, uniqueness, DMP targeting information and more. Ooyala will normally generate and attempt to save a persistent ID as a cookie on your device. In environments that do not support third party cookies, such as iOS and Android native applications, this automatic id management does not work and it is then up to the integration to provide an id that is consistent for the individual end user and that persists between sessions. This ID can be any string as long as it is unique to each individual end user, although the GUID or UUID format is recommended. If the application has a login or other user ID that is consistent across devices, this ID can be used to track uniqueness across devices and reuse DMP tracking data across devices and platforms.	string	No
pulse_referrer_url	referrer URL: To override the URL from where the ad requests originate in the HTTP header's referrer property.	string	No
pulse_seekmode	seek mode: Set how the session behaves when a viewer seeks or scrubs past one or more mid-roll ad breaks. The different modes allow you to ignore any scrubbed ad breaks (default), play all scrubbed ad breaks, play only the first scrubbed ad break or play only the last scrubbed ad break. Valid values: <ul style="list-style-type: none">• IGNORE - Do not enforce midroll ad breaks to be played. This is the default.• PLAY_FIRST - If the viewer seeks past one or more ad breaks, the first one is played before content resumes.• PLAY_LAST - If the viewer seeks past one or more ad breaks, the last one is played before content resumes.• PLAY_ALL - If the viewer seeks past one or more ad breaks, all of them are played before content resumes.	string	No
pulse_show_ad_title	Show the ad title in the ad marquee. The default value is 'false'.	Boolean	No
pulse_tags	tags: Tags are freeform keywords that can be used for targeting and reporting purposes. Although the tags keywords are able to handle UTF-8 characters, we recommend refraining from using special characters like quotes (' and ") ampersand (&) comma (,) semi-colon (;).	string	No



Name	Description	Type	Required?
pulse_use_vast_skip_offset	Note: In an integration between Ooyala Player V4 and Ooyala Pulse, no special characters are allowed. A string containing a comma-separated list of values.	Boolean	No
pulse_width	width: This parameter only applies to the HTML5 SDK. Enable or disable using the skip offset defined in third party VAST tickets over using the skip offset defined in the insertion policies in Pulse. By default, the skip offset from Pulse is used.	integer	No
	width: The width of the video player. This is used to determine the best size of the delivered media files.		

WHERE PARAMETERS ARE READ FROM

The following table lists the parameters you can set in the Ooyala Pulse ad plugin for Player V4, and the relationship between the different locations where these parameters are read from, if not set in the integration.

Note: Consider the following rules when reading the table:

- In general:
 - Embedded parameters override custom metadata set on individual assets in the Backlot UI.
 - Custom metadata overrides ad set parameters set in the Backlot UI.
 - Ad set parameters override content metadata from assets in the Backlot UI.
- ⁽¹⁾ indicates the location from which the base value for that particular parameter is read.
- ⁽²⁾ indicates that values set here are merged with the column(s) to right.
- ⁽³⁾ indicates that values set here override, or are merged with, the column(s) to the right, depending on the value for pulse_override_metadata.

Embedded Parameters	Custom Metadata	Ad Set Parameters	Content Metadata
Pulse Settings			
pulse_device_container ⁽¹⁾			
pulse_host	pulse_host	vpDomain	
		vpDomain	
		(1)	
pulse_persistent_id ⁽¹⁾			
Pulse Request Settings			
pulse_enforce_cachebusting ⁽¹⁾			
pulse_height ⁽¹⁾			
pulse_insertion_point	pulse_insertion_point	Ad Position	
		(1)	
pulse_linear_cuepoint	pulse_linear_cuepoints	Player level midroll ad breaks	
		(1)	



Embedded Parameters	Custom Metadata	Ad Set Parameters	Content Metadata
<code>pulse_linear_slot_size</code> ⁽¹⁾	<code>pulse_linear_slot_size</code> ⁽¹⁾		
<code>pulse_max_bitrate</code> ⁽¹⁾			
<code>pulse_max_linear_break_duration</code> ⁽¹⁾	<code>pulse_max_linear_break_duration</code> ⁽¹⁾		
<code>pulse_non_linear_cuepoints</code> ⁽¹⁾	<code>pulse_non_linear_cuepoints</code> ⁽¹⁾	Non Linear ad breaks	
		(1)	
<code>pulse_referrer_url</code>	<code>pulse_referrer_url</code> ⁽¹⁾		
<code>pulse_seekmode</code> ⁽¹⁾			
<code>pulse_use_vast_skipoffset</code> ⁽¹⁾			
<code>pulse_width</code> ⁽¹⁾			
Pulse Content Metadata			
<code>pulse_category</code>	<code>pulse_category</code>	Player level shares	
		(1)	
		In the ad set, this parameter contains both the category and the content partner separated by a comma.	
<code>pulse_content_form</code>	<code>pulse_content_form</code>	Short form content max. length (sec.)	
		(1)	
		This parameter together with the video asset's duration determine the content form.	
<code>pulse_content_id</code>	<code>pulse_content_id</code>		content ID
			(1)
<code>pulse_content_partner</code>	<code>pulse_content_partner</code>	Player level shares	
		(1)	
		In the ad set, this parameter contains both the category and the content partner separated by a comma.	
<code>pulse_custom_parameters</code> ⁽¹⁾			
<code>pulse_duration</code>	<code>pulse_duration</code>		length ⁽¹⁾
<code>pulse_flags</code> ⁽³⁾	<code>pulse_flags</code> ⁽²⁾	Player level Videoplaza flags	
		(1)	
<code>pulse_tags</code> ⁽³⁾	<code>pulse_tags</code> ⁽²⁾	Player level tags	



Embedded Parameters	Custom Metadata	Ad Set Parameters	Content Metadata
		(1)	

Other

`pulse_override_metadata(1)`
`pulse_show_ad_title(1)`

VAST AND VPAID AD PLUGIN

Use the `ad_manager_vast.js` ad plugin to load VAST and VPAID ads in Player V4. The Ad Manager Controller (developed using the [Ad Manager Framework](#) on page 186) handles ad management and placement of VAST and VPAID ads.

SUPPORTED PLATFORMS

Ooyala Player V4 supports the following platforms for VAST and VPAID:

- HTML5
- iOS mobile SDK
- Android mobile SDK

VAST 2.0 AND 3.0

Video Ad Serving Template (VAST) is an ad format that uses XML to describe linear ads (video ads), non-linear ads (overlay ads) and companion ads.

Player V4 is VAST 2.0 and VAST 3.0 compliant, which means that you can use the ad tags of these standards to run overlays, as well as pre-, mid- and post-rolls, on your video content. You can specify which VAST version to use in your XML. For VAST 3.0, the Ooyala player supports the IAB VMAP standard to let you program your ad breaks on the video player and supports pre-roll, mid-roll, post-roll, podded, clickthrough, companion, and skippable ads.

VPAID 2.0

[IAB](#) defines VPAID (Video Player Ad-Serving Interface Definition) as a standard that "establishes a common interface between video players and ad units, enabling a rich interactive in-stream ad experience." Go to <http://www.iab.com/guidelines/digital-video-player-ad-interface-definition-vpaid-2-0/> for details on VPAID.

Ooyala's V4 player is VPAID 2.0 compliant, supporting pre-roll, mid-roll, post-roll, podded, clickthrough, skippable, and interactive ads.

LIVERAIL VAST ADS

The VAST and VPAID plugin supports LiveRail ad tag URLs that are VAST XML. This correlates to the LiveRail (VAST) Overlay and LiveRail (VAST) In-Stream ad set types in Backlot.

Note: The VAST and VPAID plugin does not support LiveRail ads that do not use the VAST XML format.



ADS AND FIREFOX PRIVATE BROWSING

For consumers using [Private Browsing windows](#) in Firefox, some ads may fail to load because privacy mode blocks certain kinds of tracking URLs.

Integrating VAST and VPAID Ads

You can integrate VAST and VPAID ads with Player V4 using the [VAST and VPAID Ad Plugin](#) on page 171 (`ad_manager_vast.js`).

ENABLING THE VAST/VPAID AD SOURCE(S) FOR YOUR ACCOUNT

Before you can use VAST and VPAID ad integration:

1. Log into the Ooyala [Customer Portal](#).
2. Submit a ticket requesting that the VAST and VPAID ad source be added to your Ooyala account.

CUSTOMIZING THE AD CONTROL BAR AND MARQUEE FUNCTIONALITY

You can modify [skin.json](#) to customize ad control bar and ad marquee functionality for the VAST and VPAID plugin rendering VAST 2.0 or VAST 3.0 ads.

Note: For the VAST and VPAID plugin rendering VPAID 2.0 ads, the ad control bar and ad marquee configurations in skin.json are ignored (and are forcibly set to off) to avoid blocking ad interactivity.

STEP 1: CREATE AN AD SET USING BACKLOT

After you have VAST and/or VPAID enabled for your account, you can associate VAST and/or VPAID ad tags with your videos. For VAST and VPAID ads, you must create an ad set with Backlot using:

- **Backlot UI:** To create an ad set using the Backlot UI, see [Creating Ad Sets for Integrating with Ad Sources](#).
- **Backlot API:** To create an ad set programmatically, see [Ad Sets](#).

Note:

- For Ad Source, choose **VAST Compliant In-Stream**. Disregard all other options with "VAST" in the name.
- For this version, the steps for configuring VAST and VPAID ad sets are identical.

VAST Compliant In-Stream Ad Set

Field	Description
Ad Set name	You will enter a name for the ad set.
Ad Tag	You will input your VAST 2.0 or VAST 3.0 ad tag here (this is provided by an ad server).
Ad Position	Define where in the video stream the VAST ad should appear: <ul style="list-style-type: none">• Pre-roll• In-stream (mid-roll): Include the time in seconds where the VAST tag will be called.• Post-roll
Tracking Pixel URL	Use this field to track VAST ad impressions using your own or a 3rd-party URL.



LiveRail (VAST) In-Stream Ad Set

Field	Description
Ad Set name	You will enter a name for the ad set.
Ad Tag	You will input your VAST 2.0 or VAST 3.0 ad tag here (this is provided by an ad server).
Tracking Pixel URL	Use this field to track VAST ad impressions using your own or a 3rd-party URL.
Ad Position	Define the time (in seconds) where the overlay ad should appear.

STEP 2: ASSIGN THE AD SET TO AN ASSET USING BACKLOT OR THE PLAYER API

Assign an ad set to an asset or multiple assets using:

- **Backlot UI:** To assign your VAST ad set to a single asset, see [Managing Monetization](#). For instructions on how to assign your VAST ad set to multiple assets, see [Bulk Applying Settings](#).
- **Player API:** With the Player API, you can associate an ad set with only an asset on your web page. To associate an ad set with an asset on multiple players, you must replicate the code for each player. See [Assigning Ad Sets Dynamically](#) on page 182 for details.
- **Backlot API:** With the Backlot API, you can associate an ad set with an asset more concretely. When you associate an asset with an ad set using the Backlot API, the asset and the ad set will be paired together on any player and page on which you play the asset. See [Associate Ad Sets with Assets](#) for details.

STEP 3: INTEGRATE THE VAST AND VPAID PLUGIN WITH THE OYALA PLAYER ON THE PAGE

Note:

- If you use Backlot to generate your HTML embed code, and you choose the **V4 HTML5 Standard Player Embed Code (recommended)** option, the VAST and VPAID plugin is automatically included and should not be explicitly added to the web page where you launch the player. For details, see [Configuring Player Embed Settings in Backlot](#).
- If you choose the **V4 HTML5 Player Embed Code (Advanced)** option in Backlot, or if you manually create the HTML embed code yourself, you need to add the VAST and VPAID plugin to your web page so that the VAST and VPAID plugin loads before the player is created.

For every page on which you want to use the VAST and VPAID plugin with a player (regardless of how you associated VAST / VPAID ads with your videos):

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

1. Add the `ad_manager_vast.min.js` script to the page where you are loading the player. You must load this plugin *after* you load `core.min.js`.

```
<script language="javascript" src="url_where_hosted/
ad_manager_vast.min.js"></script>
```

2. Use the Player V4 `OO.Player.create` function to create the player.
3. Pass any global parameters. See [Configuring Ad Parameters](#) on page 180.



- Pass VAST ad tags to the Ooyala player using the `vast` parameter and its child parameters. See [VAST-Specific Embedded Parameters](#) for the list of key/value pairs you can use with the `vast` parameter.

Note: In this version, you can override only the Ad Tag setting with a page-level parameter (`vast.tagUrl`).

The following example shows how to use VAST and VPAID plugin page-level parameters with Player V4.

```
<!DOCTYPE html>
<html>
    <head>
        <title>VAST Example</title>
        <!-- V4 JS core is required. Plugins such as skin, discovery and
        Advertising need to be loaded separately -->
        <script src="url_where_hosted/core.min.js"></script>
        <script src="url_where_hosted/other-plugin/discovery_api.min.js"></
script>
        <!-- Change these html5-skin.min.css and html5-skin.js to your local
        build if necessary -->
        <script src="url_where_hosted/html5-skin.min.js"></script>
        <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
        <!-- A Video Plugin is required. This example shows the Main Video
        Plugin -->
        <script src="url_where_hosted/main_html5.min.js"></script>
        <!-- Ad module -->
        <script language="javascript" src="url_where_hosted/
ad_manager_vast.min.js"></script>
    </head>
    <body>
        <div id="container" style="width:640px; height:360px;"></div>
        <script>
            var playerParam = {
                "PCODE": "YOUR_PCODE",
                "playerBrandingId": "YOUR_PLAYER_ID",
                "skin": {
                    // Config contains the configuration setting for player
                    skin. Change to your local config when necessary.
                    "config": "url_where_hosted/skin.json"
                },
                "vast": {
                    "all_ads": [
                        {
                            "position": 0,
                            "position_type": "t",
                            "tag_url": "myUrl"
                        }
                    ]
                }
            };
            OO.ready(function() {
                window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
            });
        </script>
    </body>
</html>
```



VAST and VPAID Ad Parameters

Use the following parameters to configure VAST, VPAID, and LiveRail VAST ads in Player V4 using the [VAST and VPAID Ad Plugin](#) on page 171. See [Integrating VAST and VPAID Ads](#) on page 172 for details.

Note: These parameters are specific to VAST and VPAID ads. You can also set global parameters for VAST and VPAID ads. See [Configuring Ad Parameters](#) on page 180 for details.

Name	Description	Type Required?
vast	Parent parameter for the VAST module. Example: <pre>"vast": { "all_ads": [{ "position": 0, "position_type": "t", "tag_url": "myUrl" }] }</pre>	Yes
vast.all_ads.position	If position_type is set to 't', position is the time at which the ad should play, in milliseconds. A value of 0 should be set for pre-roll ads. Any value less than the total time of the video is recognized as a mid-roll ad. Any value greater than the total time of the video is recognized as a post-roll ad. <ul style="list-style-type: none">If position_type is set to 'p', the value of position is the percentage of the total time of the video.	integer Overrides for ad rules
vast.all_ads.position_type	Specifies how to interpret the position value. Valid Values: <ul style="list-style-type: none">t - indicates that position is the time at which the ad should play, in milliseconds.b - indicates that position is the percentage of the total time of the video.	string Yes
vast.all_ads.tagUrl	VAST ad tag. If you want to override the tag attached to the video through Backlot, supply any string value here.	string Yes
vast.tagUrl (deprecated)	Note: If specified, vast.tagUrl (deprecated) overrides this setting. A key/value pair representing the ad server or network URL. This correlates with the Ad Tag field in the VAST or VPAID ad set. Note: Use vast.all_ads.tag_url instead of this deprecated setting. This deprecated setting overrides vast.all_ads.tag_url but does not allow you to control ad placement.	string Yes if you are specifying the parameter all_ads
vast.vpaidTimeout	VPAID timeout setting. Used to configure how long to wait for a VPAID ad to play through. Allows you to exit the ad and resume the original content if ad playback is delayed past the timeout for any reason. Includes the following parameters: <ul style="list-style-type: none">iframe - time (in seconds) to wait for the VPAID iFrame to be loaded (default is 5 seconds)	string es



Name	Description	Type Required?
	<ul style="list-style-type: none"> • loaded - time (in seconds) to wait for notification that the ad has been loaded (default is 5 seconds) • started - time (in seconds) to wait for notification that the ad has started (default is 5 seconds) • stopped - time (in seconds) to wait for notification that the ad has stopped after receiving an ad end event(default is 5 seconds) <p>Upon timeout, the player stops playing the ad and returns to playing the original content.</p> <p>Example (page-level override to six seconds):</p> <pre>"vast": {"vpaidTimeout" : {"iframe":6, "started":6, "loaded":6, "stopped":6} }</pre>	

INTEGRATING ADS

To deploy ads for Player V4, you must implement at least one of the following ad plugins:

- [FreeWheel Ad Plugin](#) on page 143
- [Google IMA Ad Plugin](#) on page 147
- [Ooyala Pulse Ad Plugin](#) on page 160
- [VAST and VPAID Ad Plugin](#) on page 171

BEFORE YOU BEGIN

- To help decide which plugin(s) to use, see [Plugin Support for IAB VAST and VPAID Formats](#).
- For each plugin, you need to add the ad source to your account. Log into the [Customer Portal](#) and submit a ticket requesting that the ad source(s) (Ooyala Pulse, FreeWheel, Google IMA, LiveRail, VAST, or VPAID) be added to your Ooyala account.

Note: If you are using the Freewheel, VAST, or Google IMA ad plugins, you can load multiple ad plugins on a page ([Integrating Multiple Ad Managers](#) on page 177). Otherwise, you can load only one ad plugin per player (at a time).

INTEGRATION INSTRUCTIONS

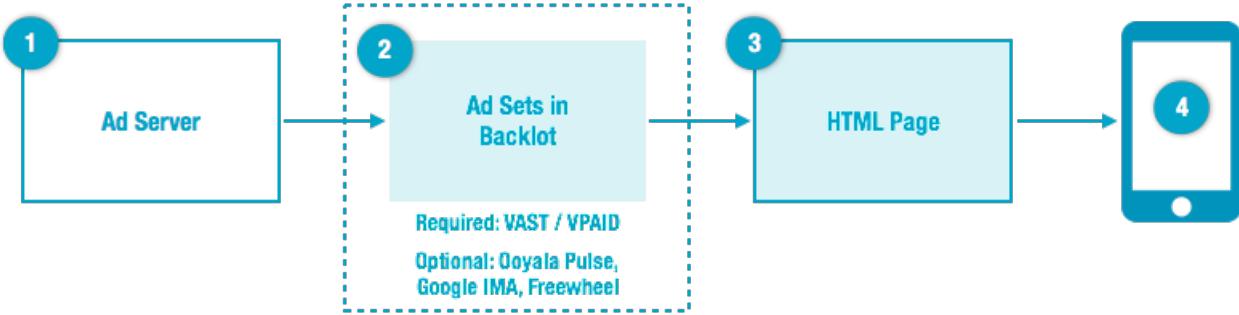
Refer to the following topics:

- [Integrating Ooyala Pulse Ads](#) on page 161
- [Integrating FreeWheel Ads](#) on page 143
- [Integrating VAST and VPAID Ads](#) on page 172
- [Integrating Google IMA Ads](#) on page 149

Ads Integration Workflow

The following figure shows the high-level workflow of integrating ads with Player V4.





STEP 1: CREATE ADS

Work with your ad server/ad manager to create ads. See [Ad Types](#) on page 187 for definitions of commonly used ad types.

STEP 2: CREATE AD SETS AND LINK TO VIDEO ASSETS IN BACKLOT

Note:

- Required for VAST and VPAID ads.
- Optional for Ooyala Pulse, Google IMA and FreeWheel ads. If you want ads associated with your video, if you do not associate an ad set with your video asset in Backlot, you must associate ads with your players and video asset via [ad manager-specific page level parameters](#).

Create ad sets in Backlot and associate them with your video assets.

1. Create an ad set in Backlot with the [Backlot UI](#) or [Backlot API](#). Ad sets are ad configurations that control the ad experience of an asset during playback.
2. Assign an ad set to an asset or multiple assets with the [Backlot UI](#), [Backlot API](#), or Player API.

Note: With the Player API, you can associate an ad set only with an asset on your web page. To associate an ad set with an asset on multiple playersm you must replicate the code for each player. With the Backlot API, you can associate an ad set with an asset more concretely. That is, when you associate an asset with an ad set using the Backlot API, the asset and the ad set will be paired together on any player and page on which you play the asset.

Note: For VAST and VPAID ads, you must create an ad set in Backlot and associate it with your asset via Backlot.

STEP 3: INTEGRATE ADS ON THE HTML PAGE

Integrate your ad manager with the V4 player at the (HTML5) page-level by loading the relevant ad plugin and, optionally, specifying parameters to configure your ads and associate ads with your players and video assets.

Note: You must load the ad plugin for ads to work with your player.

STEP 4: VIEW AD CONTENT

The video consumer is now able to view the ad content associated with the video asset.

Integrating Multiple Ad Managers

You can load and use multiple ad plugins (only the Freewheel, VAST, and Google IMA ad plugins) on a single page. If all three were loaded on a page, for example, and you had multiple embedded players on



the page, Player A and Player B could both use IMA, Player C could use Freewheel, and Player D could use VAST.

Note: If you want to use the Pulse plug-in, consider using an iFrame instead. If you load the pulse plugin and do not give it ads, you get unexpected behaviors.

See [Ad Manager Framework](#) on page 186 for an overview and [Integrating Ads](#) on page 176 for an introduction.

STEP 1: LOAD THE PLUGINS ON THE PAGE

Consider the following requirements:

- You can load multiple ad plugins on a page, as long as you use only the Freewheel, VAST, and Google IMA ad plugins.
- Plugins should all be the same Player V4 version.
- You can load only one version of the same ad plugin on a page. For example, you cannot load two different versions of the Freewheel plugin on the same page.
- Load only the plugins you intend to use on that page.

To load an ad plugin, add the following script (after you load `core.min.js`) to the page where you are loading the players. For example, to load all three ad plugins:

```
<script language="javascript" src="url_where_hosted/freewheel.min.js"></script>
<script language="javascript" src="url_where_hosted/ad_manager_vast.min.js"></script>
<script language="javascript" src="url_where_hosted/google_ima.min.js"></script>
```

Replace `url_where_hosted` with the URL where you host the resource.

STEP 2: LOAD AND CONFIGURE PLAYERS ON THE PAGE

Consider the following requirements:

- Multiple embedded players on a page can access the same ad plugin.
- However, you can use only one ad plugin per embedded player. For example, Player A and Player B can use only IMA, not IMA and VAST.

After the applicable ad managers are enabled on your page, you can [configure general ad settings](#) with embedded parameters. You can also configure your ad manager-specific settings, as described in:

- [Integrating VAST and VPAID Ads](#) on page 172
- [Integrating FreeWheel Ads](#) on page 143
- [Integrating Google IMA Ads](#) on page 149

EXAMPLE

Replace `url_where_hosted` with the URL where you host the resource.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="url_where_hosted/core.js"></script>
    <script src="url_where_hosted/ad-plugin/freewheel.js"></script>
    <script src="url_where_hosted/ad-plugin/ad_manager_vast.js"></script>
    <script src="url_where_hosted/ad-plugin/google_ima.js"></script>
    <script src="url_where_hosted/video-plugin/bit_wrapper.min.js"></script>
```





```

        );
    </script>
</body>
</html>

```

Configuring Ad Parameters

On any page where you load Player V4 with integrated ads, you can use embedded, page-level ad parameters to configure ad settings.

TYPES OF EMBEDDED PARAMETERS

There are two types of parameters you can use:

- **Global parameters** that you configure *outside* of the ad manager hash. These parameters are described later in this topic.
- **Ad manager-specific parameters** that you configure *within* the ad manager hash. These managers are described in the following topics:
 - [Ooyala Pulse Ad Parameters](#) on page 164
 - [FreeWheel Ad Parameters](#) on page 146
 - [Google IMA Ad Parameters](#) on page 157
 - [VAST and VPAID Ad Parameters](#) on page 175

PRECEDENCE VARIES PER PLUGIN

- For Ooyala Pulse, FreeWheel, and VAST ads, settings configured *inside* the ad-manager hash will override settings configured outside of the ad manager hash.
- For Google IMA, settings configured *outside* the ad-manager hash will override settings configured inside of the ad manager hash.

GLOBAL PARAMETERS

The following global parameters work for *all* ad types supported by the Ooyala V4 player in this release. You can add any of these global parameters to the `playerParam` variable to modify the default ad settings.

Note: The `showAdMarquee` and `showInAdControlBar` parameters are no longer configurable at the page level or ad manager level. You can configure the ad marquee and control bar by modifying `skin.json`.

Description	Required?
<code>allowAdClickThrough</code> : An integer value, in seconds, that is applied in two ways: <ul style="list-style-type: none"> • The Ad Manager Controller uses this parameter to control ad load timeouts (ad request plus ad playback) for all ad plugins. If the ad request was successful but the ad playback is taking too long to start, specifying this setting controls when ad playback times out. Default: 25 seconds • The Google IMA Ad Plugin on page 147 uses this parameter to control the ad request timeout for Google IMA ads. Default: 15 seconds 	Integer
<code>allowAdClickThroughOnVideo</code> : If set to <code>true</code> (the default), when the user clicks or taps on an ad video or the learn more button, this action will pause the ad and open the clickthrough url in a new tab. Note: This is set to <code>false</code> , tapping or clicking on the ad video will only pause (and later resume) the ad, but not open the clickthrough URL, and clicking or tapping on the learn more button still displays a new page.	Boolean



Description	Required?
only for use with the Ooyala VAST and VPAID Ad Plugin.	
A value, in seconds, after which the linear ad skip button will display. If set to 0, the skip button will appear immediately. If the ad is shorter than the time specified, the skip button will never appear. Can be any integer greater than or equal to 0 (default is 5 seconds).	Integer
<ul style="list-style-type: none"> ▶ If set to <code>true</code> (the default), ad playback will pause when a user clicks on it. • If set to <code>false</code>, the ad will continue to play when clicked. 	Boolean
<ul style="list-style-type: none"> ▶ If set to <code>true</code> (the default), ads will play when a video is replayed. • If set to <code>false</code>, ads will not replay when the video is replayed. 	Boolean
<ul style="list-style-type: none"> ▶ If set to <code>true</code>, enables the skip ad button during video ads. • If set to <code>false</code> (the default), disables the skip ad button. 	Boolean
<ul style="list-style-type: none"> ▶ If set to <code>true</code>, the player will display a close button (X) for overlay ads. • If set to <code>false</code> (the default), the player will not display a close button for overlay ads. 	Boolean

SERVER SIDE AD INSERTION PLUGIN PARAMETERS

To modify the ad setting for the Server Side Ad Insertion plugin, use the following parameter:

Description	Required?
<code>implementsCacheBusting</code> cache busting for ad requests when using Server Side Ad Insertion.	Boolean
<ul style="list-style-type: none"> ▶ If set to <code>true</code> (the default), URLs with the '[CACHEBUSTING]' macro will be replaced with a random string. • If set to <code>false</code>, URLs with the '[CACHEBUSTING]' macro will <i>not</i> be replaced with a random string. 	

AD MANAGER EXAMPLE

The following example shows modifying the general settings for the example ad manager. The embed code of an asset can be found in the Embed tab on the MANAGE page of Backlot.

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ads Embedded Parameters Example</title>
```



```

<!-- V4 JS core is required. Plugins such as skin, discovery and
Advertising need to be loaded separately -->
<script src="url_where_hosted/core.min.js"></script>
<!-- Change these html5-skin.min.css and html5-skin.js to your local
build if necessary -->
<script src="url_where_hosted/html5-skin.min.js"></script>
<link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>
<!-- A Video Plugin is required. This example shows the Main Video
Plugin -->
<script src="url_where_hosted/main_html5.min.js"></script>
<!-- Ad module - replace with an actual ad plugin-->
<script src="url_where_hosted/ad_manager_example.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player
                skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            //Apply global ad parameters here
            "example-ads-manager": {
                //This is the ad manager-specific hash where you apply ad
                manager-specific parameters
                "adTagUrl": "//my-ad-tag.com"
            },
            "autoplay": false,
            "preload": false,
            "allowAdClickThroughOnVideo": true,
            "replayAds": true
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
        });
    </script>
</body>
</html>

```

Assigning Ad Sets Dynamically

You can override a Backlot ad set with one that you specify at runtime.

ENABLING DYNAMIC AD SETS

Contact your Ooyala support representative to enable the dynamic ad set code feature for your account.

JAVASCRIPT FUNCTIONS USED TO ASSIGN AD SETS DYNAMICALLY

Use the following Player V4 JavaScript functions to specify an ad set at runtime. Each ad set has a unique ad set code identifier (adSetCode parameter), which you pass to identify the ad set when assigning it dynamically.

Type	Description
OO.Player.setEmbedCode	Assign an ad set to a specific embed code.



Type	Description
OO.Player.create	Assign an ad set to a specific player and video asset.

STEP 1: SET UP THE ADSETCODE IN BACKLOT

First, you need to set up the adSetCode in Backlot.

Note: The ad set code is not available by default for security reasons. Contact your Ooyala support representative to enable this feature.

1. In the Backlot UI, associate an ad set with a video asset.
2. Retrieve the adSetCode. The following figure shows an example of this association, along with the Ad Set Code that you create in the MONETIZE tab.

STEP 2: SPECIFY THE ADSETCODE ON THE WEB PAGE

Associating an Ad Set with a Player and Video Asset

To associate the ad set with a particular player and video asset, use the `OO.Player.create` function to generate a player and add the parameter `adSetCode` with the ad set code value retrieved from step 1.

```
<script>
    var playerParam = {
        "PCODE": "YOUR_PCODE",
        "playerBrandingId": "YOUR_PLAYER_ID",
        "skin": {
            // Config contains the configuration setting for player skin.
            // Change to your local config when necessary.
            "config": "url_where_hosted/skin.json"
        },
        "adSetCode": "yourAdSetCode"
    };
    OO.ready(function() {
        window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
        playerParam);
    });
</script>
```

Associating an Ad Set with an Embed Code

To assign an ad set to a specific embed code, use the `setEmbedCode` function. The `setEmbedCode` function takes an additional `options` parameter, which must be an object and must belong to the same



provider as the embed code. In the `options` hash, use the `adSetCode` parameter with the ad set code value retrieved from step 1, as shown in the following example.

```
<script>
    player.setEmbedCode(embedCode, {
        "adSetCode": "yourAdSetCode"
    });
</script>
```

Ads and Live Streams

You can configure ads to use with live streams.

SUPPORTED AD PLUGINS FOR LIVE ADS INTEGRATION

The following ad plugins support integration with live streams:

- [FreeWheel Ad Plugin](#) on page 143
- [Google IMA Ad Plugin](#) on page 147
- [VAST and VPAID Ad Plugin](#) on page 171

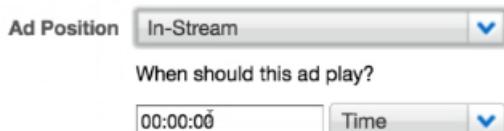
SUPPORTED AD POSITIONS

Ads for live streams are supported for the following ad positions:

- pre-roll
- mid-roll

CONFIGURING ADS FOR LIVE STREAMS

Setting up ads for live streams is very similar to setting up ads for a VOD asset. The key difference is that, for live streams, the duration is unknown. Therefore, when you configure the ad position for a mid-roll ad in the Backlot UI, you must specify **Time** (not **Percent**).



This configures the ad to play a set time after the video started playing.

DELIVERING MY ADS

My Ads (also known as "Ooyala Ads") provides a very basic framework for playing a video in front of the main content. It's ideally suited for sponsorships and video trailers-type content. Pre-roll ads/trailers may be included with videos using Player V4 (configured as My Ads in Backlot). This functionality does not require a separate plugin or the Ad Manager Framework - it is available as part of the core Ooyala player.

SUPPORTED FUNCTIONALITY

My Ads support was introduced in Player V4 version 4.11.13. The current release supports the following functionality:

- pre-roll or trailer video



- MP4 format (only). No other delivery format is supported with My Ads. Check with your Ooyala representative to make sure this delivery format is available in your processing profile.
- using either the `main_html5.min.js` or `bit_wrapper.min.js` video plugin (My Ads does not work with other video plugins)
- My Ads are supported on mobile devices (Android and iOS) beginning with Player V4 version 4.12.6.

LIMITATIONS IN THIS RELEASE

- Pre-roll ads are supported. Mid-roll or post-roll ads are not supported.
- My Ads cannot be combined with other pre-roll ads from another ad plugin.
- Only one ad per Ad Set is supported.

CONFIGURING MY ADS FOR PLAYER V4

You configure My Ads in Backlot.

1. Upload your video source to Backlot as an 'Ad' (see [Uploading a Video Advertisement](#)).
2. Create a new Ad Set (see [Creating an Ad Set](#)).
3. Add the 'Ad' asset to the ad set (see [Creating an Ad Set](#)). Specify the following parameters:

Note: For Player V4, Ad Sets can contain only one ad.

Description
Select My Ads .
Source
Specify the 'Ad" asset to use for this ad (as created earlier in step 1).
Content
Select pre-roll .
Position
Select either:
URL
• Use Ad's default Click URL
• Use Custom URL

Note: The following settings are not available in this release: Tracking URL and Ad Frequency options.

4. In Backlot, associate the Ad Set with the asset (see [Applying an Ad Set to a Video](#)).

EMBEDDING PLAYER V4 TO PLAY MY ADS

To embed a player that plays a My Ad ad, you specify only the most basic information (see [Basic Embedding Information](#) on page 88, because all configuration occurs in Backlot. If you play an asset associated with a My Ad ad, then ad playback is automatic.

Note: In this release, there are no page-level configuration parameters to override My Ads settings in Backlot.

The following code example shows a basic page embedding for a video associated with a My Ads ad. The video (`YOUR_ASSET_ID`) is configured to play a pre-roll My Ad automatically during playback. Both the video (`YOUR_ASSET_ID`) and the MP4 video must be hosted in the Backlot account associated with the specified `pcode`.

Note: To play Ooyala Ads, the current release requires one of the following video plugins: `bit_wrapper.min.js` or `main_html5.min.js`. Using any other video plugins will cause playback to fail.



```

<!DOCTYPE html>
<html>
<head>
    <title>My Test Player V4 Web Page</title>

    <script src="url_where_hosted/core.min.js"> </script>

    <script src="url_where_hosted/html5-skin.min.js"> </script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css"/>

    <script src="url_where_hosted/main_html5.min.js"> </script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        // Config option below contains the configuration setting for
        player skin. Change to your local config when necessary.
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": { "config": "url_where_hosted/skin.json"
            }
        };
        OO.ready(function( ) { window.pp = OO.Player.create("container",
        "YOUR_ASSET_ID", playerParam); })
    </script>
</body>
</html>

```

ABOUT ADS

The following topics provide background information about deploying ads in Player V4.

Ad Manager Framework

Ooyala's Ad Manager Framework (AMF) for Player V4 is a unified framework for HTML web. This framework allows ad servers to develop ad managers for Ooyala's HTML5 players. Ooyala has built ad plugins for major ad managers to integrate with the AMF.

PLUGIN SUPPORT FOR IAB VAST AND VPAID FORMATS

The Ooyala Player V4 ad plugins have varying support for IAB VAST and VPAID formats:

Plugin	Supports
Ooyala Pulse Ad Plugin on page 160	VAST 2.0, VAST 3.0, VPAID 1.0, and VPAID 2.0 formats
FreeWheel Ad Plugin on page 143	VAST XML
Google IMA Ad Plugin on page 147	Ads from DFP or AdSense ad servers that are generated in the VAST 2.0, VAST 3.0, VPAID 1.0, and VPAID 2.0 formats
VAST and VPAID Ad Plugin on page 171	VAST 2.0, VAST 3.0, and VPAID formats and LiveRail VAST ads



Note:

- If you are using Adap.tv or Brightroll as your ad server with VAST or VPAID, you should use our VAST and VPAID plugins.
- If you are using DFP or AdSense as your ad server with VAST or VPAID, you should use the Google IMA plugin.

CAPABILITIES OF THE AD MANAGER FRAMEWORK

- **Ad Manager Controller:** The AMF has a controller in the player, called the *Ad Manager Controller*, that handles basic functionality to allow ad managers to plug into Ooyala with less effort.
- **Ad Plugins:** The AMF allows any third-party ad plugin to manage the entire ad session, from ad request to ad playback. The plugin acts as a layer on top of the video player, pausing the video player when appropriate to display ads, and resuming the player again when the ad break is complete.
- **Run-time Management:** The framework has the ability to manage when an ad should play, handing off control to the ad manager.
- **UI Interactions:** The framework can handle UI interactions. For example, it can hide or show the control bar, and send control bar notifications to the ad manager.
- **APIs:** The framework provides a set of APIs to help partners develop customized modules for use in the Ooyala Player.
- **Extensibility:** Partners and publishers can extend this framework to easily build custom ad plugins with any ad source or provider. Using the Ad Manager Framework, partners can build custom skins, custom ad and analytics modules.

Ad Types

The following table describes the most common ad types in Player V4.

Type	Description
Linear Ad	A video ad that interrupts the main video asset.
Non-linear Overlay Ad	An ad (usually a small rectangle) that appears at the bottom of the video player that does not interrupt the main video asset.
Linear Overlay Ad	An overlay ad that takes over the whole player screen that stops (interrupts) the main video asset.
Podded Ads	A single ad request that returns a group of ads sequenced to play together back-to-back in an ad break. Podded ads will play wherever the ad break is scheduled.
Manually Podded Ads	Multiple ads (each with their own ad tag) that you have manually set to play at the same time in the video timeline. The end user has the same user experience as with podded ads.
Ad Bumper	A small video that gets inserted in the beginning of the video before your pre-roll ads play. This video can act as an introduction to your ads.
Companion Ad	An ad that appears outside of the video player (on the web page). The video player notifies the web page when to display the companion ad on the page.



Ad-Related Events

Ad-related events are exposed on the Player V4 message bus, including the events in the following table.

- Events related to ads all have the prefix OO.EVENTS.
- You can subscribe to the events to invoke your own callback functions when they occur.

For documentation on all Player V4 events, see apidocs.oyala.com.

Event	Description
WILL_PLAY_ADS	Ads for a given time slot will start playing now. This event may be followed by multiple AD POD_STARTED or AD POD_ENDED events.
ADS_PLAYED	Ads for a given time slot have finished playing.
AD POD_STARTED	A set of ads from a given ad manager has started playing. This event may be followed by multiple WILL_PLAY_SINGLE_AD or SINGLE_AD_PLAYED events.
AD POD_ENDED	A set of ads from a given ad manager has finished playing (ended).
WILL_PLAY_SINGLE_AD	A single linear ad will start playing now.
SINGLE_AD_PLAYED	A single linear ad has finished playing.
WILL_PLAY_NONLINEAR_AD	A nonlinear ad will start showing now.
NONLINEAR_AD_PLAYED	A nonlinear ad has finished showing.
HIDE_NONLINEAR_AD	A nonlinear ad will be hidden. This will not count against the clock timer for how long the ad should be shown.
SHOW_NONLINEAR_AD	A nonlinear ad will be shown.
WILL_SHOW_COMPANION_ADS	A companion ad should be played. The ad details are included in the event parameters.
ADS_ERROR	This indicates that an ad manager encountered an error. An ad manager could raise this event at any time.
ADS_CLICKED	The ad that is currently playing was clicked.
WILL_PAUSE_ADS	A pause was requested on the ad that is currently playing.
WILL_RESUME_ADS	A resume was requested on the ad that is currently playing.
PLAYHEAD_TIME_CHANGED	This event contains the current position and total duration of the ad that is currently playing. This is used to update the scrubber bar.



ANALYTICS IN PLAYER V4

To report playback statistics for analytics, use any of the analytics plugins that integrate with Player V4.

Note: Ooyala IQ Analytics is automatically included with Player V4. No separate plugin is needed. For details, see:

- [Ooyala IQ Analytics User Guide](#)
- [Ooyala IQ Analytics](#)

ANALYTICS PLUGINS

You can use the following analytics plugins.

- [Adobe Analytics \(Omniture\) Plugin](#) on page 190
- [comScore Analytics Plugin](#) on page 196
- [Conviva Analytics Plugin](#) on page 197
- [Google Analytics Plugin](#) on page 199
- [Nielsen Analytics Plugin](#) on page 204
- [YOUBORA Analytics Plugin](#) on page 207

These plugins were developed using the [Analytics Framework](#) on page 210.

RUN-TIME PLAYBACK FLOW

During playback, a playback event triggers notification about the event to the analytics provider.



REQUIREMENTS

If you want to track bitrate information for analytics purposes, you must use one of the following video plugins:

- Bitmovin plugin for DASH and HLS ([bit_wrapper.min.js](#)) (except HLS HTML5 in Safari)
- Akamai HD video plugin for Akamai packaged HDS ([akamaiHD_flash.min.js](#))

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

- OSMF Flash plugin for HDS ([osmf_flash.min.js](#))

Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

The HLS and MP4 Main video plugin ([main_html5.min.js](#)) does not report bitrate information.

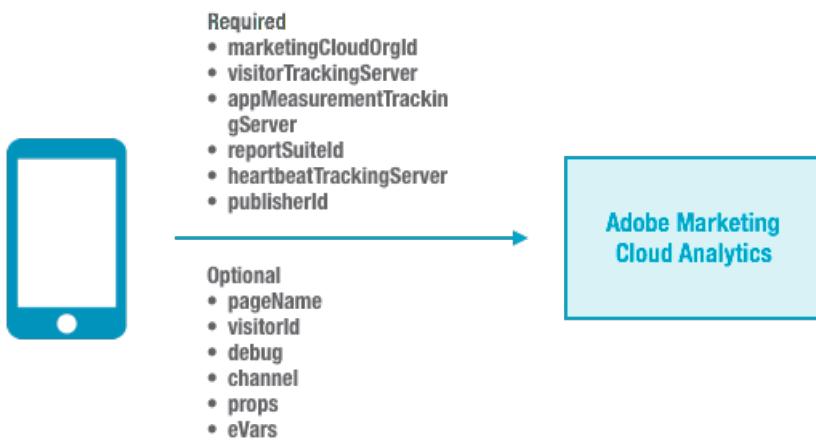


ADOBE ANALYTICS (OMNITURE) PLUGIN

Use the Player V4 Adobe Analytics plugin (`omniture.min.js`) to track Adobe Marketing Cloud analytics for your V4 Ooyala Player. This integration uses the Adobe Heartbeat SDK to publish video-related events, such as video plays, quartile video completion rates, and content metadata to your instance of Adobe Marketing Cloud.

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to the Adobe Marketing Cloud.



REQUIRED ADOBE PLUGINS

To use the Player V4 Adobe Analytics plugin (`omniture.min.js`), you must load several Adobe plugins related to the Adobe Heartbeat SDK.

- **VideoHeartbeat.min.js** is the main library that gathers analytics for video playback.
- **AppMeasurement.js** and **VisitorAPI.js** send account and tracking information to the Video Heartbeat SDK.

For more details on what these plugins do, go to:

- https://marketing.adobe.com/resources/help/en_US/mcvid/mcvid-setup-analytics.html
- https://marketing.adobe.com/resources/help/en_US/sc/implement/js_implementation.html

SUPPORTED PLAYERS

For this release, the Player V4 Adobe Analytics plugin supports integrating Adobe Analytics with the Ooyala HTML5 web player versions 4.4.8 and above.

PREREQUISITES

Prior to using the Player V4 Adobe Analytics plugin you must contact your Adobe representative and request the following account information for your integration with Ooyala Player V4:

- Report Suite ID
- Tracking Server
- Marketing Cloud Organization ID



- Publisher ID

You will pass this information to Ooyala along with other player parameters.

STEP 1: GET THE ADOBE ANALYTICS PLUGINS

Note: This applies only if you are self-hosting player resources, as described in [Hosting Player V4 Resources](#) on page 81.

1. Download the latest version of the following plugins. For the download URLs, see the [Ooyala Player V4 Release Notes](#) on page 235.
 - VideoHeartbeat.min.js
 - AppMeasurement.js
 - VisitorAPI.js
 - omniture.min.js
2. Host the above plugins as described in [Hosting Player V4 Resources](#) on page 81.

STEP 2: LOAD THE PLUGIN SCRIPTS

On the page where you want to run the player, add the following scripts:

- VideoHeartbeat.min.js is the main Adobe Heartbeat library that gathers analytics for video playback.
- AppMeasurement.js and VisitorAPI.js send account and tracking information to the Video Heartbeat SDK.
- omniture.min.js must be loaded last.

Be sure to load all of these plugins *after* you load core.min.js, and in the order specified.

Note: Wherever you see `url_where_hosted` in sample code, replace this (in your code) with the URL that points to where the resource is hosted. For a list of Ooyala-hosted paths, see [Ooyala-hosted Player V4 Resources](#) on page 77. The URL can point to a location on the same host (internal link) or on a separate host (prefixed with `http://` or `https://`). If you host resources yourself (see [Hosting Player V4 Resources](#) on page 81), be sure to check for any path dependencies within the files.

```
<script src="url_where_hosted/VideoHeartbeat.min.js"></script>
<script src="url_where_hosted/AppMeasurement.js"></script>
<script src="url_where_hosted/VisitorAPI.js"></script>
<script src="url_where_hosted/omniture.min.js"></script>
```

STEP 3: SPECIFY PAGE-LEVEL PARAMETERS

As with other Player V4 plugins, on the page where you want to run the player, you can specify page-level parameters within the `playerparams` variable that is passed in during player creation (see [Page-level Parameters for Player V4](#) on page 99).

Required Parameters

Specify the following **required** page-level parameters as a child of the `omniture` parameter. You will need to get the values of these parameters from your Adobe representative. Go to the following Adobe topic for details: [Implement the Marketing Cloud ID Service for Analytics](#).

Description

MarketingCloudOrgId
Adobe
Marketing
Cloud
Organization



Description

ID.
Go
to
https://marketing.adobe.com/resources/help/en_US/movid/movid-requirements.html
for
details.

You **to** TrackingServer
standard
or
secure
tracking
server
domain
(without
the
"http"
or
"https"
prefix).

You **Measurement** TrackingServer
standard
or
secure
tracking
server
domain
(without
the
"http"
or
"https"
prefix).

The **ReportSuiteId**
report
suite
or
report
suites
(multi-
suite
tagging)
that
you
wish
to
track.



Description

HeartbeatTrackingServer
standard
or
secure
tracking
server
domain
(without
the
"http"
or
"https"
prefix).

PublisherId
publisher
ID.

Specify
name
a
name
for
your
page.

UniqueId
identifier
for
each
visitor.

Optional Parameters

Specify the following optional page-level parameters. You can get the values of these parameters from your Adobe representative. Go to https://marketing.adobe.com/resources/help/en_US/mcvid/mcvid-setup-analytics.html for additional details.

Description

Enable
(YES)
or
disable
(NO)
debugging
information.

Specify
a
name
for
your
channel.

Traffic
variables
you
can



Description

use
to
report
on
custom
dimensions.
Keys
in
props
must
be
of
the
form
"prop#".
Go
to
https://marketing.adobe.com/resources/help/en_US/sc/implement/props_vs_eVars.html
for
details.

Conversion

variables

you
can
use
to
report
on
custom
dimensions.

Keys

in
eVars
must
be
of
the
form
"eVar#".

Go
to

https://marketing.adobe.com/resources/help/en_US/sc/



Description

implement/props_vs_eVars.html
for details.

Specifies at SSL whether the heartbeatTrackingServer pings via HTTPS ("heartbeatSSL":true) or HTTP ("heartbeatSSL":false, the default).

EXAMPLE

The following example shows how to load the Adobe Analytics plugins and use the page plugins. You will need to modify all items in bold with your data for this example to run.

```
<html>
<head>
    <title>Adobe Analytics Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css" />
    <!-- A Video Plugin is required. This example shows the Main Video Plugin -->
    <script src="url_where_hosted/main_html5.min.js"></script>
    <!-- Analytics Plugins -->
    <script src="url_where_hosted/VideoHeartbeat.min.js"></script>
    <script src="url_where_hosted/AppMeasurement.js"></script>
    <script src="url_where_hosted/VisitorAPI.js"></script>
    <script src="url_where_hosted/omniture.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            "omniture": {
                "marketingCloudOrgId": "YOUR_MARKETING_CLOUD_ORG_ID",
                "visitorTrackingServer": "TEST.sc.TEST.net",
                "appMeasurementTrackingServer": "TEST.sc.TEST.net",
                "reportSuiteId": "YOUR_REPORT_SUITE_ID",
                "pageName": "Test Page Name",
            }
        }
    </script>
</body>
</html>
```



```

        "visitorId": "test-vid",
        "debug": true,
        "channel": "Test Heartbeat Channel",
        "heartbeatTrackingServer": "TEST.en.TEST.net",
        "publisherId": "YOUR_PUBLISHER_ID",
        "props": {
            "prop1": "sports",
            "prop25": "football"
        },
        "eVars": {
            "eVar9": "en"
        }
    }
};

OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});

</script>
<body>
</html>

```

COMSCORE ANALYTICS PLUGIN

Use the Player V4 comScore analytics plugin to track [comScore](#) analytics for your V4 Ooyala Player in your comScore account. The comScore analytics plugin was developed by comScore using the Ooyala Analytics Framework. comScore will maintain and update this plugin.

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to comScore Analytics.



SUPPORTED PLAYERS

For this release, Ooyala supports integrating comScore with the HTML5 web player version 4.3.3 and above.

INTEGRATION TASKS

If you would like to use the comScore plugin with Player V4, please contact Ooyala [Tech Support](#) and file a support ticket requesting details on how to enable and use the plugin.



CONVIVA ANALYTICS PLUGIN

Use the Player V4 Conviva analytics plugins (`conviva-core-sdk.min.js` and `conviva.min.js`) to track [Conviva](#) analytics for Ooyala Player V4.

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to Conviva Analytics.



SUPPORTED PLAYERS

The Conviva plugin supports integrating Conviva with the Ooyala HTML5 web player versions 4.12.6 and above. The Conviva plugin works with all Player V4 video plugins.

REQUIREMENTS

If you want to track bitrate information for analytics purposes, you must use one of the following video plugins:

- Bitmovin plugin for DASH and HLS ([`bit_wrapper.min.js`](#)) (except HLS HTML5 in Safari)
- Akamai HD video plugin for Akamai packaged HDS ([`akamaiHD_flash.min.js`](#))

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

- OSMF Flash plugin for HDS ([`osmf_flash.min.js`](#))

Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

The HLS and MP4 Main video plugin ([`main_html5.min.js`](#)) does not report bitrate information.

PREREQUISITES

Prior to using the Conviva plugin, you must contact your Conviva representative and request the following account information:

- your Conviva `customerKey`
- the `gatewayUrl` to use for your account

You will need to specify these values when you embed Player V4 on a web page.

WHERE TO FIND THE PLUGINS

The latest version of the plugins are posted in the following path, where **LATEST_PLAYER_VERSION** is the latest player version (4.12.6 and above):

```
//player.ooyala.com/static/v4/stable/LATEST_PLAYER_VERSION/analytics-plugin/  
conviva-core-sdk.min.js
```



```
//player.ooyala.com/static/v4/stable/LATEST_PLAYER_VERSION/analytics-plugin/  
conviva.min.js
```

INTEGRATION STEPS

To integrate using the Conviva plugins for Player V4:

Step 1: Download and Host the Latest Conviva Plugins

Note: This applies only if you are self-hosting player resources, as described in [Hosting Player V4 Resources](#) on page 81.

1. Download the latest version of the following plugins, where **LATEST_PLAYER_VERSION** is the latest player version (4.12.6 and above).

```
player.ooyala.com/static/v4/stable/LATEST_PLAYER_VERSION/analytics-plugin/  
conviva-core-sdk.min.js  
player.ooyala.com/static/v4/stable/LATEST_PLAYER_VERSION/analytics-plugin/  
conviva.min.js
```

2. If you are self-hosting player resources, host the Conviva plugin as described in [Hosting Player V4 Resources](#) on page 81.

Step 2: Add Conviva Integration on the Web Page

Add the following scripts to the page on which you load the player. Be sure to load these plugins *after* you load core.min.js.

```
<script src="url_where_hosted/conviva-core-sdk.min.js"></script>  
<script src="url_where_hosted/conviva.min.js"></script>
```

Note: You must load conviva-core-sdk.min.js *before* conviva.min.js.

Step 3: Create the Player Using the Create Function

Use the Player V4 OO.Player.create function to create the player. For information on the create function, see [apidocs.ooyala.com](#).

Step 4: Specify Page-level Parameters for the Conviva Plugin

Specify the following page-level parameters.

Parameter	Type	Description
Required		
gatewayUrl	String	URL used to report player statistics to Conviva Analytics. Issued by Conviva.
customerKey	String	Customer key associated with your Conviva account. Issued by Conviva.
applicationName	String	Name of the application to report to Conviva.
Optional		
customMetadata	String	Custom metadata to report to Conviva. List of key/value pairs that you define. Examples: <ul style="list-style-type: none">• user statistics the site can track (age, gender, etc.)• other metadata (such as site category) Custom metadata is passed through to Conviva.



EXAMPLE

```
<html>
<head>
    <title>Conviva Analytics Plugin Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
advertising need to be loaded separately -->
    <script src= "url_where_hosted/core.min.js" ></script>
    <script src= "url_where_hosted/html5-skin.min.js" ></script>
    <link rel= "stylesheet" href= "url_where_hosted/html5-skin.min.css" />
    <!-- A Video Plugin is required. This example shows the Main Video
Plugin -->
    <script src= "url_where_hosted/main_html5.min.js" ></script>
    <!-- Analytics Plugins -->
    <script src= "url_where_hosted/conviva.min.js" ></script>
</head>
<body>
    <div id= "container" style= "width:640px; height:360px;" ></div>
    <script>
        var playerParam = {
            "PCODE" : "YOUR_PCODE" ,
            "playerBrandingId" : "YOUR_PLAYER_ID" ,
            "skin" : {
                // Config contains the configuration setting for the player
                skin.
                // Change to your local config when necessary.
                "config" : "url_where_hosted/skin.json"
            },
            "conviva":{
                "gatewayUrl": "https://my-test.testonly.conviva.com",
                "customerKey": "abcde12345customerkey",
                "applicationName": "testAppName",
                "customMetadata": {"testKey": "testValue",
"account": "myAccount", "test": "true"}
            }
        };
        OO.ready( function () {
            window.pp = OO.Player.create( "container" , "YOUR_ASSET_ID" ,
playerParam);
        });
    </script>
</body>
</html>
```

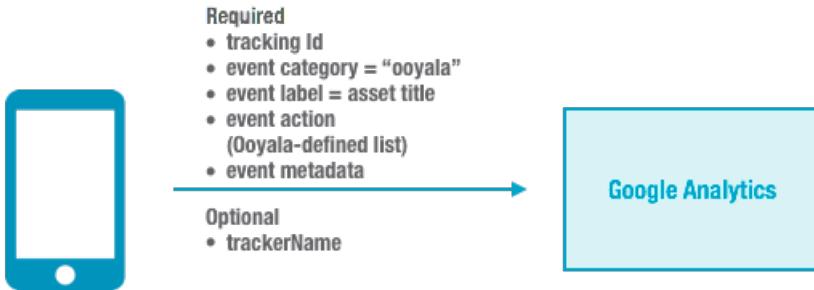
GOOGLE ANALYTICS PLUGIN

Use the Player V4 Google Analytics plugin to track statistics for your Ooyala Player V4 in your Google account Analytics. For general information about Google Analytics, see the [Google Analytics](#) site.

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to Google Analytics.





SUPPORTED PLAYERS

The Player V4 Google Analytics plugin supports integrating Google Analytics with the Ooyala HTML5 web player versions 4.7.9 and above.

PREREQUISITES

Prior to using the Player V4 Google Analytics plugin, you must have a Google Analytics account. To that account, you must have added at least one property for which you want to collect data, and know your Tracking ID associated with each property. For more information, see the Google documentation: [Get Started with Analytics](#).

STEP 1: GET THE LATEST GOOGLE ANALYTICS PLUGIN

Note: This applies only if you are self-hosting player resources, as described in [Hosting Player V4 Resources](#) on page 81.

1. Download the latest version of the googleAnalytics.min.js plugin. For the download URLs, see the [Ooyala Player V4 Release Notes](#) on page 235.
2. Host this plugin like you host other Player V4 plugins.

STEP 2: LOAD GOOGLE ANALYTICS ON YOUR WEB PAGE

Begin by loading Google Analytics on the web page from which you run Player V4. You have three options.

Note: For UA-XXXXX-Y, you must specify the proper tracking ID associated with the property for which you want to collect data.

Option A: Load Google Analytics (analytics.js)

With this option, you obtain a snippet from the Analytics interface (refer to the Tracking Code page under the Admin tab), and then insert this code into the page - ideally within the <head> tag. This snippet contains your unique "Tracking ID". Here is an example code snippet:

```

<!-- Google Analytics -->
<script>
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||
    function(){
        (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
        Date();a=s.createElement(o),
        m=s.getElementsByTagName(o)
        [0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','https://www.google-analytics.com/
    analytics.js','ga');

    ga('create', 'UA-XXXXX-Y', 'auto');
    ga('send', 'pageview');
</script>

```



```
<!-- End Google Analytics -->
```

Note: If you copied the snippet from your account Admin page, the value of UA-XXXXX-Y should already be set to the proper tracking ID.

For additional information about analytics.js, see the [Google Analytics documentation](#).

Option B: Load Google Analytics (Legacy) (ga.js)

With this option, you obtain the Legacy Google Analytics (ga.js) code snippet (at <https://developers.google.com/analytics/devguides/collection/analyticsjs/>), and then insert this code into the page - ideally within the <head> tag. Here is an example code snippet:

```
<script type="text/javascript">
var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-XXXXX-X']);
_gaq.push(['_trackPageview']);

(function() {
var ga = document.createElement('script'); ga.type = 'text/javascript';
ga.async = true;
ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://'
www') + '.google-analytics.com/ga.js';
var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
})();</script>
```

For additional information about ga.js, see the [Legacy Google Analytics documentation](#).

Option C: Use the Google Tag Manager

With this option, you obtain the code snippet for the Google Tag Manager from the "Install GTM" link under the "Admin" Tab. Here is an example code snippet:

```
<!-- Google Tag Manager -->
<noscript><iframe src="//www.googletagmanager.com/ns.html?id=GTM-M8ML7S"
height="0" width="0" style="display:none;visibility:hidden"></iframe></
noscript>
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.createElement(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?l+'_'+i:j.async=true;j.src=
'//www.googletagmanager.com/gtm.js?id='+i
+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-M8ML7S');</script>
<!-- End Google Tag Manager -->
```

Insert this code immediately after the opening <body> tag on your page.

Note: You need to add the following custom variables to the Tag Configuration:

- OOLabel (Variable Type: **Data Layer Variable**, Data Layer Variable Name: **label**)
- OOAction (Variable Type: **Data Layer Variable**, Data Layer Variable Name: **action**)
- OOCategory (Variable Type: **Data Layer Variable**, Data Layer Variable Name: **category**)

You need to implement an additional tag in case you want to see audience stats (such as page visits or current users) in addition to events.

For additional information, see the [Google Tag Assistant](#).



STEP 3: LOAD THE OYALA GOOGLE ANALYTICS PLUGIN ON YOUR WEB PAGE

Load the `url_where_hosted/googleAnalytics.min.js` script on the page where you want to launch the player. Note that this plugin requires Google Analytics to be set up already on the page (using one of the three methods described above).

STEP 4: SPECIFY A TRACKER (OPTIONAL)

As with other Player V4 plugins, on the page where you want to run the player, you can specify a page-level parameter within the `playerparams` variable that is passed in during player creation (see [Page-level Parameters for Player V4](#) on page 99). The tracker will target events sent by the plugin to the provided tracker name (`trackerName`). This tracker name must be equal to the tracker name defined in the `ga('create')` function call used to set up the tracker. If you want a tracker name for the Ooyala Debug Page, this page-level parameter must be set and be equal to the tracker name provided in the `url` parameter "`ga_tracker_name`".

Example page level parameter:

```
"googleAnalytics": {  
    "trackerName": "myTrackerName"  
}
```

For more information, see the following Google topics:

- [Creating Trackers](#)
- [Multiple tracking codes on web pages](#)

PLAYER V4 EVENTS FOR THE GOOGLE ANALYTICS PLUGIN

The Player V4 Google Analytics plugin reports run-time playback events.

Event Category

Events sent by the plugin have the event category "Ooyala".

Event Actions

Events sent by the plugin have the following event actions:

Event	Sent When
playbackStarted	<ul style="list-style-type: none">• Playback has started.• Playback has resumed. This is fired when returning from pauses and buffers.
playProgressStarted	Playback has reached 0% of the total duration. This event maps to the Video Starts metric in Ooyala IQ (see Common Metric Definitions and Examples).
playProgressQuarter	Playback has reached 25% of the total duration.
playProgressHalf	Playback has reached 50% of the total duration.
playProgressThreeQuarters	Playback has reached 75% of the total duration.
playProgressEnd	Playback has reached 100% of the total duration.
contentReady	The content is ready to be played.
adPlaybackStarted	An ad playback starts.
adPlaybackFinished	An ad playback finishes.
playbackPaused	Playback is paused.



Event Labels

For events sent by the Google Analytics plugin, the event label is the title of the content being played.

VIEWING ANALYTICS

In the Google Analytics GUI, go to the Reporting tab to view your analytics. Useful metrics include:

To View	Go To	Notes
Go To		
Notes		
Live Events	Reporting -> Real Time -> Events	Real time event notifications from active users. You can view these events as part of a live-updating graph to the right or as a list underneath the graph.
Audience Overview	Reporting -> Audience -> Overview	General overview of session related analytics over time periods of your choosing.
Events	Reporting -> Behavior -> Events	Various ways to view the event distribution sent from your users.

EXAMPLE

```
<html>
<head>
    <title>Google Analytics Plugin Example</title>
    <!-- Google Analytics -->
    <script>
        (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
            (i[r].q=i[r].q||[]).push(arguments),i[r].l=1*new Date();a=s.createElement(o),
            m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
        })(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

        ga('create', 'UA-XXXXX-Y', 'auto');
        ga('send', 'pageview');
    </script>
    <!-- End Google Analytics -->

    <!-- V4 JS core is required. Plugins such as skin, discovery and Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css" />
    <!-- A Video Plugin is required. This example shows the Main Video Plugin -->
    <script src="url_where_hosted/main_html5.min.js"></script>
    <!-- Google Analytics plugin -->
    <script src="//player.oyala.com/static/v4/production/analytics-plugin/googleAnalytics.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {

```



```

        // Config contains the configuration setting for player
        skin. Change to your local config when necessary.
        "config": "url_where_hosted/skin.json"
    },
    // trackerName is optional
    "googleAnalytics":{
        "trackerName":"myTrackerName"
    }
};
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>
<body>
</html>

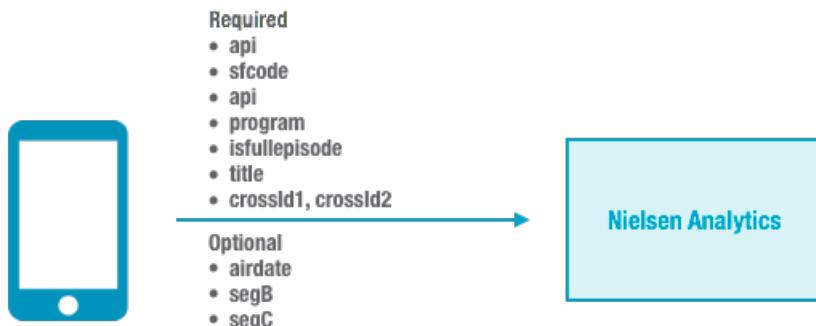
```

NIELSEN ANALYTICS PLUGIN

Use the Player V4 Nielsen analytics plugin (`Nielsen.min.js`) to track [Nielsen Analytics](#) for Ooyala Player V4.

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to Nielsen Analytics.



SUPPORTED PLAYERS

The Nielsen Analytics plugin supports Nielsen Analytics integration with the Ooyala HTML5 web player versions 4.8.5 and above.

Note: The Ooyala mobile SDKs for iOS and Android already support Nielsen integration. For details, see [Integration with Nielsen Analytics](#).

PREREQUISITES

Before using the Nielsen plugin for Player V4, contact your Nielsen representative and ask for your `apid` (a unique ID provided by Nielsen). You will need to specify this value when you embed Player V4 on a web page.



STEP 1: GET THE LATEST NIELSEN PLUGIN

Note: This applies only if you are self-hosting player resources, as described in [Hosting Player V4 Resources](#) on page 81.

1. Download the latest version of the Nielsen.min.js plugin. For the download URL, see the [Ooyala Player V4 Release Notes](#) on page 235.
2. Host this plugin like you host other Player V4 plugins.

STEP 2: ADD NIELSEN INTEGRATION ON THE WEB PAGE

Add the following script to the page on which you load the player. Be sure to load this plugin *after* you load core.min.js.

```
<script src="url_where_hosted/Nielsen.min.js"></script>
```

STEP 3 (OPTIONAL): LOAD THE NIELSEN SDK ON THE WEB PAGE

Load one of the following versions of the Nielsen SDK on the page on which you are loading the player. If you do not load one of these options, the Nielsen plugin will load the Global version by default.

Version	Link
Global	http://cdn-gl.imrworldwide.com/novms/js/2/ggcmb510.js
Certificate	http://secure-dcr-cert.imrworldwide.com/novms/js/2/ggcmb510.js
Production	http://secure-dcr.imrworldwide.com/novms/js/2/ggcmb510.js

STEP 4: SPECIFY PAGE-LEVEL PARAMETERS FOR THE NIELSEN PLUGIN

As with other Player V4 plugins, on the page where you want to run the player, you can specify the following page-level parameters within the `playerparams` variable that is passed in during player creation (see [Page-level Parameters for Player V4](#) on page 99). For details on these parameters, go to the [Nielsen SDK Documentation](#).

Required Parameters

Parameter	Description
apid	A unique ID provided by Nielsen.
sfcode	Location of collections environment. <ul style="list-style-type: none">• Use "dcr-cert" for testing.• Use "dcr" for production.
apn	Unique string identifying your player/site.
program	Program name.
isfullepisode	Full episode flag (used to distinguish between short and long form content).
title	Title for the video asset.
crossId1	Standard episode ID (should be unique).
crossId2	Identifies the content originator.

Optional Parameters



Parameter	Description
airdate	Original (local) air date and time.
segB	Segment B (custom category breakout).
segC	Segment C (custom category breakout).

STEP 5: SEND A TEST PAGE TO NIELSEN

Send a test page to Nielsen so that they can verify and certify the page.

EXAMPLE

```

<html>
<head>
    <title>Nielsen Analytics Plugin Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css" />
    <!-- A Video Plugin is required. This example shows the Main Video
Plugin -->
    <script src="url_where_hosted/main_html5.min.js"></script>
    <!-- Analytics Plugins -->
    <script src="url_where_hosted/Nielsen.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player
                // skin. Change to your local config when necessary.
                "config": "url_where_hosted/skin.json"
            },
            "Nielsen": {
                "apid": "YOUR_APID",
                "sfcode": "dcr-cert",
                "apn": "Ooyala V4",
                "program": "myProgram",
                "isfullepsode": "N",
                "title": "My Title",
                "crossId1": "EP01S9S290015",
                "crossId2": "ABC",
                "airdate": "20150420 21:00:00",
                "segB": "Comedy",
                "segC": "Drama"
            }
        };
        OO.ready(function() {
            window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
        });
    </script>
</body>
</html>
```

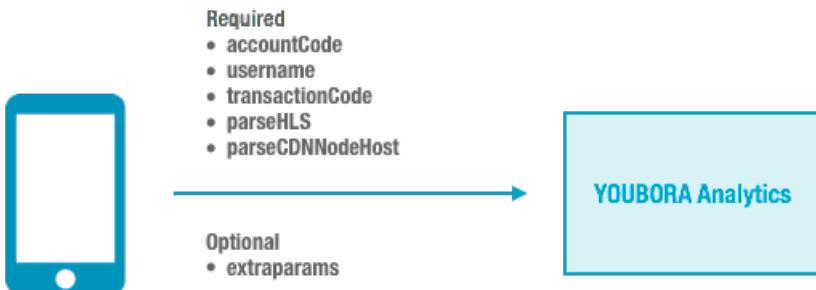


YOUTORA ANALYTICS PLUGIN

Use the Player V4 [Nice People At Work](#) (NPAW) YOUTORA plugin (`youbora.min.js`) to track NPAW YOUTORA quality of service (QoS) analytics for Ooyala Player V4. For general information about YOUTORA integration, see [Ooyala QoS Solution Powered by NPAW's YOUTORA](#).

RUN-TIME FLOW DURING PLAYBACK

During playback, a playback event triggers notification about the event to YOUTORA Analytics.



SUPPORTED PLAYERS

The Player V4 YOUTORA Analytics plugin supports integrating YOUTORA Analytics with the Ooyala HTML5 web player versions 4.6.9 and above.

PREREQUISITES

Prior to enabling the Ooyala QoS Solution powered by YOUTORA, you must have a customer relationship with [NPAW](#) with an account for YOUTORA.

REQUIREMENTS

If you want to track bitrate information for analytics purposes, you must use one of the following video plugins:

- Bitmovin plugin for DASH and HLS ([`bit_wrapper.min.js`](#)) (except HLS HTML5 in Safari)
- Akamai HD video plugin for Akamai packaged HDS ([`akamaiHD_flash.min.js`](#))

Note: The Akamai HD Video Plugin for Akamai Packaged HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

- OSMF Flash plugin for HDS ([`osmf_flash.min.js`](#))

Note: The OSMF Flash Video Plugin for HDS for Player V4 has been deprecated and is scheduled to be disabled. For details and alternatives, see the [OVP Release Notes](#).

The HLS and MP4 Main video plugin ([`main_html5.min.js`](#)) does not report bitrate information.

STEP 1: LOAD THE YOUTORA PLUGIN

To add the YOUTORA plugin to your player, use one of the following URLs to load the plugin (in addition to the player files):

- <http://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
- <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>



Because NPAW maintains the YOUBORA Analytics plugin for Player V4, use one of these links to load the latest version.

Note: Unlike other Player V4 plugins, it is not recommended that you try to host this plugin yourself.

Here are some examples of how to load this in the <head> section of your web page:

How to Load Using	Example
Hardcoded HTTP	<pre><script src="http://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js"></script></pre>
Hardcoded HTTPS	<pre><script src="https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js"></script></pre>
Protocol of the page from which it is being called	<pre><script src="//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js"></script></pre>

STEP 2: SPECIFY PAGE-LEVEL PARAMETERS

As with other Player V4 plugins, you need to specify the YOUBORA parameters within the `playerparams` variable that is passed in during player creation (see [Page-level Parameters for Player V4](#) on page 99).

```
var playerParams = {
  "youbora": {
    // accountCode is required
    "accountCode": "myCode",
    // other optional YOUBORA params
  }
}
OO.ready(function() {
  window.pp = OO.Player.create('playerDiv', 'myVideoEmbedCode',
  playerParams);
})
```

You can specify the following parameters. Refer to the [YOUBORA](#) product documentation for additional information about each parameter.

Parameter	Data Type	Description
accountCode	{string}	Required. Provided by NicePeopleAtWork. Specifies to YOUBORA Analytics the customer account to which the data is sent.
username	{string}	User account associated with your system. Used to identify your users within YOUBORA Analytics. For details, see the YOUBORA product documentation.
transaction	{string}	Specifies to YOUBORA Analytics a transaction code. This is a general purpose string that can be used to filter users when sorting through collected data on the YOUBORA Analytics platform. Typically, NPAW recommends using a parameter the customer can correlate with its content management system (CMS) or paid transactions. Examples: purchaseCode, sessionCode, and so on. For details, see the YOUBORA product documentation.
parseHLS	{boolean}	Configures YOUBORA's parsing HLS algorithm. One of the following values: <ul style="list-style-type: none">• <code>parseHLS = true</code> enables the parsing HLS algorithm• <code>parseHLS = false</code> disables the parsing HLS algorithm For details, see the YOUBORA product documentation.



Parameter	Data Type	Description
parseCDNNodeHost	boolean	<p>Configures YOUBORA's CDN Node detection. One of the following values:</p> <ul style="list-style-type: none"> • parseCDNNodeHost = true enables CDN Node detection • parseCDNNodeHost = false disables CDN Node detection <p>For details, see the YOUBORA product documentation.</p>
extraParams	object	<p>Supplemental, custom parameters to pass into the plugin to help with filtering the gathered data in YOUBORA Analytics. You can specify up to ten parameters using the keyname pattern paramN, where N is a number 1 - 10. The values represent the data that you want to send to YOUBORA. On YOUBORA's site, you map what each key means (for example, param1 means one thing, param2 means something else, and so on).</p> <pre>extraParams: { param1: 'param1', param2: 'param2', param3: 'param3', param4: 'param4', param5: 'param5', param6: 'param6', param7: 'param7', param8: 'param8', param9: 'param9', param10: 'param10' }</pre>

For details, see the [YOUBORA](#) product documentation.

EXAMPLE

```
<html>
<head>
    <title>NPAW YOUBORA Example</title>
    <!-- V4 JS core is required. Plugins such as skin, discovery and
Advertising need to be loaded separately -->
    <script src="url_where_hosted/core.min.js"></script>
    <script src="url_where_hosted/html5-skin.min.js"></script>
    <link rel="stylesheet" href="url_where_hosted/html5-skin.min.css" />
    <!-- A Video Plugin is required. This example shows the Bitmovin Plugin -->
    <script src="url_where_hosted/bit_wrapper.min.js"></script>
    <!-- YOUBORA plugin -->
    <script src="//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/
sp.min.js"></script>
</head>
<body>
    <div id="container" style="width:640px; height:360px;"></div>
    <script>
        var playerParam = {
            debug:true,
            "PCODE": "YOUR_PCODE",
            "playerBrandingId": "YOUR_PLAYER_ID",
            "skin": {
                // Config contains the configuration setting for player skin.
                Change to your local config when necessary.
                "config": "//player.ooyala.com/static/v4/candidate/latest/skin-
plugin/skin.json",
            },
        }
    </script>

```



```

        "youbora": {
            "accountCode": "YOUR_ACCOUNT_CODE",
            "username": "user1",
            "transactionCode": "payingCustomer",
            "parseHLS": false,
            "parseCDNNodeHost": false,
            "extraParams": {
                "param1": "myData1",
                "param2": "myData2"
            }
        }
    };
OO.ready(function() {
    window.pp = OO.Player.create("container", "YOUR_ASSET_ID",
playerParam);
});
</script>
</body>
</html>

```

ANALYTICS FRAMEWORK

The Analytics Framework is a unified framework for HTML web. The framework allows analytics providers to develop analytics plugins for Ooyala's HTML5 players. Partners and publishers can extend this framework to easily build custom analytics plugins with any analytics source or provider, allowing you to integrate the Ooyala Player V4 with your other third-party analytics providers.

PLUGINS BUILT USING THE ANALYTICS FRAMEWORK

- Ooyala used the Analytics Framework to build [a plugin](#) to integrate Adobe Analytics (Omniture) capabilities with Player V4.
- [Nice People At Work](#) used the Analytics Framework to build [a plugin](#) to integrate YOUBORA analytics capabilities with Player V4.
- [comScore](#) used the Analytics Framework to build [a plugin](#) to integrate comScore analytics capabilities with Player V4.

OPEN SOURCE HTML JAVASCRIPT ANALYTICS FRAMEWORK API

The open source HTML5 JavaScript Analytics Framework API allows third-party analytics providers to plug into Player V4. For more information, see [html5-analytics-plugins on Github](#) and [Analytics Framework API](#).



PLAYER V4 JAVASCRIPT API

The Player V4 API is a JavaScript API for customizing your player functionality and UI elements, along with message bus event handling. See apidocs.ooyala.com for the Player V4 JavaScript API Reference.

ABOUT THE PLAYER V4 JAVASCRIPT API

This topic provides an overview of key concepts for the Player V4 JavaScript API.

API REFERENCE

See apidocs.ooyala.com for the Player V4 JavaScript API Reference.

PLAYER V4 API FUNCTIONS

Once you can instantiate a new Player V4 object:

```
new Player()
```

you can manage the player instance using the functions described at http://apidocs.ooyala.com/player_v4_js/OO.Player.html.

PLAYER V4 STATES

A running player undergoes various states during its lifecycle. For example:

- The video and metadata are **loading** into the player.
- Video playback is **ready** to begin.
- The player is currently **playing** video content.
- The player is currently **paused** (after playback had begun).
- The player is **buffering** - playback has stopped because it does not have enough video data to continue and is downloading more.
- The player has encountered an **error** that prevents playback of the video.

The current player state determines which functionality is available in the current context. For example, the play operation is available when the player is in the **ready** state, and the pause operation is available when it is in the **playing** state. State changes are triggered by events (see below). For a list of player states, see http://apidocs.ooyala.com/player_v4_js/OO.STATE.html.

PLAYER V4 EVENTS

Player run-time events can be triggered in either of the following ways:

- manually by end user actions (for example, clicking the play or pause button)
- programmatically via an API call

Here are some examples of common playback events:

- playback has started
- playback has completed
- an ad has started playing
- one or more ads have completed playing
- an error has occurred



Events can trigger changes in the player state. For example, a user clicking the play button fires the "play has started" event and changes the player state to **playing**. For a list of player events, see http://apidocs.ooyala.com/player_v4_js/OO.EVENTS.html.

PLAYER V4 ERRORS

For Player V4 errors and error handling, see [Handling Errors for Player V4 Using JavaScript](#) on page 227 and the following topics:

- http://apidocs.ooyala.com/player_v4_js/OO.ERROR.html
- http://apidocs.ooyala.com/player_v4_js/OO.ERROR.API.html
- http://apidocs.ooyala.com/player_v4_js/OO.ERROR.API.SAS.html
- http://apidocs.ooyala.com/player_v4_js/OO.ERROR.PLAYBACK.html

PROGRAMMING CLOSED CAPTIONING USING PLAYER V4 JAVASCRIPT

Closed Captioning (CC) APIs enable you to get/set supported closed captions and localize the Player UI language.

The Ooyala Player V4 supports closed captioning (also known as subtitles) for VOD and Live Streams. Ooyala supports closed captions for VOD in our iOS and Android SDKs and for Player V4 (HTML5). Live stream closed captions are supported for Live HLS streams on Safari.

JavaScript Closed Captioning APIs enable you to get supported closed captions and set closed caption languages. In live streaming mode, the closed caption languages are derived from the stream itself. The Closed Caption functions are:

- `getCurrentItemClosedCaptionsLanguages()`
- `setClosedCaptionsLanguage()`

You can review the list of supported closed caption languages in the topic [Supported Closed Caption Languages](#).

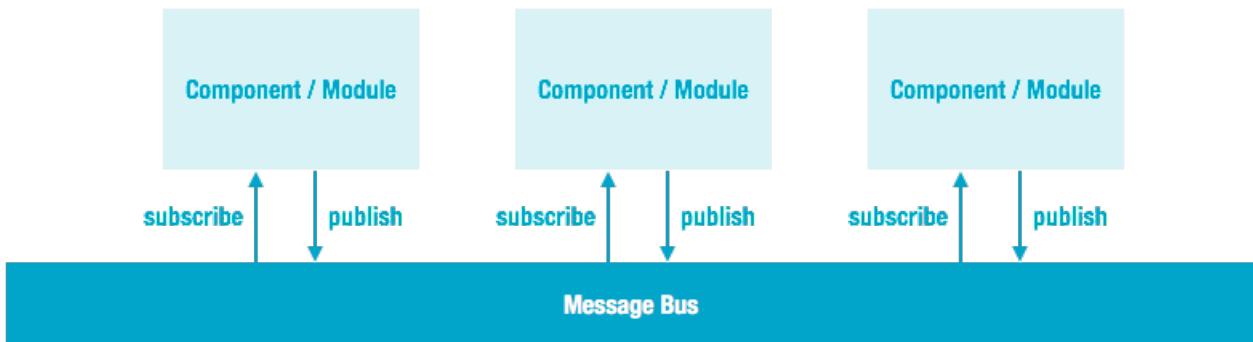
Note: When the closed caption file is uploaded, not played, Ooyala uses a strict XML parser for handling Closed Captioning. Your closed caption files at minimum need to work with your browser's parser.

EVENT MODEL FOR PLAYER V4

Player V4 incorporates the use of a message bus to handle events.

The Ooyala Player V4 architecture enables you to use a message bus to handle events. The message bus enables publish/subscribe messaging and simplifies interaction with the player. Any number of modules can subscribe to messages and publish messages. As illustrated in the following figure, the message bus routes published messages to modules that have subscribed to those messages.





Each player is independent with regards to the message bus. There is one message bus per each player. Each component decides which message it wants to communicate to or listen to via the message bus.

Message Bus API

Communicate with the message bus using publish, subscribe, and other functions.

The Ooyala Player exposes several functions for communication with the message bus. See the [Player JavaScript API Reference](#) for more information.

Player V4 Message Bus Events

Use message bus events to subscribe to or publish player events from video to ad playback.

All Ooyala APIs are shown at [Ooyala API Documentation](#).

The Player V4 JavaScript APIs, as well as the player events, are shown at [Player JavaScript API Reference](#).

The player events are default events that are published by the event bus. Your modules can subscribe to any and all of these events. To get a list of all these default events that are published by the message bus, you can use the Message Bus `subscribe()` function:

```
player.mb.subscribe("*", 'example', function(eventName) {});
```

Event Publish and Subscribe

You can publish and subscribe to messages for event handling.

CREATE, CONNECT, CHECK, AND LOAD

The player message bus provides a means for modules to publish and subscribe to messages for event handling. It uses `OO.Player.create` and `onCreate` to create a message bus, connect modules to the message bus, check for 3rd party modules and connect them and then load everything.

`OO.Player.create`

When called, the `OO.Player.create()` function:

- Creates a new message bus.
- Goes over the list of registered modules and creates all of them.
- Connects all the registered modules to the message bus.

`onCreate`

When called, `function onCreate(player)`:



- Checks for any additional modules (custom, 3rd party or other type).
- Enables these additional modules to connect to the message bus.
- Sends a message to the message bus signaling each module to start up.

You must call `onCreate()` before anything can happen; otherwise, the existing and additional or third-party modules are not connected to the message bus and are not initialized.

Listening to a Message Bus Event

You can use the message bus to listen to a single event or multiple events.

The following example illustrates how to use the message bus to listen for an event and to use `addDependent()` to block a pause event and display a confirmation message.

You can use the `addDependent()` function to block events on other events. In the following example, we use the `addDependent()` function to block a pause function so that when a pause button is pressed, we can display a message box and ask for confirmation. For more information about using the `addDependent()` function, see the [Player JavaScript API Reference](#).

To develop the head element and the logic for presenting and creating the Player, see [Basic Tutorial for Player V4](#) on page 53.

To learn about event handling, see [Event Model for Player V4](#) on page 212. For a list of all events, see the [Player JavaScript API Reference](#).

Suppose you would like to customize how your web page responds to user interaction with the video player. In this example we intercept the pause and play events. When the user clicks the pause control, a confirmation dialog appears before the video is paused. When the user clicks the play or pause control, a textbox displays the new state of the player.

The purpose of this web page is to provide the user with an option to switch to fullscreen mode, so the design will include:

- A head element containing a `script` tag specifying the loading and initialization request.
- A body element containing:
 - UI presentation and layout for the Player container.
 - Event listeners that monitor and respond to player events.
 - A `script` tag that creates the Player.
 - An HTML `input` element that displays text indicating whether the video is playing or paused.

Follow these steps to build your web page:

- [Step 1: Listening to Multiple Events](#) on page 217

See [Complete Example](#) on page 215 to view the code for the entire web page.

STEP 1: LISTENING TO MULTIPLE EVENTS

Sometimes it is useful to listen and respond to multiple events at once. One way to do this is to subscribe to all events. The [Event Model for Player V4](#) on page 212 provides you with a message bus, accessed via the created Player object's `mb` object. In this case we will use the message bus `subscribe()` function specifying a wildcard ("*") for the event name:

```
function onCreate(player) {
    /* Subscribe to all events:
     * this allows you to create logic based on multiple events. */
    player.mb.subscribe("*", 'example', function(eventName) {
});
```



```
}
```

Now that we are listening for every possible event, we can include logic based on multiple events within the callback function via its `eventName` parameter. In this example we write all events, except for downloading and playhead time changed events (`OO.EVENTS.DOWNLOADING` and `OO.EVENTS.PLAYHEAD_TIME_CHANGED`), to the JavaScript console log:

```
function onCreate(player) {  
  
    /*  
     * Subscribe to all events:  
     * this allows you to create logic  
     * based on multiple events.  
     */  
    player.mb.subscribe("*", "example", function(eventName) {  
  
        /* Write all events, except for downloading and playhead time  
        changed events,  
         * to the JavaScript console: */  
        if (eventName != OO.EVENTS.DOWNLOADING &&  
            eventName != OO.EVENTS.PLAYHEAD_TIME_CHANGED)  
            console.log(eventName);  
    });  
}
```

COMPLETE EXAMPLE

You now have a complete working example of a web page that loads multiple Ooyala V4 Players, responds to user interaction with the video player by intercepting pause and play events with a confirmation dialog and textbox that displays messages, and creates the player:

```
<!DOCTYPE html>  
<html>  
    <head>  
        <link rel="stylesheet" href="url_where_hosted/skin-plugin/html5-  
skin.min.css"/>  
        <script language="javascript" src="url_where_hosted/core.min.js"></  
script>  
        <script type="text/javascript" src="url_where_hosted/video-plugin/  
main_html5.min.js"></script>  
        <script type="text/javascript" src="url_where_hosted/skin-plugin/  
html5-skin.min.js"></script>  
    </head>  
    <body>  
        <div id='container' style='width:640px;height:480px'></div>  
        <script>  
            /*  
             * Use this to control playing state messages.  
             */  
            justStartedPlaying = true;  
  
            /*  
             * Always add event listeners and error handling  
             * within the onCreate() function.  
             * Include this function in the embedded parameters  
             * in the Player.create() call.  
             */  
            function onCreate(player) {  
  
                /*
```



```

        * Subscribe to all events:
        * this allows you to create logic
        * based on multiple events.
    */
    player.mb.subscribe("*", 'example', function(eventName) {
        if (eventName != OO.EVENTS.DOWNLOADING &&
            eventName != OO.EVENTS.PLAYHEAD_TIME_CHANGED)
            console.log(eventName);
    });
}

OO.ready(function() {
    window.pp = OO.Player.create('container', 'YOUR_ASSET_ID', {
        "playerBrandingId": "YOUR_PLAYER_ID",
        "PCODE": "YOUR_PCODE",
        skin: {
            config: "url_where_hosted/skin-plugin/
skin.json",
        },
        // Include the onCreate() function you defined
        above in these embedded parameters.
        onCreate : window.onCreate
    });
});

</script>

Messages: <input type="text" id="messagesTxt" size="108" value=" ">

</body>
</html>

```

Listening to Multiple Message Bus Events

You can use the message bus to listen to multiple events.

The following is a complete example that illustrates how to use the message bus to listen for multiple events.

To develop the head element and the logic for presenting and creating the Player, see [Basic Tutorial for Player V4](#) on page 53.

To learn about event handling, see [Event Model for Player V4](#) on page 212. For a list of all events, see the [Player JavaScript API Reference](#).

In this example, you will learn how to use the `onCreate()` function to handle events related to the playing of a video. In this advanced application, the user plays a video, and the event handlers determine when the video has finished playing. At that point, the event handler destroys the player and displays a message to the user.

The purpose of this web page is to monitor and respond to downloading, playhead time changes, and played events, so the design will include:

- A head element containing a `script` tag specifying the loading and initialization request.
- A body element containing:
 - UI presentation and layout for the Player container.
 - Event listeners that monitor and respond to downloading, playhead time changes, and played events.
 - A `script` tag that creates the Player.

Follow these steps to build your web page:



- [Step 1: Listening to Multiple Events](#) on page 217
- [Step 2: Destroying the Player](#) on page 217

See [Complete Example](#) on page 218 to view the code for the entire web page.

STEP 1: LISTENING TO MULTIPLE EVENTS

Sometimes it is useful to listen and respond to multiple events at once. One way to do this is to subscribe to all events. The [Event Model for Player V4](#) on page 212 provides you with a message bus, accessed via the created Player object's `mb` object. In this case we will use the message bus `subscribe()` function specifying a wildcard ("*") for the event name:

```
function onCreate(player) {
    /* Subscribe to all events:
     * this allows you to create logic based on multiple events. */
    player.mb.subscribe("*", 'example', function(eventName) {
        });
}
```

Now that we are listening for every possible event, we can include logic based on multiple events within the callback function via its `eventName` parameter. In this example we write all events, except for downloading and playhead time changed events (`OO.EVENTS.DOWNLOADING` and `OO.EVENTS.PLAYHEAD_TIME_CHANGED`), to the JavaScript console log:

```
function onCreate(player) {
    /*
     * Subscribe to all events:
     * this allows you to create logic
     * based on multiple events.
     */
    player.mb.subscribe("*", 'example', function(eventName) {
        /* Write all events, except for downloading and playhead time
        changed events,
         * to the JavaScript console: */
        if (eventName != OO.EVENTS.DOWNLOADING &&
            eventName != OO.EVENTS.PLAYHEAD_TIME_CHANGED)
            console.log(eventName);
    });
}
```

STEP 2: DESTROYING THE PLAYER

Once the video has finished playing, we would like to properly destroy the player and dispose of its resources. To do this we listen for the `OO.EVENTS.PLAYED` event:

```
/* Subscribe to the PLAYED event,
 * which only occurs when the video has finished playing: */
player.mb.subscribe(OO.EVENTS.PLAYED, 'example',
function(eventName) {
});
```



To destroy the player and dispose of its associated resources, call the Ooyala V3 Player object's `destroy()` method:

```
        /* Subscribe to the PLAYED event,
         * which only occurs when the video has finished playing: */
        player.mb.subscribe(OO.EVENTS.PLAYED , 'example',
    function(eventName) {

        /* Destroy the player and properly dispose of its resources:
        */
        player.destroy();

        /* Notify the user:
        alert('The player has been destroyed!');
    });
}
```

Note: In iOS you can only have one HTML5 player running at a time on a web page. You can call the Ooyala V3 Player object's `destroy()` method to ensure a player does not prevent the playback of another video in a different player on the page.

Here is the completed `onCreate()` function:

```
function onCreate(player) {

    /*
     * Subscribe to all events:
     * this allows you to create logic
     * based on multiple events.
    */
    player.mb.subscribe("*" , 'example', function(eventName) {

        /* Write all events, except for downloading and playhead time
        changed events,
         * to the JavaScript console: */
        if (eventName != OO.EVENTS.DOWNLOADING &&
            eventName != OO.EVENTS.PLAYHEAD_TIME_CHANGED)
            console.log(eventName);
    });

    /* Subscribe to the PLAYED event,
     * which only occurs when the video has finished playing: */
    player.mb.subscribe(OO.EVENTS.PLAYED , 'example',
    function(eventName) {

        /* Destroy the player and properly dispose of its resources:
        */
        player.destroy();

        /* Notify the user:
        alert('The player has been destroyed!');
    });
}
}
```

COMPLETE EXAMPLE

You now have a complete working example of a web page that responds to multiple events when the user plays a video. The event handlers determine when the video has finished playing, at which point the player is destroyed and a confirmation message is displayed to the user:

```
<!DOCTYPE html>
```



```

<html>
    <head>
        <link rel="stylesheet" href="url_where_hosted/skin-plugin/html5-
skin.min.css"/>
        <script language="javascript" src="url_where_hosted/core.min.js"></
script>
        <script type="text/javascript" src="url_where_hosted/video-plugin/
main_html5.min.js"></script>
        <script type="text/javascript" src="url_where_hosted/skin-plugin/
html5-skin.min.js"></script>
    </head>
    <body>
        <b>Destroy Player Example</b><br/><br/>
        <div id='container' style='width:640px;height:480px'></div>

        <script>
            function onCreate(player) {

                /* Subscribe to all events:
                 * this allows you to create logic based on multiple events. */
                player.mb.subscribe("*" , 'example', function(eventName) {

                    /* Write all events, except for downloading and playhead time
changed events,
                     * to the JavaScript console: */
                    if (eventName != OO.EVENTS.DOWNLOADING &&
                        eventName != OO.EVENTS.PLAYHEAD_TIME_CHANGED)
                        console.log(eventName);
                });

                /* Subscribe to the PLAYED event,
                 * which only occurs when the video has finished playing: */
                player.mb.subscribe(OO.EVENTS.PLAYED , 'example',
function(eventName) {

                    /* Destroy the player and properly dispose of its resources:
*/
                    player.destroy();

                    /* Notify the user: */
                    alert('The player has been destroyed!');
                });
            }

            OO.ready(function() {
                window.pp = OO.Player.create('container',
'YOUR_ASSET_ID', {
                    "playerBrandingId": "YOUR_PLAYER_ID",
                    "url_where_hosted/skin-plugin/skin.json",
                    "PCODE": "YOUR_PCODE",
                    "skin": {
                        "config": {}
                    },
                    "onCreate": window.onCreate
                });
            });
        </script>
    </body>
</html>

```



Writing to the Message Bus

Use the `publish()` function to write to the message bus, setting parameters as needed.

To write to the message bus, use the `publish()` function. For example, you can set the player to fullscreen mode by publishing the `WILL_CHANGE_FULLSCREEN` event with an event parameter of `true`:

```
this.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN, true);
```

To exit from fullscreen, set the `WILL_CHANGE_FULLSCREEN` event parameter to `false`:

```
this.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN, false);
```

The following is a complete example in which the `publish()` function is used to control fullscreen mode for the player. To develop the head element and the logic for presenting and creating the Player, see [Basic Tutorial for Player V4](#) on page 53.

To learn about event handling, see [Event Model for Player V4](#) on page 212. For a list of all events, see the [Player JavaScript API Reference](#).

ABOUT THIS EXAMPLE

In this example, you will learn how to use the `onCreate()` function to handle events related to the use of fullscreen mode. In this advanced application, the user clicks a button to switch to fullscreen mode, which is automatically disabled after 5 seconds via an event handler that monitors the playhead time (the video duration).

The purpose of this web page is to provide the user with an option to switch to fullscreen mode, so the design will include:

- A head element containing a `script` tag specifying the loading and initialization request.
- A body element containing:
 - UI presentation and layout for the Player container.
 - An event listener that monitors the playhead time.
 - A `script` tag that creates the Player.
 - An HTML `button` for switching to fullscreen mode, with an associated button click event handler.

Complete the following steps to build your web page:

- [Step 1: Capture a Button Click](#) on page 220
- [Step 2: Monitor the Play Duration](#) on page 222

See [Complete Example](#) on page 223 to view the sample code for the entire web page.

STEP 1: CAPTURE A BUTTON CLICK

We will create the button used for switching to fullscreen mode, along with its associated event handler.

Let's create a button with the label **Swap to Fullscreen**. We will assign an `id` attribute called `swapbutton` so that we can associate the button with an event handler:

```
<button id="swapbutton">Swap to Fullscreen</button>
```



We must also define a button click event handler, and refer to the button using its `id` attribute of `swapbutton`. We will take advantage of the [jQuery](#) library included with the Ooyala V4 Player, referencing the button using the jQuery functionality included in the `OO` namespace:

```
OO.$("#swapbutton").click(function(){
    // logic to handle the fullscreen button
});
```

Now we can add the logic to handle the fullscreen event after the button is clicked. The [Event Model for Player V4](#) on page 212 provides you with a message bus, accessed via the created Player object's `mb` object. In this case, we will use the message bus `publish()` function to notify the Player that it is changing to fullscreen mode:

```
player.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN,true);
```

In this case, we pass the `OO.EVENTS.WILL_CHANGE_FULLSCREEN` event with a value of `true`, indicating that the Player is to switch to fullscreen mode. For a list of all events, see the [Player JavaScript API Reference](#). Here is the web page design we have so far:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="url_where_hosted/skin-plugin/html5-skin.min.css"/>
    <script language="javascript" src="url_where_hosted/core.min.js"></script>
    <script type="text/javascript" src="url_where_hosted/video-plugin/main_html5.min.js"></script>
    <script type="text/javascript" src="url_where_hosted/skin-plugin/html5-skin.min.js"></script>
  </head>
  <body>
    <b>Full Screen Example</b><br/><br/>
    <div id='container' style='width:640px;height:480px'></div>
    <script>

      OO.ready(function() {
        window.pp = OO.Player.create('container', 'YOUR_ASSET_ID', {
          "playerBrandingId": "YOUR_PLAYER_ID",
          "PCODE": "YOUR_PCODE",
          "skin": {
            config: "url_where_hosted/skin-plugin/skin.json"
          },
          // Include the onCreate() function you defined above in these
          // embedded parameters.
          "onCreate" : window.onCreate
        });
      });
      // This function defines a button that enables fullscreen mode when
      // clicked.
      // The id, swapbutton, matches the id attribute of the <button>
      // control defined below.
      OO.$("#swapbutton").click(function()
      {
        // Use the message bus API to publish the WILL_CHANGE_FULLSCREEN
        // event.
        // This time the Boolean argument (true) indicates
        // that the full screen setting is to be enabled.
        player.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN,true);
      })
    </script>
  </body>
</html>
```



```

</script>
<br />
<!-- Place a button below the player container.
Its id attribute (swapbutton) is referenced above in the click event. --
&gt;
&lt;button id="swapbutton"&gt;Swap to Fullscreen&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

STEP 2: MONITOR THE PLAY DURATION

You can manage all events, errors, and customization related to the Player object within the `onCreate()` function, which is included in the body as shown here:

```

<body>
<b>Full Screen Example</b><br /><br />
<div id='container' style='width:640px;height:480px'></div>
<script>
    // Always add event listeners and error handling within the onCreate()
function.
    // Include this function in the embedded parameters in the
Player.create() call.
    function onCreate(player) {
        }

```

In this case, we are monitoring the playback duration. We would like to disable fullscreen mode once the playhead reaches 5 seconds, so we will need to subscribe to the `OO.EVENTS.PLAYHEAD_TIME_CHANGED` event. This event calls the handler with several arguments. `argument[1]` is the duration in seconds, so we will test whether `argument[1]` exceeds 5 seconds before publishing the `OO.EVENTS.WILL_CHANGE_FULLSCREEN` event with a value of `false` (to disable fullscreen mode):

```

<script>
    // Always add event listeners and error handling within the onCreate()
function.
    // Include this function in the embedded parameters in the
Player.create() call.
    function onCreate(player) {
        // Listen for the playhead time change event,
        // and disable fullscreen mode after 5 seconds of play:
        player.mb.subscribe(OO.EVENTS.PLAYHEAD_TIME_CHANGED, 'test',
function(event) {
        // The PLAYHEAD_TIME_CHANGED event calls the handler with
several arguments.
        // argument[1] has the duration.
        if(arguments[1] > 5)
        {
            // Use the message bus API to publish the
WILL_CHANGE_FULLSCREEN event.
            // The Boolean argument (false) accompanies the event,
            // indicating the full screen setting is to be disabled.
            player.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN,false); }
        });
    }

```



Now that we have the `onCreate()` function, we must include it in the embedded parameters for the `OO.Player.create()` method call:

```
        window.pp = OO.Player.create('container', 'YOUR_ASSET_ID', {
            "playerBrandingId": "YOUR_PLAYER_ID",
            "PCODE": "YOUR_PCODE",
            "skin": {
                config: "url_where_hosted/skin-plugin/skin.json"
            },
            // Include the onCreate() function you defined above in these
            // embedded parameters.
            "onCreate" : window.onCreate
        });
    
```

COMPLETE EXAMPLE

Here is a complete working example of a web page that lets the user click a button to switch to fullscreen mode, which is automatically disabled after 5 seconds via an event handler that monitors the playhead time.

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="url_where_hosted/skin-plugin/html5-
skin.min.css"/>
        <script language="javascript" src="url_where_hosted/core.min.js"></
script>
        <script type="text/javascript" src="url_where_hosted/video-plugin/
main_html5.min.js"></script>
        <script type="text/javascript" src="url_where_hosted/skin-plugin/html5-
skin.min.js"></script>
    </head>
    <body>
        <b>Full Screen Example</b><br/><br/>
        <div id='container' style='width:640px;height:480px'></div>
        <script>
            // Always add event listeners and error handling within the onCreate()
            // function.
            // Include this function in the embedded parameters in the
            Player.create() call.
            function onCreate(player) {
                // Listen for the playhead time change event,
                // and disable fullscreen mode after 5 seconds of play:
                player.mb.subscribe(OO.EVENTS.PLAYHEAD_TIME_CHANGED, 'test',
function(event) {
                // The PLAYHEAD_TIME_CHANGED event calls the handler with
                // several arguments.
                // argument[1] has the duration.
                if(arguments[1] > 5)
                {
                    // Use the message bus API to publish the
                    WILL_CHANGE_FULLSCREEN event.
                    // The Boolean argument (false) accompanies the event,
                    // indicating the full screen setting is to be disabled.
                    player.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN,false); }
                }
            );
        }
        OO.ready(function() {
            window.pp = OO.Player.create('container', 'YOUR_ASSET_ID', {
```



```

        "playerBrandingId": "YOUR_PLAYER_ID",
        "PCODE": "YOUR_PCODE",
        "skin": {
            config: "url_where_hosted/skin-plugin/skin.json"
        },
        // Include the onCreate() function you defined above in these
        // embedded parameters.
        "onCreate" : window.onCreate
    });
});
// This function defines a button that enables fullscreen mode when
// clicked.
// The id, swapbutton, matches the id attribute of the <button>
// control defined below.
OO.$("#swapbutton").click(function()
{
    // Use the message bus API to publish the WILL_CHANGE_FULLSCREEN
    // event.
    // This time the boolean argument (true) indicates
    // that the full screen setting is to be enabled.
    player.mb.publish(OO.EVENTS.WILL_CHANGE_FULLSCREEN,true);
}
);
</script>
<br />
<!-- Place a button below the player container.
Its id attribute (swapbutton) is referenced above in the click event. -->
<button id="swapbutton">Swap to Fullscreen</button>
</body>
</html>

```

Information with Listeners and Method Calls

Use event listeners and Player V4 API methods to retrieve the asset's information you want to work with.

The simplistic example here uses getter methods to retrieve the asset's basic information (title, description, and duration) and bitrate- and buffer-related details. There are many other events and methods available.

In addition, the basic structure of an event listener is shown here, using `mb.subscribe`.

```

.
.
.

// Buffer listener
// Need to subscribe to an event if you want updates for the length of the
// buffer.
// Ideally you'd listen for the BUFFERING event.
window.bufferLength = -100;
player.subscribe('playheadTimeChanged', 'myPage', function(eventName) {
    var newBufferLength = player.getBufferLength();
    if (bufferLength === newBufferLength) { return; }
    window.bufferElement.innerHTML += "Buffer length is " +
player.getBufferLength() + "<br/>"
    window.bufferLength = newBufferLength;
});

// Bitrate listener
// You *must* listen to bitrateInfoAvailable in order to request it.
player.subscribe('bitrateInfoAvailable', 'myPage', function(eventName) {
    var rates = player.getBitratesAvailable();
    if (rates.length > 0) {

```



```

        for (var i=0; i < rates.length; i++) {
            window.bitrateElement.innerHTML += "Rate: " + rates[i] + "<br/>"
        }
    });

// Metadata
// Content information is available after contentTreeFetched, but it is
best to wait until
// playbackReady for duration.
player.subscribe('playbackReady', 'myPage', function(eventName) {
    window.metadataElement.innerHTML += "Title is: " + player.getTitle() +
"<br/>";
    window.metadataElement.innerHTML += "Description is: " +
player.getDescription() + "<br/>";
    window.metadataElement.innerHTML += "Duration is: " +
player.getDuration() + "<br/>";
});
}

window.bufferElement = getElement('buffer');
window.bitrateElement = getElement('bitrate');
window.metadataElement = getElement('metadata');

.
.
.
```

PROGRAMMING BITRATES AND BUFFERING FOR PLAYER V4 USING JAVASCRIPT

Ooyala provides bitrate and buffering functions and events. In addition, Ooyala uses a number of heuristics to make playback as smooth as possible.

Bitrates and buffer control on HDS, HLS, and DASH are supported in the V4 Player when using the `bit_wrapper.min.js` video plugin . See the example code in [Information with Listeners and Method Calls](#) on page 224.

Bitrate controls on the player UI and bitrate APIs are supported for all content types (Ooyala VOD, remote VOD, remote live, remote live DV).

The following table shows the JavaScript methods for bitrates and buffering. For detailed API docs, see http://apidocs.ooyala.com/player_v4_js/OO.Player.html.

Note: The following APIs do not apply to the mobile SDKs.

Table 1: JavaScript Methods for Bitrates

get Methods	set Methods
<ul style="list-style-type: none"> • <code>getBitratesAvailable()</code> • <code>getCurrentBitrate()</code> 	<ul style="list-style-type: none"> • <code>setTargetBitrate(<i>id</i>)</code>

JAVASCRIPT CONSOLE LOG FOR SETTARGETBITRATE

Calls to the bitrate control API `setTargetBitrate` supply additional information to the browser's built-in JavaScript console log. You can use this log for end user video configuration.



DETERMINING AVAILABLE BITRATES

The example below will show determining what bitrates are available and then will show what happens when you attempt to set some incorrect and correct values. This example assumes the player ID is `pp`. You will often see `undefined` in the console logs, which is normal for the bitrate API.

```
pp.getBitratesAvailable()
[Object > bitrate: 470000 height: 180 id: "0" width: 320_proto_: Object,
Object > bitrate: 788000 height: 360 id: "1" width: 640_proto_: Object,
Object > bitrate: 1157000 height: 360 id: "2" width: 640_proto_: Object,
Object > bitrate: 1760000height: 720id: "3"width: 1280_proto_: Object,
Object > bitrate: 2512000height: 720id: "4"width: 1280_proto_: Object,
Object > bitrate: 3986000height: 1080id: "5"width: 1920_proto_: Object,
Object > bitrate: 4657000height: 1080id: "6"width: 1920_proto_: Object,
Object > bitrate: 0height: 0id: "auto"width: 0_proto_: Object]
```

To set the target bitrate, use one of the `id` values returned from `getBitratesAvailable()`.

MANUALLY SETTING A BITRATE

You can manually set a bitrate after the `PLAYING` event occurs with `setTargetBitrate()`.

Note: Manually setting the bitrate will override ABR.

1. Listen to the `BITRATE_INFO_AVAILABLE` event.
2. Get the desired bitrate "id" (string).
3. Listen to the `PLAYING` event.
4. Once the `PLAYING` event is triggered, call `setTargetBitrate("id")`.

For more information on how to use `setTargetBitrate`, see http://apidocs.ooyala.com/player_v4_js/OO.Player.html.

ABOUT ABR

In addition to the bitrate and buffer functions and events, Ooyala uses a number of heuristics to make playback as smooth as possible on every device, such as bandwidth estimation, content size, screen size and so forth. ABR playback (HLS, specifically) is preferred whenever available on the underlying devices.

ABR is a technology that allows the Ooyala Player to adjust/change the bitrate of the stream delivered based on the bandwidth available to the viewer as measured over a period of time. The purpose of ABR is to compensate for drops or increases in bandwidth by lowering or upgrading the stream quality. The ABR changes to the stream quality do not happen immediately. This is not an instantaneous process. To make a bitrate change to the video stream, the player:

- Detects the change in bandwidth.
- Waits to confirm it is permanent vs. a momentary fluctuation.
- Sends a request to the Akamai Flash Media Server to shift to stream at the new bitrate.
- Waits for Akamai to deliver the new bitrate stream.
- Starts delivering the new bitrate to the viewer.

Note: ABR and bitrate selection are currently mutually exclusive. You can manually set a bitrate, but this will override ABR.

ABR takes effect only when the player has detected there is sufficient bandwidth available to support a higher bitrate and higher quality stream. To upgrade the player makes multiple checks over a period of time. If all of the checks show the viewer has the required bandwidth to support a higher stream, then it will upgrade. If any of the checks fail to show the required bandwidth, then the process is repeated. The



upgrade also depends on the buffer length meeting a minimal threshold, and being stable or increasing. Dropping the bitrate of the stream being delivered is done to favor smooth and uninterrupted playback. Therefore, if the player sees the buffer size dropping, then it will start the process of downgrading the bitrate in order to have a lower bitrate stream available when the current buffer is exhausted.

Note: The current Ooyala player does not support `activePanelChanged`, `ratingsApiReady`, and `relatedMediaReady` events. Although the `apiReady`, `playerCreated`, `loadComplete`, and `playerEmbedded` events are not provided in the player, you can use the `PLAYBACK_READY` Event for equivalent functionality.

HANDLING ERRORS FOR PLAYER V4 USING JAVASCRIPT

Error Handling functions and events enable you to get information about Player errors.

The Ooyala Player V4 error functions and events are described in the following tables.

ERROR FUNCTIONS

The following table describes the supported error handling functions.

Function	Description
<code>getErrorCode</code>	Displays the video error code
<code>getState</code>	Retrieves the current player state. Possible states are defined in <code>OO.STATE</code> as follows <ul style="list-style-type: none">• <code>LOADING</code>• <code>READY</code>• <code>PLAYING</code>• <code>PAUSED</code>• <code>BUFFERING</code>• <code>ERROR</code>• <code>DESTROYED</code>

ERROR MESSAGES

The following namespaces, described in the [Player JavaScript API Reference](#), contain the error messages that can be handled through the V4 JavaScript Message bus. Use message bus events to handle errors by subscribing to or intercepting the `OO.EVENTS.ERROR` event:

Error Namespace	Description
<code>OO.ERROR</code>	Represents the Ooyala V4 Player Errors. This is the parent namespace for all errors, and includes errors for channel content.
<code>OO.ERROR.API</code>	Represents errors related to content, metadata, and server connection.
<code>OO.ERROR.API.SAS</code>	Represents all authorization-related errors.
<code>OO.ERROR.PLAYBACK</code>	Represents all errors related to playback.
<code>OO.ERROR.VC</code>	Represents errors related to video element creation and encoding.



ERROR EVENTS

The following table describes the error handling events published by the message bus.

Error Event	Description
PLAYER_CREATED	The very first message that is published after player is created, used to notify all plugins to initialize themselves.
ERROR	Published when error has occurred, second parameter has error details.
DESTROY	Published when player is being destroyed.

WORKING WITH THE ERROR EVENT

The OO.EVENTS.ERROR event is published to the message bus. The mb.subscribe triggers the function you specify (the callback function) when an error happens, and the error continues to propagate to other subscribers. Use mb.subscribe and handle the error as desired according to any existing error-handling functions you might already have in place (such as printing to the console, stopping playback, displaying an error screen, etc). In the following example illustrating the use of the Message Bus subscribe() method, the callback function YourErrorHandlerFunctionNameHere is invoked if the OO.EVENTS.ERROR event is published to the message bus:

```
<script>
    function onCreate(player) {

        player.mb.subscribe(
            OO.EVENTS.ERROR,
            'test',
            function(event, payload) {
                YourErrorHandlerFunctionNameHere();
            }
        );
    }

    OO.ready(function() {
        window.pp = OO.Player.create('container', 'YOUR_ASSET_ID', {
            "playerBrandingId": "YOUR_PLAYER_ID",
            "PCODE": "YOUR_PCODE",
            skin: {
                config: "url_where_hosted/skin-plugin/skin.json",
            },
            // Include the onCreate() function you defined above in these
            // embedded parameters.
            onCreate: window.onCreate
        });
    });
</script>
```

DEBUGGING PLAYER V4

General tips for how to debug Player V4 are described below.



- One of things you can do to debug Player V4 is watch debug messages and message bus events in the JS debug console. For that you can set the debug player parameter to `true` and watch events from the message bus. See the code sample below:

```
OO.ready(function() {
    window.pp = OO.Player.create('container', 'YOUR_ASSET_ID',
{
    playerBrandingId: "pbid",
    pcode: "PCODE",
    layout: "chromeless",
    debug: true,
    // other params
    onCreate: function(player) {
        player.mb.subscribe('*', 'test',
function(event) {
            if (!event.match(/playheadTime/)) {
                console.log("VTC: " + event +
JSON.stringify(arguments, null, '\t'));
            }
        });
    }
});
```

- To make debugging easier, point to the unminified versions of the player and modules. For example, Minified: http://player.ooyala.com/static/v4/stable/4.1.4/ad-plugin/google_ima.min.js versus Unminified: http://player.ooyala.com/static/v4/stable/4.1.4/ad-plugin/google_ima.js.



EMBEDDING PLAYER V4 IN ACCELERATED MOBILE PAGES (AMP)

You can optimize playback performance on mobile devices by embedding Ooyala Player V4 on a web page using [Accelerated Mobile Pages \(AMP\)](#), an open-source JavaScript framework for mobile web browsers. AMP renders web pages faster by managing how resources are loaded and providing other performance-enhancing features, such as:

- Download most important resources first, and fetch lazy-loaded resources ahead of time
- Optimize font downloads and use of styles
- Pre-calculate layout of all page elements before loading external resources, avoiding time-consuming page layout calculations at load time
- Pause media when it is not in focus
- Keep external libraries to a minimum

For more information, see [How AMP Works](#) in the AMP documentation.

SUPPORTED PLAYER CAPABILITIES

The following Player V4 capabilities are supported in this release:

- Configure the embedCode, pcode, playerID, and skin.json file
- Playback experience targeted for mobile devices
- Requires the main_html5 video plugin

UNSUPPORTED PLAYER CAPABILITIES

The following Player V4 capabilities are currently not supported:

- Page-level parameters
- Ads plugins
- Other video plugins (only main_html5 is supported)
- Discovery
- Playlist
- Ooyala Player Token
- Autoplay

SUPPORTED LAYOUTS

AMP provides the `layout` attribute, a common attribute that can be used in any element on an AMP HTML page. This optional attribute expands the capabilities of CSS so you can more easily create a responsive page design. The AMP Ooyala Player supports the following layout types:

- `fill`
- `fixed`
- `flex-item`
- `responsive`

For more information, see [Layout & media queries](#) in the AMP documentation.



INTEGRATION STEPS

To add Ooyala Player V4 to an AMP-optimized HTML page, use a special AMP HTML tag to load the AMP version of Ooyala Player that is hosted on the Google CDN.

1. Create an AMP HTML page, as described at https://www.ampproject.org/docs/get_started/create.
2. In the <head> section, insert the following line to load the Ooyala Player V4 AMP plugin:

```
<script async custom-element="amp-ooyala-player" data-amp-report-test="amp-ooyala-player" src="https://cdn.ampproject.org/v0/amp-ooyala-player-0.1.js"></script>
```

3. In the <body> section, embed the player, as shown in the following example:

```
<amp-ooyala-player  
height=200  
width=400  
data-embedcode="Vxc2k0MDE6Y_C7J5podo3UDx1FxGaZrQ"  
data-pcode="5zb2wx01ZcNCe_HVT3a6cawW298X"  
data-playerid="26e2e3c1049c4e70ae08a242638b5c40"  
data-playerversion="v4">  
</amp-ooyala-player>
```

The amp-ooyala-player element handles loading the Ooyala player in an iframe (amp_iframe.html). For a list of parameters for the amp-ooyala-player element, see <https://www.ampproject.org/docs/reference/components/amp-ooyala-player>. You can also use [Common Attributes](#).

Note:

- For Player V4, you must specify the player version (data-playerversion="v4"). If omitted, the default is Player V3, which is deprecated.
- The URL to a skin.json config file (data-config) is optional for Player V4. It does not apply to Player V3.

CODE EXAMPLE

```
<!doctype html>  
<html amp lang="en">  
  <head>  
    <meta charset="utf-8">  
    <link rel="canonical" href="amps.html" />  
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">  
    <style>body {opacity: 0}</style><noscript><style>body {opacity: 1}</style></noscript>  
    <script async custom-element="amp-ooyala-player" data-amp-report-test="amp-ooyala-player" src="https://cdn.ampproject.org/v0/amp-ooyala-player-0.1.js"></script>  
    <script async src="https://cdn.ampproject.org/v0.js"></script>  
  </head>  
  <body>  
    <amp-ooyala-player  
      height=200  
      width=400  
      layout="responsive"  
      data-embedcode="Vxc2k0MDE6Y_C7J5podo3UDx1FxGaZrQ"  
      data-pcode="5zb2wx01ZcNCe_HVT3a6cawW298X"  
      data-playerid="26e2e3c1049c4e70ae08a242638b5c40"  
      data-playerversion="v4">  
    </amp-ooyala-player>
```



```
</body>  
</html>
```



USING PLAYER V4 IN FACEBOOK INSTANT ARTICLES

Facebook Instant Articles (FBA) is a mobile optimization platform designed for better mobile experiences. Facebook Instant Articles are fast-loading versions of web articles where users stay in Facebook to view content rather than follow a link to another site. Publishers can incorporate the Ooyala Player into instant articles to deliver an optimal playback experience for consumers. For an introduction, see [Build your mobile video site with Facebook Instant Articles](#).

PREREQUISITES

To use Player V4 with Facebook Instant Articles, you need:

- a Facebook account with a page you administer. To begin, go to Facebook's [Getting Started](#) instructions.
- access to a site where you can write, design and publish your own blog. See Facebook's [Instant Articles Partners](#) for partner resources.

SUPPORTED PLAYER CAPABILITIES

The following Player capabilities are supported in this release:

- integration of Player V4 embedded within an HTML iFrame
- monetization through pre-roll and mid-roll ads (where permitted by Facebook)
- event reporting to Ooyala IQ Analytics
- multiple embedded players on a page

INTEGRATION STEPS

1. Embed the player on a web page using an iFrame, as described in [Embedding Player V4 in an HTML iframe](#) on page 95. For example:

```
<iframe width="480" height="320" src="url_where_hosted/iframe.html?
  ec=YOUR_ASSET_ID
  &pbid=YOUR_PLAYER_ID
  &pcode=YOUR_PCODE"
  frameborder="0" allowfullscreen>
</iframe>
```

Note: The &options tag is not yet supported.

2. Enclose each embedded iFrame with a Facebook-defined figure class ("op-interactive"). Example:

```
<figure class="op-interactive">
  <iframe width="480" height="320" src="url_where_hosted/iframe.html?
    ec=YOUR_ASSET_ID
    &pbid=YOUR_PLAYER_ID
    &pcode=YOUR_PCODE"
    frameborder="0" allowfullscreen>
  </iframe>
<figure>
```

Note: You can embed multiple players on the same page.



3. In Facebook, you simply register the URL of the page containing one or more embedded Ooyala player videos. Follow Facebook's [Getting Started](#) instructions.
 - a. Go to the page you are managing.
 - b. Choose Publishing Tools -> Configuration.
 - c. Follow the instructions for Step 1: Set Up Instant Articles.
 - d. Under Tools, for each page where you have an embedded player configured per the above steps, add its URL to Your Registered URLs.
4. Once approved, the videos will become available to insert into Facebook. Follow Facebook's instructions for inserting content into your instant articles.
5. Facebook Instant Articles will then publish the article to your Facebook page. When users click on the link to an article, they are taken to the instant article (on a mobile device, they will see it within Facebook's in-app browser).



OOYALA PLAYER V4 RELEASE NOTES

OOYALA-HOSTED PLAYER PATHS

Important: Ooyala generally recommends that customers use the /production hosted player resource path. Customers looking to have the latest capabilities urgently should use the /production/latest hosted resource path. Otherwise, customers on the /production path will get the versions on the /production/latest paths after at least two weeks of operating on the /production/latest path. Also, if you are hosting using the Ooyala Standard embed method (see [Configuring Player Embed Settings in Backlot](#)), you can toggle the player path directly in Backlot, requiring no change to your website to move between these two versions. For more information on Ooyala Player V4 hosted resource options, see [Ooyala-hosted Player V4 Resources](#) on page 77.

Current Version	Ooyala Path	Last Updated
Player V4 Web v4.24.3	//player.ooyala.com/static/v4/production/latest/	2018-04-18
Player V4 Web v4.22.11	//player.ooyala.com/static/v4/production/	2018-04-04

OOYALA PLAYER V4.24.3 RELEASED (2018-04-18)

This is a maintenance release only. No new features were introduced in this release.

Important:

Chrome is blocking unmuted autoplay. [Please read more about the changes coming in the Chrome browser](#) and how you might be affected.

Fixed in this Release

- Fixed an issue related to blocking of autoplay on some devices with the Chrome browser on Android.

Known Issues in this Release

- **For Multi-Audio support**, the following are known issues in this release related to multiple audio tracks:
 - If a user first starts a player, and then changes the audio track and subsequently refreshes the page, the user might find that the initial audio track plays initially when the page is refreshed.
 - Going full screen with the audio selection menu open might disable the closing of the audio selection menu unless returning to full-screen.
 - High-Availability failover is not fully supported in combination with multi-audio type streams in this version.
 - Selecting different audio tracks on Chrome with Android is not supported for HLS-type streams with this release.
 - Selecting the multi-audio menu just before a preroll ad starts might result in the audio menu being displayed throughout the ad.
 - Full localization of the audio selection menu is not fully supported in this release.
 - Language selection closes the CC window and the video is paused.
- When using the Pulse plugin for ads, the use of autoplay is currently not supported.
- When transitioning between a flat video and a VR 360 video in Chrome on Android devices, the videos will pause and require the user to manually start the next video.
- For VR 360-type videos, the click-to-drag option for navigating videos is disabled.



- For DASH-type streams on Android devices, if closed captions are enabled, they will not be retained as enabled on a subsequent video (the end user will need to re-enable closed captions again).
- High Availability switchover is not working as expected. For High Availability, when using Safari, the browser must load the main_htm5.js plugin first (ahead of bit_wrapper.js) in order to work properly.
- For Freewheel ads running with muted autoplay on Android devices, if the user unmutes once the main video starts playing, mid-roll and post-roll ads will still play in a muted state until unmuted by the user.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the *iosPlayMode* page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release, you must also pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.24.3/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/iframe.html
Localization Files	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/en.json • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/es.json • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/jp.json • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/zh.json
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.24.3/skin-plugin/assets/images/ooyala.png



Resource	Name of File to Host
(*= included with Standard embed code)	
Video Recommendation (Discovery) plugin *	//player.ooyala.com/static/v4/stable/4.24.3/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.ooyala.com/static/v4/stable/4.24.3/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.ooyala.com/static/v4/stable/4.24.3/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin (deprecated)*	//player.ooyala.com/static/v4/stable/4.24.3/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.24.3/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.ooyala.com/static/v4/stable/4.24.3/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.ooyala.com/static/v4/stable/4.24.3/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.ooyala.com/static/v4/stable/4.24.3/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/conviva-core-sdk.min.js



Resource	Name of File to Host
(*= included with Standard embed code)	
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.24.3/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.24.3/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.24.2 RELEASED (2018-04-16)

This Ooyala Player Web release introduces new features and fixes previous issues.

Important:

Chrome is blocking unmuted autoplay. [Please read more about the changes coming in the Chrome browser](#) and how you might be affected.

New in this Release

- **Support for Muted Autoplay with Chrome v66**
 - With Chrome v66 (planned for release on April 17, 2018), Chrome will introduce blocking of video playback by default if the video is not muted. This version Ooyala Web Player or the previous release 4.23.6 is required to enable the autoplay of video on Chrome 66+. However, changes were in this version where if the browser autoplay blocking checks during player initialization timeout, the player will not attempt to autoplay. In this version, the player will instead attempt to autoplay content in a muted state. For more on how the autoplay operates between player version, please see [Overview of Autoplay with Ooyala Player](#).
- **Support for multiple audio tracks**
 - This release begins the Ooyala Player support for presenting and selecting different audio tracks. More details about Ooyala's support for multiple audio tracks will be published in the near future.

Fixed in this Release

- No known issues were fixed in this release.

Known Issues in this Release

- For Multi-Audio support, the following are known issues in this release related to multiple audio tracks:



- If a user first starts a player, and then changes the audio track and subsequently refreshes the page, the user might find that the initial audio track plays initially when the page is refreshed.
- Going full screen with the audio selection menu open might disable the closing of the audio selection menu unless returning to full-screen.
- High-Availability failover is not fully supported in combination with multi-audio type streams in this version.
- Selecting different audio tracks on Chrome with Android is not supported for HLS-type streams with this release.
- Selecting the multi-audio menu just before a preroll ad starts might result in the audio menu being displayed throughout the ad.
- Full localization of the audio selection menu is not fully supported in this release.
- Language selection closes the CC window and the video is paused.
- When using the Pulse plugin for ads, the use of autoplay is currently not supported.
- When transitioning between a flat video and a VR 360 video in Chrome on Android devices, the videos will pause and require the user to manually start the next video.
- For VR 360-type videos, the click-to-drag option for navigating videos is disabled.
- For DASH-type streams on Android devices, if closed captions are enabled, they will not be retained as enabled on a subsequent video (the end user will need to re-enable closed captions again).
- High Availability switchover is not working as expected. For High Availability, when using Safari, the browser must load the main_htm5.js plugin first (ahead of bit_wrapper.js) in order to work properly.
- For Freewheel ads running with muted autoplay on Android devices, if the user unmutes once the main video starts playing, mid-roll and post-roll ads will still play in a muted state until unmuted by the user.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the *iosPlayMode* page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release, you must also pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.24.2/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/iframe.html



Resource	Name of File to Host
(*= included with Standard embed code)	
Localization Files	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/en.json • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/es.json • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/jp.json • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/zh.json
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.24.2/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin *	//player.ooyala.com/static/v4/stable/4.24.2/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.ooyala.com/static/v4/stable/4.24.2/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.ooyala.com/static/v4/stable/4.24.2/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin (deprecated)*	//player.ooyala.com/static/v4/stable/4.24.2/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.24.2/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.ooyala.com/static/v4/stable/4.24.2/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.ooyala.com/static/v4/stable/4.24.2/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.ooyala.com/static/v4/stable/4.24.2/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/VideoHeartbeat.min.js



Resource	Name of File to Host
(*= included with Standard embed code)	
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.24.2/analytics-plugin/Nielsen.min.js
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.24.2/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.23.6 RELEASED (2018-04-03)

This Ooyala Player Web release introduces new features and fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- With Chrome v66 (planned for release on April 17, 2018), Chrome will introduce blocking of video playback by default if the video is not muted. Ooyala Web Player (4.23.6) is required to enable the autoplay of video on Chrome 66+. The video will automatically be muted if necessary to enable the autoplay to work.



- Added Korean as a default language available for the Ooyala Player.

Fixed in this Release

- Added some missing localization strings for Japanese, Chinese, and Spanish.
- Fixed an issue that caused ads to play muted on Safari, even though the user had unmuted the player.

Known Issues in this Release

- VPAID ad creatives need to made aware of changes to autoplay support in Chrome and Safari be formatted for autoplay support, otherwise autoplay of VPAID ads may not work as expected.
- When using autoplay on Android with Freewheel, if you unmute playback during the content playback, midroll and postroll ads will still play muted.
- When using the Pulse plugin for ads, the use of autoplay is currently not supported.
- When transitioning between a flat video and a VR 360 video in Chrome on Android devices, the videos will pause and require the user to manually start the next video.
- For DASH-type streams on Android devices, if closed captions are enabled, they will not be retained as enabled on a subsequent video (the end user will need to re-enable closed captions again).
- High Availability switchover is not working as expected. For High Availability, when using Safari, the browser must load the `main_htm5.js` plugin first (ahead of `bit_wrapper.js`) in order to work properly.
- For Freewheel ads running with muted autoplay on Android devices, if the user unmutes once the main video starts playing, mid-roll and post-roll ads will still play in a muted state until unmuted by the user.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the `iosPlayMode` page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release (4.23.6), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.23.6/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.23.6/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.23.6/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.23.6/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.23.6/skin-plugin/iframe.html



Resource	Name of File to Host
(*= included with Standard embed code)	
Images and Fonts	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.23.6/skin-plugin/assets/images/oyala-watermark.png • //player.oyala.com/static/v4/stable/4.23.6/skin-plugin/assets/images/oyala.png
Video Recommendation (Discovery) plugin *	//player.oyala.com/static/v4/stable/4.23.6/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.oyala.com/static/v4/stable/4.23.6/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.oyala.com/static/v4/stable/4.23.6/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin *	//player.oyala.com/static/v4/stable/4.23.6/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.23.6/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.oyala.com/static/v4/stable/4.23.6/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.oyala.com/static/v4/stable/4.23.6/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.oyala.com/static/v4/stable/4.23.6/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.23.6/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.23.6/analytics-plugin/AppMeasurement.js • //player.oyala.com/static/v4/stable/4.23.6/analytics-plugin/VisitorAPI.js • //player.oyala.com/static/v4/stable/4.23.6/analytics-plugin/omniture.min.js



Resource	Name of File to Host
(*= included with Standard embed code)	
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.23.6/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.23.6/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.23.6/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.23.6/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.23.6/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.22.11 RELEASED (2018-03-13)

This Ooyala Player Web release introduces additional ways to mute and unmute the player. It also fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- This player release introduces the ability to programmatically mute and unmute audio playback using the Player V4 JavaScript API. For example, calling `window.pp.unmute()` in the page will unmute the video. For details, see `OO.Player.mute()` and `OO.Player.unmute()` in the [Player V4 JavaScript API documentation](#).
- This player release introduces a new page-level parameter (`muteFirstPlay`) to control audio when a video starts. Player behavior can be affected by the `autoplay` setting:
 - `"muteFirstPlay":true` causes the video to start muted.
 - `"muteFirstPlay":false` causes the video to start unmuted.

Note: If `"autoplay":true` and the browser requires muted autoplay, then the video will play muted even if `"muteFirstPlay":false`.
- `"muteFirstPlay":true` and `"autoplay":true` causes the video to autoplay muted.



- This player release adds a new page-level parameter (*bitrateOverride*) that you can use specify the initial bitrate (user bandwidth, in kbps) for rendering FreeWheel ads.

Fixed in this Release

- Fixed an issue in which closed captions were not displaying by default for DASH streams and the Bitmovin plugin.
- Fixed an issue in which the manual resolution/bitrate selection menu was always ordered from highest to lowest.
- Fixed an issue in which the Pause screen would sometimes blur even when there was no title or description. The Pause screen now blurs the background if text is layered on the Pause screen.
- Fixed an issue for Live assets on Android Chrome in which closed captions did not display after the user disabled closed captions and then enabled them again.
- The control bar will no longer flash when the user transitions to a new video via Discovery or playlists.
- Fixed an issue in which, for VAST ads on Android with Chrome, the main video would go into a paused state after the pre-roll ad played completely.
- Fixed an issue in which the image transparency setting in Backlot for a Branding Overlay image was ignored.
- Fixed an issue in which the `OO.Player.isAdPlaying()` method in the Player V4 JavaScript API would return an incorrect state (returned `false` when it should have returned `true`).
- Fixed an issue with autoplay on Safari 11 that requires muting to autoplay.
- For the Samsung Mobile Internet browser, we now block autoplay to work around a playback loading problem with the autoplay setting. The user can manually click Play to start video playback.
- Fixed an issue in which playback would sometimes stop working after seeking while the video was paused.
- Fixed an issue on Android Chrome for videos associated with Google IMA ads in which, when the user clicked the Discovery tray and selected a video, the Discovery screen would sometimes loop several times before the selected video would play.
- Fixed an issue with Google IMA ads running in Safari on iOS in which the Play button would appear during a mid-roll ad.
- Fixed an issue in Google Chrome Canary with muted autoplay: if the user unmuted once the video started playing, the volume bar would still display as muted.

Known Issues in this Release

- High Availability (Live) on Chrome browser with Android is not supported in this release.
- The seek back operation in the Live DVR window is not currently supported for Android Chrome.
- For Freewheel ads running with muted autoplay on Android devices, if the user unmutes once the main video starts playing, mid-roll and post-roll ads will still play in a muted state until unmuted by the user.
- If using the *initialTime* page-level parameter in combination with pre-roll ads, the ads may be skipped on Chrome with Android and Safari desktop.
- On some Android devices with Chrome, Freewheel overlay ads may only display in full screen mode.
- For VPAID-type midroll ads, if closed-captions are enabled, closed captions may display during the ads with Safari on iOS devices.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the *iosPlayMode* page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release (4.22.11), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.



Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.22.11/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.22.11/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin *	//player.ooyala.com/static/v4/stable/4.22.11/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.ooyala.com/static/v4/stable/4.22.11/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.ooyala.com/static/v4/stable/4.22.11/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin *	//player.ooyala.com/static/v4/stable/4.22.11/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.22.11/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.ooyala.com/static/v4/stable/4.22.11/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.ooyala.com/static/v4/stable/4.22.11/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.ooyala.com/static/v4/stable/4.22.11/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	



Resource	Name of File to Host
(*= included with Standard embed code)	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.22.11/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.22.11/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.21.12 RELEASED (2018-02-14)

This Ooyala Player Web release introduces player-side support for mid-stream checking of geographic restrictions and support for DVR in iOS Safari. It also fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release



- This release introduces player-side support for mid-stream checking of geographic restrictions in publishing rules. For details, see [Creating Publishing Rules](#) and the [Ooyala Backlot Release Notes](#).
- This release adds support for DVR with iOS Safari (requires the [main_html5.min.js](#) plugin).

Fixed in this Release

- Fixed various issues relating to muted autoplay on Safari desktop browsers and ads.
- Fixed issues related to autoplay occasionally failing on Android devices with Chrome, which would sometimes result in disabling playback.
- Fixed an issue that would cause the Discovery button to appear in the control bar when the Standard player embed was used (this was resolved for all player versions).
- Fixed an issue in which the SSAI plugin would cause issues with a jQuery variable.
- Fixed an issue in which the [initialBitrate](#) parameter was not working correctly.

Known Issues in this Release

- On some Android devices with the Chrome browser, when using muted autoplay, the autoplay event may periodically not fire properly. When this occurs, it may cause additional issues for playback, such as unintended muting of ads, disabling replay, and requiring a manual restart of playback. The autoplay problems do not manifest on all devices, and their occurrence is intermittent. Therefore, consider not using autoplay with Chrome on Android until changes with the ecosystem make them more reliable (Chrome combined with Google IMA/Freewheel) in an upcoming Ooyala Player release. Using autoplay on Chrome on Android with the Pulse plugin currently does not support autoplay.
- Failover with High Availability Live streams does not work properly if `bit_wrapper.js` is invoked first on Safari. Therefore, `main_html5.js` must be listed first at the page level in order for failover to work on Safari desktop browsers.
- When using VPAID 2.0 ads on Safari with Google IMA plugin on iPad devices, mid-roll ads may play with audio only.
- Seeking past the midroll ad allows skipping of the midroll ad when using the Freewheel plugin.
- On Android Chrome, closed captions are not displayed after disabling and enabling the Captions button in a LIVE asset.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the `iosPlayMode` page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release (4.21.12), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.21.12/core.min.js

Skin Resources

HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.21.12/skin-plugin/html5-skin.min.js
--------------	--



Resource	Name of File to Host
(*= included with Standard embed code)	
Skin CSS *	//player.oyala.com/static/v4/stable/4.21.12/skin-plugin/html5-skin.min.css
Skin Config File	//player.oyala.com/static/v4/stable/4.21.12/skin-plugin/skin.json
Skin iFrame	//player.oyala.com/static/v4/stable/4.21.12/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.21.12/skin-plugin/assets/images/oyala-watermark.png • //player.oyala.com/static/v4/stable/4.21.12/skin-plugin/assets/images/oyala.png
Video Recommendation (Discovery) plugin *	//player.oyala.com/static/v4/stable/4.21.12/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.oyala.com/static/v4/stable/4.21.12/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.oyala.com/static/v4/stable/4.21.12/video-plugin/main_html5.min.js
Ad Plugins	
Oyala Pulse Ad Plugin *	//player.oyala.com/static/v4/stable/4.21.12/ad-plugin/pulse.min.js
Oyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.21.12/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.oyala.com/static/v4/stable/4.21.12/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.oyala.com/static/v4/stable/4.21.12/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.oyala.com/static/v4/stable/4.21.12/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.21.12/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.21.12/analytics-plugin/AppMeasurement.js



Resource	Name of File to Host
(*= included with Standard embed code)	
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.21.12/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.21.12/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.20.8 RELEASED (2018-01-23)

This Ooyala Player release introduces the ability to hide the Playlist plugin (while still autoplaying a sequence of videos) and fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- This release adds the ability to hide the Playlist plugin but still have it drive the autoplay of a sequence of videos. By default, the Playlist plugin is displayed. To hide it, configure the playlistsPlugin page-level parameter by setting hideUi to true.

```
{
  "playlistsPlugin": {
    "hideUi": true
  }
}
```



```
}
```

See [Page-level Parameters for Player V4](#) on page 99 for details.

Fixed in this Release

- Fixed an issue that caused duplicate ad requests for pre-roll ads under certain conditions with the VAST plugin.

Known Issues this Release

- The *initialBitrate* parameter is not working in this release.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the *iosPlayMode* page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release (4.20.8), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.20.8/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.20.8/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin *	//player.ooyala.com/static/v4/stable/4.20.8/other-plugin/discovery_api.min.js
Video Plugins	



Resource	Name of File to Host
(*= included with Standard embed code)	
Bitmovin Video Plugin for DASH and HLS *	//player.oyala.com/static/v4/stable/4.20.8/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.oyala.com/static/v4/stable/4.20.8/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin *	//player.oyala.com/static/v4/stable/4.20.8/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.20.8/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.oyala.com/static/v4/stable/4.20.8/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.oyala.com/static/v4/stable/4.20.8/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.oyala.com/static/v4/stable/4.20.8/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/VideoHeartbeat.min.js //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/AppMeasurement.js //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/VisitorAPI.js //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/conviva-core-sdk.min.js //player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.oyala.com/static/v4/stable/4.20.8/analytics-plugin/Nielsen.min.js



Resource	Name of File to Host
(*= included with Standard embed code)	

YOUTORA Analytics Plugin //smartplugin.youbora.com/v5/javascript/oyyalav4/stable/sp.min.js

Other Feature Plugins

Playlists plugin //player.oyala.com/static/v4/stable/4.20.8/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.20.7 RELEASED (2017-12-20)

This Ooyala Player release introduces playback support for VR 360 videos.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- VR 360 Video Support: This player version introduces support for VR 360, 360-degree virtual reality video playback for video on demand (VOD). Not yet supported for Live playback. For more information on what is supported and how to use Ooyala's VR 360 content, see [VR 360 Videos](#)

Fixed in this Release

- No previously known issues were fixed in this release.

Known Issues in this Release

- Using Playlists with VR 360 videos is not supported with Safari desktop browser.
- FreeWheel (FW) ads are not currently supported in combination with VR 360 videos.
- Full screen icon doesn't lead to full screen mode on iOS with VR 360 videos in portrait mode, and is displayed in a pseudo full-screen (native banners are still displayed).
- With VR 360 videos on Safari Mobile on iOS 10 and lower devices, the video must be hosted on the same domain as the player.
- Overlays with the VAST ad plugin are not supported.
- Watermarks are not clickable.
- If the user sets Safari to Never Auto-Play manually (not the default setting), the video will not play.
- With muted autoplay, some VPAID ads may still play the ads unmuted, even with the player in a muted state. This is due to a browser issue.
- The [initialBitrate](#) parameter is not working in this release.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the [iosPlayMode](#) page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- If you are self-hosting skin.json, if you deploy this release (4.20.7), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links



For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.20.7/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.20.7/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin *	//player.ooyala.com/static/v4/stable/4.20.7/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.ooyala.com/static/v4/stable/4.20.7/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.ooyala.com/static/v4/stable/4.20.7/video-plugin/main_html5.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin *	//player.ooyala.com/static/v4/stable/4.20.7/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.20.7/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.ooyala.com/static/v4/stable/4.20.7/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.ooyala.com/static/v4/stable/4.20.7/ad-plugin/freewheel.min.js



Resource	Name of File to Host
(*= included with Standard embed code)	
VAST and VPAID Plugin *	//player.ooyala.com/static/v4/stable/4.20.7/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.20.7/analytics-plugin/Nielsen.min.js
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.20.7/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.20.5 RELEASED (2017-12-14)

This Ooyala Player release introduces new accessibility and ad features and also fixes previous issues.



Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- Support for Screen Readers to navigate the full Closed Caption menu.
- Ability to adjust the maximum number of redirects or wrappers for the Google IMA Ad plugin.
- Ability to enable the Ooyala Player control bar during ads (via the new `playerControlsOverAds` page-level parameter). Example: { "playerControlsOverAds":true}

Caution: This capability should be used *only* for when VPAID ads have no interactivity. If this option is used in combination with ads that include user interaction (navigation links within the ad), they may conflict with the player control and prevent user from navigating the video.

Fixed in this Release

- Updates and fixes for browsers that require muting of video to allow autoplay.
- Fixed an issue that prevented autoplay from working with Safari when the initial time was set.

Known Issues in this Release

- Overlays with the VAST ad plugin are not supported with this release.
- Watermarks are not clickable in this release.
- If the user sets Safari to 'Never Auto-Play' manually (not the default setting), the video will not play.
- With muted autoplay, some VPAID ads may still play the ads unmuted, even with the player in a muted state.
- The `initialBitrate` parameter is not working in this release.

Important Notes for this Release

- Autoplay on iPhone devices requires setting the `iosPlayMode` page-level parameter to 'inline'.
- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.20.5), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
(*= included with Standard embed code)	
Core Player *	//player.ooyala.com/static/v4/stable/4.20.5/core.min.js
Skin Resources	
HTML5 Skin *	//player.ooyala.com/static/v4/stable/4.20.5/skin-plugin/html5-skin.min.js
Skin CSS *	//player.ooyala.com/static/v4/stable/4.20.5/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.20.5/skin-plugin/skin.json



Resource	Name of File to Host
(*= included with Standard embed code)	
Skin iFrame	//player.oyala.com/static/v4/stable/4.20.5/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.20.5/skin-plugin/assets/images/oyala-watermark.png • //player.oyala.com/static/v4/stable/4.20.5/skin-plugin/assets/images/oyala.png
Video Recommendation (Discovery) plugin *	//player.oyala.com/static/v4/stable/4.20.5/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS *	//player.oyala.com/static/v4/stable/4.20.5/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4 *	//player.oyala.com/static/v4/stable/4.20.5/video-plugin/main_html5.min.js
Ad Plugins	
Oyala Pulse Ad Plugin *	//player.oyala.com/static/v4/stable/4.20.5/ad-plugin/pulse.min.js
Oyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.20.5/ad-plugin/ssai_pulse.min.js
Google IMA Plugin *	//player.oyala.com/static/v4/stable/4.20.5/ad-plugin/google_ima.min.js
FreeWheel Plugin *	//player.oyala.com/static/v4/stable/4.20.5/ad-plugin/freewheel.min.js
VAST and VPAID Plugin *	//player.oyala.com/static/v4/stable/4.20.5/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.20.5/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.20.5/analytics-plugin/AppMeasurement.js • //player.oyala.com/static/v4/stable/4.20.5/analytics-plugin/VisitorAPI.js



Resource	Name of File to Host
(*= included with Standard embed code)	
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.20.5/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.20.5/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.20.5/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.20.5/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.20.5/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.20.5/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.19.3 RELEASED (2017-11-29)

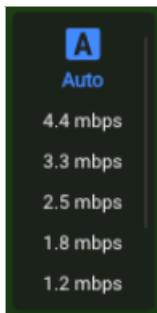
This Ooyala Player release introduces support for Ooyala's High Availability capability for Live streams, as well as different variations of the Quality Selection menu (Bitrate, Resolution, and Bitrate + Resolution). This release also fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- This release introduces support for Ooyala's High Availability capability for Live streams.
- This release adds support for different display variations of the Quality Selection menu (Bitrate, Resolution, and Bitrate + Resolution). The current default is Bitrate, so that what the viewer sees are available bitrates listed in kbps (or mbps if applicable).





You can change the default to show the Quality Selection menu options by:

- available stream resolutions. Example: 1080p
- available stream resolution and bitrate together. Example: 1080p (2500 kbps)

The `qualitySelection` configuration is under "controlBar" in `skin.json` (under `controlBar`). To enable this feature, specify the following:

```
"qualitySelection":{  
    "format": "resolution"  
}
```

where "format" is one of the following values:

- "bitrate" (default)
- "resolution"
- "resolution bitrate"

For details, see [Player V4 JSON Schema](#).

Fixed in this Release

- Updated the Mute icon overlay shown on videos where muting is required to autoplay.
- Fixed an issue in which the replay button sometimes remained on screen during a replay event on Safari.

Known Issues in this Release

- For Chrome browsers, playback does not always resume if the subchannel is disabled when the video is paused or seeked back.
- With Safari 11, setting an initial time for playback, combined with autoplay, will not start the video automatically.
- The `initialBitrate` parameter is not working in this release.

Important Notes for this Release

- In order to use the (muted) autoplay function with Safari Mobile, you must also set the video to play inline. If you do not set the player to play inline (i.e. it automatically goes full-screen), the video will not start automatically.
- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting `skin.json`, if you deploy this release (4.19.3), you also need to pull the most recent version of the `skin.json` file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.



Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.19.3/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.19.3/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.19.3/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.19.3/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.19.3/video-plugin/main_html5.min.js	included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.19.3/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.19.3/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.19.3/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.19.3/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.19.3/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/VideoHeartbeat.min.js 	not included



Resource	Name of File to Host	Included in Standard Embed Code?
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/omniture.min.js 	
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.19.3/analytics-plugin/Nielsen.min.js	not included
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.19.3/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.18.15 RELEASED (2017-11-14)

This Ooyala Player patch release fixes previous issues. No new features are introduced in this release.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Fixed in this Release

- **iPhone autoplay** - To use (muted) autoplay with iOS Safari (supported with iOS 10+), you must also use the "iosPlayMode": "inline" page-level parameter as well. If this is not specified, then the autoplay setting will be ignored.
- For some conditions, the up-next video would be blurred.
- For multiple video embeds on a single page, the player skin sometimes would not load for subsequent players.



- Removal of the client-side heartbeat authorization (`auth_heartbeat`).

Known Issues in this Release

- Ooyala Ads (My Ads) do not autoplay on mobile browsers (when autoplay is enabled).
- On Safari 11, if the user has changed their browser settings manually to "Always Auto-Play", mid-roll ads with Google IMA will mute the mid-roll ad even though the main video is unmuted.
- In Safari 11, if the "Never Autoplay" option is enabled:
 - For the first video of a playlist, if the viewer clicks the play button, the video plays in a muted state.
 - If a mid-roll Freewheel ad is set to autoplay, during playback, the ad keeps buffering.
- In Safari 11, if the "Allow All Auto-play" option is enabled, after playing a Pulse pre-roll ad, the main video plays but the sound is muted and no unmute button appears; the viewer must unmute by clicking the volume button.
- In Safari 11, if the viewer clicks the Replay button after viewing the video, the replay button still shows while the video is replaying.
- The `player initialTime` parameter is sometimes not respected by the Safari browser or by Chrome mobile on Android devices.
- The `initialBitrate` parameter is not working in this release.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.18.15), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.18.15/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.18.15/skin-plugin/assets/images/ooyala.png 	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Video Recommendation (Discovery) plugin	//player.oyala.com/static/v4/stable/4.18.15/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.oyala.com/static/v4/stable/4.18.15/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.oyala.com/static/v4/stable/4.18.15/video-plugin/main_html5.min.js	included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.oyala.com/static/v4/stable/4.18.15/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.18.15/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.oyala.com/static/v4/stable/4.18.15/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.oyala.com/static/v4/stable/4.18.15/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.oyala.com/static/v4/stable/4.18.15/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/AppMeasurement.js • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/VisitorAPI.js • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/conviva-core-sdk.min.js • //player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.18.15/analytics-plugin/googleAnalytics.min.js	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.18.15/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.18.15/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.18.13 RELEASED (2017-11-09)

This Ooyala Player release introduces new features, including muted autoplay, accessibility enhancements for video quality (ABR) selection, and new error messages for viewers who attempt to access a video ahead of its configured flight time.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- Added support for muted autoplay on mobile browsers (Safari and Chrome Mobile). To enable autoplay, in page-level settings, specify "autoplay":true.
 - In browsers where muted autoplay is not required (Chrome, Firefox, Edge, and Internet Explorer), playback will be unmuted.
 - In browsers where muted autoplay is required (Safari, iOS Safari, and Chrome Mobile), playback will be muted.
- Added error messaging for users who attempt to access a video ahead of the Flight Time that indicates when the video will be available.
- Added support for screen readers (accessibility) for the video quality (bitrate) selection menu.
- In this release, MPEG-DASH is by default the priority format over HLS if (a) the device supports it, and (b) your account has MPEG-DASH stream packaging as part of your profile. If you set the encoding priority at the page level, however, the page-level encoding will still have priority.

Fixed in this Release

- Fixed an issue with unmuted autoplay being blocked on Safari v11 browsers. With this version, if autoplay is set when rendering in Safari, the ads or content will start muted, along with a visual indicator to the viewer that the video is muted.
- Fixed an issue with inline playback for iOS 10+ when used in combination with Discovery and pre-roll / post-roll ads.
- Fixed an issue in which a closed caption API value of 'none' did not turn off closed captions.
- Fixed an issue in which closed captions were not rendering correctly on iOS 11.
- Fixed an issue with the IMA page-level parameter of t=100 (used for post-rolls), which sometimes did not trigger ads.
- Various cosmetic enhancements to make the player transitions smoother and more continuous, particularly during player startup and transitioning between videos.



Known Issues in this Release

- Ooyala Ads (My Ads) do not autoplay on mobile browsers (when autoplay is enabled).
- On Safari 11, if the user has changed their browser settings manually to "Always Auto-Play", mid-roll ads with Google IMA will mute the mid-roll ad even though the main video is unmuted.
- In Safari 11, if the "Never Autoplay" option is enabled:
 - For the first video of a playlist, if the viewer clicks the play button, the video plays in a muted state.
 - If a mid-roll Freewheel ad is set to autoplay, during playback, the ad keeps buffering.
- In Safari 11, if the "Allow All Auto-play" option is enabled, after playing a Pulse pre-roll ad, the main video plays but the sound is muted and no unmute button appears; the viewer must unmute by clicking the volume button.
- In Safari 11, if the viewer clicks the Replay button after viewing the video, the replay button still shows while the video is replaying.
- The player `initialTime` parameter is sometimes not respected by the Safari browser or by Chrome mobile on Android devices.
- The `initialBitrate` parameter is not working in this release.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.18.13), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.18.13/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.18.13/skin-plugin/assets/images/ooyala.png	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Video Recommendation (Discovery) plugin	//player.oyala.com/static/v4/stable/4.18.13/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.oyala.com/static/v4/stable/4.18.13/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.oyala.com/static/v4/stable/4.18.13/video-plugin/main_html5.min.js	included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.oyala.com/static/v4/stable/4.18.13/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.18.13/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.oyala.com/static/v4/stable/4.18.13/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.oyala.com/static/v4/stable/4.18.13/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.oyala.com/static/v4/stable/4.18.13/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/AppMeasurement.js • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/VisitorAPI.js • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/conviva-core-sdk.min.js • //player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.18.13/analytics-plugin/googleAnalytics.min.js	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.18.13/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.18.13/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.17.8 RELEASED (2017-10-26)

This is a patch release only. No new features were introduced in this release.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Fixed in this Release

- Fixed an issue with videoView callbacks in the Freewheel ad plugin.

Known Issues in this Release

- When Windows Narrator is speaking the scrubber bar name in the play state, the narrator unexpectedly stops. This is caused by an issue in Windows Narrator.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.17.8), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.17.8/core.min.js	included

Skin Resources

HTML5 Skin	//player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/html5-skin.min.js	included
------------	---	----------



Resource	Name of File to Host	Included in Standard Embed Code?
Skin CSS	//player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> //player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/assets/images/ooyala-watermark.png //player.ooyala.com/static/v4/stable/4.17.8/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.17.8/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.17.8/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.17.8/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.17.8/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.17.8/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.17.8/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.17.8/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.17.8/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.17.8/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.17.8/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.17.8/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		



Resource	Name of File to Host	Included in Standard Embed Code?
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.8/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.17.8/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.17.7 RELEASED (2017-10-11)

This is a patch release only. No new features were introduced in this release.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Fixed in this Release

- If the DASH manifest contains an invalid offset, the player now starts with the correct initial start time instead of skipping the first segment.



Known Issues in this Release

- When Windows Narrator is speaking the scrubber bar name in the play state, the narrator unexpectedly stops. This is caused by an issue in Windows Narrator.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.17.7), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.17.7/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none">//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/assets/images/ooyala-watermark.png//player.ooyala.com/static/v4/stable/4.17.7/skin-plugin/assets/images/ooyala.png	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.17.7/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.17.7/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.17.7/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.17.7/video-plugin/osmf_flash.min.js	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Akamai HD Video Plugin for Akamai Packaged HDS	//player.oyala.com/static/v4/stable/4.17.7/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.oyala.com/static/v4/stable/4.17.7/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.oyala.com/static/v4/stable/4.17.7/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.oyala.com/static/v4/stable/4.17.7/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.oyala.com/static/v4/stable/4.17.7/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.oyala.com/static/v4/stable/4.17.7/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.oyala.com/static/v4/stable/4.17.7/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/VideoHeartbeat.min.js • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/AppMeasurement.js • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/VisitorAPI.js • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/conviva-core-sdk.min.js • //player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.oyala.com/static/v4/stable/4.17.7/analytics-plugin/Nielsen.min.js	not included



Resource	Name of File to Host	Included in Standard Embed Code?
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.17.7/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.17.6 RELEASED (2017-10-02)

This is a patch release only. No new features were introduced in this release.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Fixed in this Release

- Fixed an issue that caused closed captions to fail to render with Live Streams.
- Fixed an issue related to playback of HLS live streams containing only an audio track.
- In some browsers, the Closed Caption language was displayed twice when the language was changed. This issue has been fixed.
- Using the IMA plugin with AdRules, podded ads previously did not play the second ad.
- Fixed an issue related to accessibility. When using a screen reader in full-screen mode, the screen reader would sometimes read elements behind the player.

Known Issues in this Release

- When Windows Narrator is speaking the scrubber bar name in the play state, the narrator unexpectedly stops. This is caused by an issue in Windows Narrator.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.
- As part of Google's Flash deprecation effort, the Google IMA SDK version used in this Player release no longer supports Flash playback (VPAID ads).
- If you are self-hosting skin.json, if you deploy this release (4.17.6), you also need to pull the most recent version of the skin.json file and apply any of your customizations.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.17.6/core.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.17.6/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.17.6/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.17.6/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.17.6/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.17.6/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.17.6/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.17.6/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.17.6/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.17.6/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.17.6/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.17.6/ad-plugin/freewheel.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.17.6/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.6/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.17.6/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

SAFARI 11 TO BLOCK UNMUTED AUTOPLAY (2017-09-21)

Important: Apple announced that it will start blocking unmuted autoplay of videos on its Safari 11 desktop browser. This could impact your desktop playback experience. See the following topics for details:

- [Notification: Autoplay for Safari 11](#)



- Example script to add pop-up message asking viewers to add your site to the allowed list for unmuted autoplay. [Autoplay for Safari 11: Script Example](#)

OOYALA PLAYER V4.17.4 RELEASED (2017-09-13)

This Ooyala Player release introduces new features, enhances performance, and fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

New in this Release

- **Content Preloading:** Content preloading enables loading of the stream after the player is loaded and before the user starts playback (for non-autoplay settings), and, for Google IMA ads, during the 4th quartile of the ad break when ads are played (if ads are skipped before the fourth quartile of the ad is played, preloading of content will not occur). Preloading significantly speeds up the time-to-first-frame experience because the video is buffered before the viewer plays the videos. See [Page-level Parameters for Player V4](#) on page 99 for details. Also, preloading is disabled if the `initialTime` parameter is used.
- **Note:** Using content preloading can increase your overall video stream consumption because, for any given player embed, videos are streamed during preloading even when playback is not initiated. Typically, a video will preload the stream up to filling the buffer, the size of which will vary by browser.
- **Inline Playback on Safari Mobile:** This release introduces Inline playback (playing videos inline on the page rather than in full screen mode) on Safari Mobile (available on iOS 10+). The default for Safari Mobile remains full screen. To enable inline playback for Safari Mobile on iOS 10+ devices, use the `iosPlayMode` page-level parameter (example: `"iosPlayMode": "inline"`). See [Page-level Parameters for Player V4](#) on page 99 for details.
- **Skippable Ads on Safari Mobile using Google IMA:** In conjunction with inline playback on Safari Mobile, skippable ads can be served to iOS 10+ devices if the player is playing inline. To enable skippable ads on Safari Mobile using Google IMA, use the `google-ima-ads-manager.enableIosSkippableAds` page-level parameter (see [Google IMA Ad Parameters](#) on page 157):

```
"google-ima-ads-manager":  
  { "enableIosSkippableAds":true }
```

Note: If inline playback and skippable ads are enabled, then any ad break will shrink to play in-line, even if the user was viewing the content full screen prior to the ad break. Otherwise, the ad would play in the background.

- **Faster time to playback:** Improved thumbnail logic enables faster time to playback, particularly for Chrome Mobile.
- **Accessible Player Controls:** In this release, the Ooyala now supports the WCAG 2.0 standard for Accessibility.
- **Tool Tips for Player Controls:** This player release introduces tooltips in the Player UI. This enables popup explanatory text on a mouse hover action. Tooltips are disabled by default. To enable tooltips, edit the skin.JSON file and set the `tooltips.enabled` setting to `"enabled":true`:

```
{  
  "controlBar":{  
    "tooltips":{  
      "enabled":true  
    }  
  }  
}
```

For more information, see [Customizing the Player V4 Skin with skin.json](#) on page 120.

Fixed in this Release



- Fixed an issue with device registration with Widevine DRM that affected versions 4.16.x.

Known Issues in this Release

- While speaking, the scrubber bar name in the play state in the Windows Narrator is getting stuck (Issue needing resolved within Windows Narrator)
- Screen reader able to focus on page elements even though video is in full-screen.
- On older versions of Safari (prior to v10.1.2), there are problems with the focus indicator getting stuck and screen reader issues. Upgrading to newer version of Safari (v10.1.2 or higher) resolves these problems.
- In some browsers, the Close Caption language is displaying in two different languages.

Important Notes for this Release

- DRM content is not supported with incognito or private browser mode.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.17.4/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.17.4/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.17.4/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.17.4/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.17.4/video-plugin/main_html5.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.17.4/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.17.4/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.17.4/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.17.4/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.17.4/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.17.4/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.17.4/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.17.4/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/googleAnalytics.min.js	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.17.4/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.17.4/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.16.12 RELEASED (2017-09-06)

This Ooyala Player release fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Fixed in this Release

- Fixed an issue with live streaming where playback would stall if segments requests were slow to respond.
- Worked around an issue introduced in Safari on Mac OS 10.4.5 that was primarily seen when seeking beyond a start point larger than 40 minutes.

Known Issues in this Release

- Live Closed Captions is not available with the Bitmovin Video plugin in this release.
- The Microsoft Edge browser still uses HTML5 instead of Flash even if the "platform": "flash" parameter is passed.
- DASH DRM device registration fails and returns a DRM server error.

Important Notes for this Release

- Starting with this release, DRM content is not supported with incognito or private browser mode.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.16.12/core.min.js	included

Skin Resources



Resource	Name of File to Host	Included in Standard Embed Code?
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.16.12/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.16.12/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.16.12/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.16.12/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.16.12/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.16.12/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.16.12/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.16.12/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.16.12/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.16.12/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.16.12/ad-plugin/freewheel.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.16.12/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.16.12/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.16.12/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.16.10 RELEASED (2017-08-22)

This Ooyala Player release introduces new features, enhances performance, and fixes previous issues.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.



Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.oyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

- Accessibility Enhancements:
 - Added support for screen readers, such as JAWS, Apple VoiceOver, and Accessibility mode on mobile devices for Player controls using Aria tags.
 - Added tooltips for control video player control buttons. To enable tooltips:
 - As an page-level player param: `{"skin": {"inline": {"controlBar": {"tooltips": {"enabled": true}}}}}`
 - In skin.json: Set this to TRUE: `controlBar.tooltips.enabled`
- Added support for the `description_url` macro for IMA and VAST ad plugins. Previously, this needed to be added manually. This URL is extremely important, especially for publishers using the Network Partner Management (NPM) feature. The description URL is compared to an NPM publisher's list of classified domains to determine whether ads are eligible to serve. If the description URL does not match a classified domain, no ads will be served.
- Page-level plugin overrides for the standard embed code. See [Specifying Plugins for the Standard Embed Code](#) on page 75.

Fixed in this Release

- Changed default timeout for the IMA plugin for media from 8s to 15s.
- Fixed an issue with the VAST ad plugin that caused some VPAID impressions to not be tracked.

Known Issues in this Release

- On IE and Edge browsers, when navigating to full-screen mode, the tab position moves to the Pause/Play button instead of remaining with the Full Screen button.
- On Windows 7 + IE 11, the control bar is not visible from the player when used with the JAWS screen reader. However, the screen reader is otherwise able to interface with the player control normally.
- Share and Closed Caption buttons are not enabled in accessibility mode.
- DASH DRM device registration fails and returns a DRM server error.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
----------	----------------------	----------------------------------

Core Player	<code>//player.oyala.com/static/v4/stable/4.16.10/core.min.js</code>	included
-------------	--	----------

Skin Resources

HTML5 Skin	<code>//player.oyala.com/static/v4/stable/4.16.10/skin-plugin/html5-skin.min.js</code>	included
------------	--	----------

Skin CSS	<code>//player.oyala.com/static/v4/stable/4.16.10/skin-plugin/html5-skin.min.css</code>	included
----------	---	----------



Resource	Name of File to Host	Included in Standard Embed Code?
Skin Config File	//player.ooyala.com/static/v4/stable/4.16.10/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.16.10/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> //player.ooyala.com/static/v4/stable/4.16.10/skin-plugin/assets/images/ooyala-watermark.png //player.ooyala.com/static/v4/stable/4.16.10/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.16.10/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.16.10/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.16.10/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.16.10/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.16.10/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.16.10/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.16.10/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.16.10/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.16.10/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.16.10/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.16.10/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		



Resource	Name of File to Host	Included in Standard Embed Code?
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.16.10/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.16.10/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.16.8 RELEASED (2017-08-03)

This Ooyala Player release provides maintenance fixes that improve performance.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.ooyala.com/static/v4/stable/latest/
```



This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

Fixed in this Release

- Fixed an issue that would cause a 3-second floor on player initialization under certain race conditions. This change reduced player load latency for the specific race conditions in which it is realized.
- Fixed an issue in which video playback would skip the first few seconds with certain DRM-protected Dash streams.
- Fixed an issue in which the Pulse Ad Manager did not immediately set itself to a ready state when there were no ads.
- MyAds now loads only as needed when using the standalone player.

Known Issues in this Release

- The player does not resume content on iPad devices after a Google IMA error resulting from attempts to play Flash ads (VPAID 1.0).
- DASH DRM device registration fails and returns a DRM server error.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.16.8/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.16.8/skin-plugin/assets/images/ooyala.png	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.16.8/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.16.8/video-plugin/bit_wrapper.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.16.8/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.16.8/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.16.8/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.16.8/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.16.8/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.16.8/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.16.8/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.16.8/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.16.8/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.16.8/analytics-plugin/conviva.min.js 	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.16.8/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.oyala.com/static/v4/stable/4.16.8/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/oyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.oyala.com/static/v4/stable/4.16.8/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.14.9 RELEASED (2017-07-24)

This maintenance release fixes an issue that would cause a 3-second floor on player initialization under certain race conditions when ad plugins are used.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.oyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

Fixed in this Release

- Fixed an issue that would cause a 3-second floor on player initialization under certain race conditions. This change reduced player load latency for the specific race conditions in which it is realized.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.oyala.com/static/v4/stable/4.14.9/core.min.js	included

Skin Resources



Resource	Name of File to Host	Included in Standard Embed Code?
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> //player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/assets/images/ooyala-watermark.png //player.ooyala.com/static/v4/stable/4.14.9/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.14.9/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.14.9/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.14.9/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.14.9/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.14.9/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.14.9/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.14.9/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.14.9/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.14.9/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.14.9/ad-plugin/freewheel.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.14.9/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.9/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.14.9/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER V4.15.7 RELEASED (2017-07-10)

This Ooyala Player release provides enhancements to the bit_wrapper plugin as well as important bug fixes.

Important: This player version was *removed* from //player.ooyala.com/static/v4/production/latest due to a known issue with encrypted DASH streams. However, it is still available in the direct



version links (see below). In this player version, the first segment of the video is skipped (the first 4-10s of the video, depending on the encoding settings). This version is therefore not recommended for use with DASH streams encrypted with Widevine Modular DRM.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.ooyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

The bit_wrapper plugin was enhanced in the following ways:

- It now uses an upgraded version (v7.2) of Bitmovin.
- It now bundles all dependencies except for the .swf file needed for Flash playback. Therefore, if you are using the location or locationBaseUrl page-level parameter, you will only need to specify this for the .swf dependency (used for Flash-based playback).
- ABR is more responsive - it resolves faster to the optimal bitrate for the current network conditions.

Fixed in this Release

- Fixed an issue that would cause a 3-second floor on player initialization under certain race conditions. This change reduced player load latency for the specific race conditions in which it is realized.
- Video assets with no audio track are now playable with the Bitmovin plugin.
- VPAID ads that send player events outside of the ad break are now ignored.
- Aria tags and tab controls have been added for first-level Player elements, enabling readability for Screen Readers.
- For Chrome Mobile browsers, disabled cache-busting in the HLS manifest that was no longer needed.
- Enhanced the Playlist plugin so that, if a Playlist API call fails, it now broadcasts the message on the message bus and continues with the embed code specified initially.
- If the Playlist plugin fails, it now reverts back to the main video specified on player creation.
- Fixed an issue with the autoplayAfterInitialPlay parameter, which was previously autoplaying the initial play also.
- With Windows 7 / Explorer 11, MP4 fallback now works when the selected stream is DASH.

Known Issues in this Release

- When viewing content during an ad break with full screen, the player may transition out of full screen automatically.
- Not all Player controls meet accessibility requirements. Partial support was added in this release for Screen Reader compatibility.
- DASH is currently disabled for Safari (HTML5 and Flash). Only HLS will be used for Safari with this release.
- Cross-device Resume (XDR) is currently not supported when Ooyala Ads (My Ads) are used.
- DASH DRM device registration fails and returns a DRM server error.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.



Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.15.7/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.15.7/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.15.7/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.15.7/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.15.7/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.15.7/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.15.7/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.15.7/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.15.7/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.15.7/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.15.7/ad-plugin/google_ima.min.js	included



Resource	Name of File to Host	Included in Standard Embed Code?
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.15.7/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.15.7/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.15.7/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.15.7/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)



OOYALA PLAYER V4.14.8 RELEASED (2017-06-14)

This maintenance release fixes playback issues on a very specific Chrome version (48.0.2564.109), fixes issues with certain types of VPAID creatives and the Google IMA ad plugin, and enhances error message details.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.ooyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

Fixed in this Release

- For a very specific version of Chrome (48.0.2564.109), this release disables the bit_wrapper.js plugin from being used. If you have significant traffic from this browser (which is uncommon), you will need to include main_html5.js for playback on this specific browser (both are included in the Standard Player embed).

Note: The main_html5 plugin does not support DRM. Therefore, DRM will not work with Chrome v48.0.2564.109 with this player release.

- For “Network” type errors, this release logs the URL of the request with the error information.
- This release fixes failures originating from certain types of VPAID creatives and the Google IMA ad plugin.

Known Issues in this Release

- For Google IMA ads on iOS 10.1.1 and iPhone 6, ads play in normal-screen mode (not full-screen mode) and uses video controls.
- On some iOS devices, after a VPAID mid-roll ad plays, the video starts playing from the initial point of the video.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.14.8/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/skin.json	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Skin iFrame	//player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/assets/images/ooyala-watermark.png • //player.ooyala.com/static/v4/stable/4.14.8/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.14.8/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.14.8/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.14.8/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.14.8/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.14.8/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.14.8/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.14.8/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.14.8/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.14.8/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.14.8/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.14.8/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/VideoHeartbeat.min.js 	not included



Resource	Name of File to Host	Included in Standard Embed Code?
	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/omniture.min.js 	
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.8/analytics-plugin/Nielsen.min.js	not included
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.14.8/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

PLAYBACK DEPRECATIONS (2017-06-04)

The following playback functionality has been deprecated and is scheduled to be disabled:

- Akamai HD Video Plugin for Akamai Packaged HDS for Player V4
- OSMF Flash Video Plugin for HDS for Player V4
- Player V3
- Player V2

For details and alternatives, see the [OVP Release Notes](#).

BACKLOT SUPPORT FOR STANDARD PLAYER EMBED CODES (2017-05-31)

Backlot now provides you with the option to choose between a standard and advanced embed code (see MANAGE > Embed). If you use the standard Player embed code, you automatically include certain plug-



ins so that you don't need to add them to the web page. For details, see the [Ooyala Backlot Release Notes](#).

OOYALA PLAYER RELEASE V4.14.7 (2017-05-17)

This player version adds Japanese localization to the player skin and a new page-level parameter for the `playlists.js` plugin.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.ooyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

This Ooyala player release includes the following enhancements:

- **Japanese localization.** This release adds Japanese localization to Player V4 in the `skin.json` file. Refer to the Localization property in the [Player V4 JSON Schema documentation](#).
- **New page-level parameter for Playlists.** You can use the new `useFirstVideoFromPlaylist` parameter to specify whether the player uses the first video from the playlist (true) or not (false, the default). Before you enable this feature, contact Ooyala Tech Support to have your player plugin registered. If the plugin is not registered, playback will fail. See [Page-level Parameters for Player V4](#) on page 99.

Important Notes for this Release

- If you are self-hosting `skin.json` (see [Hosting Player V4 Resources](#) on page 81), if you deploy this release (4.14.7), you also need to pull the most recent version of the `skin.json` file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Fixed in this Release

- Fixed an issue on Windows 7 and IE 11 with Flash disabled. The `main_html5` video plugin defaults to MP4.
- Fixed a mixed content issue with the Nielsen plugin.
- Fixed an issue related to Chrome v58 and streams with Widevine DRM + DASH.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.14.7/core.min.js	included

Skin Resources

HTML5 Skin	//player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/html5-skin.min.js	included
------------	---	----------



Resource	Name of File to Host	Included in Standard Embed Code?
Skin CSS	//player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none"> //player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/assets/images/ooyala-watermark.png //player.ooyala.com/static/v4/stable/4.14.7/skin-plugin/assets/images/ooyala.png 	not included
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.14.7/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.14.7/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.14.7/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.14.7/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.14.7/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.14.7/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.14.7/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.14.7/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.14.7/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.14.7/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.14.7/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		



Resource	Name of File to Host	Included in Standard Embed Code?
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/omniture.min.js 	not included
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.14.7/analytics-plugin/Nielsen.min.js	not included
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.ooyala.com/static/v4/stable/4.14.7/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER RELEASE V4.13.9 (2017-05-15)

This player version adds page-level parameters for specifying resources associated with the [bit_wrapper.min.js](#) plugin and enhances error descriptions that are published to the message bus.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:



//player.ooyala.com/static/v4/stable/latest/

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

This Ooyala player release includes the following enhancements:

- **New page-level parameters for the Bitmovin plugin.** This player version adds two new page-level parameters (`locationBaseUrl` and `location`, which are specified within a `bit-wrapper` object). These are used in only certain circumstances to specify static resource paths to the `bit_wrapper.min.js` plugin and associated resources. See [Page-level Parameters for Player V4](#) on page 99.
- **Error descriptions published to the message bus.** For some playback errors, details have been added to the error response that identify the Ooyala API source of the errors, which will assist with troubleshooting their origin.

Important Notes for this Release

- If you are self-hosting skin.json (see [Hosting Player V4 Resources](#) on page 81), if you deploy this release (4.13.9), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host	Included in Standard Embed Code?
Core Player	//player.ooyala.com/static/v4/stable/4.13.9/core.min.js	included
Skin Resources		
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/html5-skin.min.js	included
Skin CSS	//player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/html5-skin.min.css	included
Skin Config File	//player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/skin.json	not included
Skin iFrame	//player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/iframe.html	not included
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.13.9/skin-plugin/assets/images/ooyala.png	not included



Resource	Name of File to Host	Included in Standard Embed Code?
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.13.9/other-plugin/discovery_api.min.js	included
Video Plugins		
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.13.9/video-plugin/bit_wrapper.min.js	included
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.13.9/video-plugin/main_html5.min.js	included
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.13.9/video-plugin/osmf_flash.min.js	not included
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.13.9/video-plugin/akamaiHD_flash.min.js	not included
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.13.9/video-plugin/youtube.min.js	not included
Ad Plugins		
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.13.9/ad-plugin/pulse.min.js	included
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.13.9/ad-plugin/ssai_pulse.min.js	not included
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.13.9/ad-plugin/google_ima.min.js	included
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.13.9/ad-plugin/freewheel.min.js	included
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.13.9/ad-plugin/ad_manager_vast.min.js	included
Analytics Plugins		
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.13.9/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.13.9/analytics-plugin/AppMeasurement.js 	not included



Resource	Name of File to Host	Included in Standard Embed Code?
	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/VisitorAPI.js • //player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/omniture.min.js 	
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.	not included
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/conviva-core-sdk.min.js • //player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/conviva.min.js 	not included
Google Analytics plugin	//player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/googleAnalytics.min.js	not included
Nielsen Analytics plugin	//player.oyala.com/static/v4/stable/4.13.9/analytics-plugin/Nielsen.min.js	not included
YOUTBORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/oyalav4/stable/sp.min.js	not included
Other Feature Plugins		
Playlists plugin	//player.oyala.com/static/v4/stable/4.13.9/other-plugin/playlists.js	not included

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

OOYALA PLAYER RELEASE V4.13.5 (2017-05-01)

This Ooyala player release enhances player performance by reducing the time to load and initialize the embedded player.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.oyala.com/static/v4/stable/latest/
```



This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

This maintenance release enhances player load time performance (shortens the time to load and initialize the embedded player).

Important Notes for this Release

- If you are self-hosting skin.json (see [Hosting Player V4 Resources](#) on page 81), if you deploy this release (4.13.5), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
Core Player	//player.ooyala.com/static/v4/stable/4.13.5/core.min.js
Skin Resources	
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/html5-skin.min.js
Skin CSS	//player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none">• //player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/assets/images/ooyala-watermark.png• //player.ooyala.com/static/v4/stable/4.13.5/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.13.5/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.13.5/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.13.5/video-plugin/main_html5.min.js
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.13.5/video-plugin/osmf_flash.min.js
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.13.5/video-plugin/akamaiHD_flash.min.js



Resource	Name of File to Host
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.13.5/video-plugin/youtube.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.13.5/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.13.5/ad-plugin/ssai_pulse.min.js
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.13.5/ad-plugin/google_ima.min.js
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.13.5/ad-plugin/freewheel.min.js
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.13.5/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.13.5/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.13.5/other-plugin/playlists.js

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)



OOYALA PLAYER RELEASE V4.13.4 (2017-04-04)

This Ooyala player release introduces new Ooyala-hosted paths for player resources and changes the default to HTML5 playback for the bit_wrapper.min.js plugin.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Ooyala-hosted Path for Customers with Mixed Content Conditions

Note: For customers with mixed content conditions (refer to this [Google article](#)) on a site running over HTTPS, use the following path instead:

```
//player.ooyala.com/static/v4/stable/latest/
```

This path is locked on Player V4 Web v4.10.6, which prefers Flash playback as the priority. This circumvents mixed content enforcement by browsers (assuming Flash is not blocked).

New in this Release

This release introduces the following functionality:

- **HTML5-first is the default for bit_wrapper.min.js.** The Bitmovin plugin now defaults to HTML5. You no longer need to set the platform page-level parameter to "html5" to prioritize HTML5 playback. Note that HTML5 is subject to mixed content enforcement by browsers. If you want to prefer Flash-based playback at start, you must set the player parameter "platform": "flash" at the page level. See [Page-level Parameters for Player V4](#) on page 99.
- **New Ooyala hosting paths.** Starting with Player Web 4.12.6, the Ooyala Player is hosted, distributed, and maintained on the following URL paths:

```
//player.ooyala.com/static/v4/production/
```

This path is recommended for most Ooyala customers who do not have mixed content conditions (refer to this [Google article](#) for more on mixed content). This path is maintained as one release behind the latest Ooyala Player version (below).

```
//player.ooyala.com/static/v4/production/latest/
```

This path contains the latest Ooyala player version that has been certified for release. This path is recommended for Ooyala customers who need capabilities only available in the most recent player release. These paths are automatically updated with new Player V4 releases and are subject to our 99.9% uptime SLA. For details, see [Ooyala-hosted Player V4 Resources](#) on page 77.

- **Static Ooyala hosting paths:**

```
//player.ooyala.com/static/v4/stable/latest/
```

Required for mixed-content conditions. This path is set indefinitely to Player V4 Web v4.10.6, which is last player version that does not cause browsers to enforce mixed content rules with Flash playback.

```
//player.ooyala.com/static/v4/stable/4.13.4/
```

The direct version paths are available for this release (see below). However, if you point your site to the specific version paths, your site will not automatically get player updates.

- **autoPlayUpNextVideosOnly** page-level parameter - To disable the automatic playing of Up Next or Discovery videos after the main video has played, set `playerParams.autoPlay` to false and specify `playerParams.autoPlayUpNextVideosOnly` as false. See [Page-level Parameters for Player V4](#) on page 99.

Fixed Issues

Fixes in this release include:

- GetDuration now reports all values in seconds for all browsers. It also now reports "infinity" for Live assets.



- For the bit_wrapper.min.js plugin, fixed the following issues:
 - Users encountered errors when using Playlists.
 - The player maintained a lower bitrate despite sufficient bandwidth for a higher bitrate.
 - Ooyala Ads were not playing properly on some Android devices.

Important Notes for this Release

- If you are self-hosting skin.json (see [Hosting Player V4 Resources](#) on page 81), if you deploy this release (4.13.4), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues in This Release Related to Changes

The following are known issues for this release:

- On Windows 7 with Internet Explorer ads, VAST ad playback with the bit_wrapper.min.js plugin is not working correctly.
- On Windows 10 with the Edge Browser 38.x and the bit_wrapper.min.js plugin, playback does not work correctly.
- For videos containing ads (only), XDR does not always restart at the correct time.
- For Safari and iOS, the CC button may appear even in the absence of a closed caption file.

Direct Version Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

Resource	Name of File to Host
Core Player	//player.ooyala.com/static/v4/stable/4.13.4/core.min.js
Skin Resources	
HTML5 Skin	//player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/html5-skin.min.js
Skin CSS	//player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/html5-skin.min.css
Skin Config File	//player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/skin.json
Skin iFrame	//player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/iframe.html
Images and Fonts	<ul style="list-style-type: none"> //player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/assets/images/ooyala-watermark.png //player.ooyala.com/static/v4/stable/4.13.4/skin-plugin/assets/images/ooyala.png
Video Recommendation (Discovery) plugin	//player.ooyala.com/static/v4/stable/4.13.4/other-plugin/discovery_api.min.js
Video Plugins	
Bitmovin Video Plugin for DASH and HLS	//player.ooyala.com/static/v4/stable/4.13.4/video-plugin/bit_wrapper.min.js
Main Video Plugin for HLS and MP4	//player.ooyala.com/static/v4/stable/4.13.4/video-plugin/main_html5.min.js



Resource	Name of File to Host
OSMF Flash Video Plugin for HDS	//player.ooyala.com/static/v4/stable/4.13.4/video-plugin/osmf_flash.min.js
Akamai HD Video Plugin for Akamai Packaged HDS	//player.ooyala.com/static/v4/stable/4.13.4/video-plugin/akamaiHD_flash.min.js
Ooyala Player Plugin for YouTube iFrame (Deprecated)	//player.ooyala.com/static/v4/stable/4.13.4/video-plugin/youtube.min.js
Ad Plugins	
Ooyala Pulse Ad Plugin	//player.ooyala.com/static/v4/stable/4.13.4/ad-plugin/pulse.min.js
Ooyala SSAI Pulse Plugin for Live Ad Insertion	//player.ooyala.com/static/v4/stable/4.13.4/ad-plugin/ssai_pulse.min.js
Google IMA Plugin	//player.ooyala.com/static/v4/stable/4.13.4/ad-plugin/google_ima.min.js
FreeWheel Plugin	//player.ooyala.com/static/v4/stable/4.13.4/ad-plugin/freewheel.min.js
VAST and VPAID Plugin	//player.ooyala.com/static/v4/stable/4.13.4/ad-plugin/ad_manager_vast.min.js
Analytics Plugins	
Adobe Analytics (Omniture) Plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/VideoHeartbeat.min.js • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/AppMeasurement.js • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/VisitorAPI.js • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/omniture.min.js
comScore Plugin	Contact your Ooyala account manager for access to this plugin. comScore created and maintains this plugin.
Conviva Analytics plugin	<ul style="list-style-type: none"> • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/conviva-core-sdk.min.js • //player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/conviva.min.js
Google Analytics plugin	//player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/googleAnalytics.min.js
Nielsen Analytics plugin	//player.ooyala.com/static/v4/stable/4.13.4/analytics-plugin/Nielsen.min.js
YOUTORA Analytics Plugin	//smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js
Other Feature Plugins	
Playlists plugin	//player.ooyala.com/static/v4/stable/4.13.4/other-plugin/playlists.js



Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

PLAYER V4 WEB RELEASE V4.12.6 (2017-03-13)

This Player V4 release is compatible with changes introduced in Chrome v58 related to Encrypted Media Extensions (EME) and encrypted streams (DASH with Widevine DRM). It introduces the Conviva Analytics plugin, mobile browser support for My Ads, localization and non-linear ad support for Google IMA, and changes Discovery autoplay defaults.

This player version addresses an upcoming issue in Chrome v58, which will block encrypted streams (DASH with Widevine DRM) on insecure origins. In addition to using this player version, customers using DASH + Widevine DRM must now secure any applicable websites for DASH + Widevine streams to continue operating starting with this release of the Chrome browser. Chrome v58+ requires HTTPS for using the Encrypted Media Extensions (EME) API.

Note: This player is subject to mixed content enforcement by web browsers, including for Flash streams. If embedding the Ooyala V4 Player on a site running over HTTPS, confirm that you have no HTTP streams (*i.e. mixed content*) before using this player version. If you have an HTTP site, this does not apply. Due to some customers having mixed content conditions, Player V4 release 4.12.6 cannot be placed on the .../stable/latest path without impacting their sites. As a result, this release is available only by pointing directly to the /4.12.6 paths, as shown in the default player links (below) for the core player and plugins.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Default Player Links

For reference, here are the Ooyala direct paths for the current version.

Note: These paths should not be used in your site unless absolutely necessary, as you will not get player updates automatically.

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins, CSS, font, and image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.12.6/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.12.6/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.12.6/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.12.6/skin-plugin/html5-skin.min.css>
 - **Video Plugins (at least one required)**
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.12.6/video-plugin/bit_wrapper.min.js
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.12.6/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.12.6/video-plugin/osmf_flash.min.js



- Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.12.6/video-plugin/akamaiHD_flash.min.js
- Plugin for YouTube iFrame (Deprecated): <http://player.ooyala.com/static/v4/stable/4.12.6/video-plugin/youtube.min.js>
- **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.12.6/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.12.6/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/ad-plugin/freewheel.min.js>
- **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/analytics-plugin/Nielsen.min.js>
 - Conviva Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.12.6/analytics-plugin/conviva-core-sdk.min.js> and <https://player.ooyala.com/static/v4/stable/4.12.6/analytics-plugin/conviva.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.12.6/other-plugin/discovery_api.min.js
- **Playlists Plugin**
 - <http://player.ooyala.com/static/v4/stable/4.12.6/other-plugin/playlists.js>
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.12.6/ad-plugin/ssai_pulse.js

New in this Release

This release introduces the following functionality:

- **Conviva Analytics plugin.** Player V4 now supports integration with Conviva analytics (see [Conviva Analytics Plugin](#) on page 197).
- My Ads now run on mobile browsers (see [Delivering My Ads](#) on page 184).
- The Google IMA ad plugin now supports:
 - localization settings that are configured via the Google IMA SDK for HTML5
 - non-linear ads (except overlays) when using Ad Rules/VMAP
 See [Google IMA Ad Plugin](#) on page 147.
- Discovery now defaults to autoplay, even when the main video is not set to autoplay on the page. Previously, Discovery defaulted to autoplay only if the embedded player was set to autoplay. See [Discovering Content in Player V4](#) on page 134.

Fixed Issues

Fixes in this release include:

- This player version is compatible with changes introduced in Chrome v58 related to Encrypted Media Extensions (EME) and encrypted streams (DASH with Widevine DRM).
- Fixed an issue with ad playback when using the Freewheel ad plugin. The Backlot Freewheel setting (Video Asset Network ID) now takes precedence of the asset ID. To override this, specify the asset ID (embed code) in the page-level parameter under the `fw_video_asset_id` property.



- Duplicated play buttons were seen on iOS devices when play was paused.
- Anamorphic video formats were stretching on Firefox browsers.
- Intermittent issues with closed captions in multiple languages occurred in various browsers.

Important Notes for this Release

- If you are self-hosting skin.json, if you deploy this release (4.12.6), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues in This Release Related to Changes

The following are known issues for this release:

- For the Bitmovin plugin for DASH and HLS (bit_wrapper.min.js):
 - VAST ads are skipped when played on Windows 7 with Internet Explorer 11.
 - MP4 streams are not playing on the Microsoft Edge v38 browser on Windows 10 devices. This primarily impacts My Ads, which require the MP4 format.
- If multiple players use the YouTube plugin on the same web page, playback is blocked after the first embedded YouTube asset plays.
- When using Ooyala playlists, the page-level platform setting of “html5” is ignored.

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

Player V4 Web release v4.11.14 (2017-02-17)

Bitmovin Video Plugin for DASH and HLS: This release fixes adaptive bitrate streaming (ABR) performance issues for the Bitmovin video plugin. If you are self-hosting the Bitmovin plugin in this release, you also need to host additional Bitmovin files (see [Hosting Player V4 Resources](#) on page 81).

Note: This player is subject to mixed content enforcement by web browsers, including for Flash streams. If embedding the Ooyala V4 Player on a site running over HTTPS, confirm that you have no HTTP streams (*i.e. mixed content*) before using this player version. If you have an HTTP site, this does not apply. Due to some customers having mixed content conditions, Player V4 release 4.11.14 cannot be placed on the .../stable/latest path without impacting their sites. As a result, this release is available only by pointing directly to the /4.11.14 paths, as shown in the default player links (below) for the core player and plugins.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS](#), [font](#), and [image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.11.14/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.11.14/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.11.14/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.11.14/skin-plugin/html5-skin.min.css>



- **Video Plugins (at least one required)**
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.11.14/video-plugin/bit_wrapper.min.js
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.11.14/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.11.14/video-plugin/osmf_flash.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.11.14/video-plugin/akamaiHD_flash.min.js
 - Plugin for YouTube iFrame (Deprecated): <http://player.ooyala.com/static/v4/stable/4.11.14/video-plugin/youtube.min.js>
- **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.11.14/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.11.14/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.11.14/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.11.14/ad-plugin/freewheel.min.js>
- **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.11.14/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.11.14/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.11.14/analytics-plugin/Nielsen.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.11.14/other-plugin/discovery_api.min.js
- **Playlists Plugin**
 - <http://player.ooyala.com/static/v4/stable/4.11.14/other-plugin/playlists.js>
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.11.14/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include:

- Playback froze on remote asset live streaming when the "platform": "html5" parameter is used.
- ABR switching was taking unusually long with eHLS assets.
- With playlists, only the first video played; subsequent videos did not play.
- Some live remote HLS live-streams (non-Ooyala hosted) did not start playing.

Important Notes for this Release

- If you are self-hosting skin.json, if you deploy this release (4.11.14), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues in This Release Related to Changes

The following are known issues for this release:

- My Ads does not work on mobile browsers.



- Using the VAST plugin, midroll ads get skipped on Safari Web and the Android Chrome browser.
- For some browsers on some Windows and Mac OS versions, Freewheel and Vast mid-roll ads might not play.
- For the Bitmovin plugin, Google IMA mid-roll ads might not play intermediately on some Mac OS environments.

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

Player V4 Web release v4.11.13 (2017-01-30)

Note: This player is subject to mixed content enforcement by web browsers, including for Flash streams. If embedding the Ooyala V4 Player on a site running over HTTPS, confirm that you have no HTTP streams (*i.e. mixed content*) before using this player version. If you have an HTTP site, this does not apply. Due to some customers having mixed content conditions, Player V4 release 4.11.13 cannot be placed on the .../stable/latest path without impacting their sites. As a result, this release is available only by pointing directly to the /4.11.13 paths, as shown in the default player links (below) for the core player and plugins.

- **My Ads support:** Player V4 now delivers pre-roll ads for My Ads (MP4 only) that you upload and configure in Backlot. See [Delivering My Ads](#) on page 184.
- **Overlay and Time Based Break Support in Ooyala Pulse Plugin.** Two new features are now supported in your integration between Player V4 and Pulse for ad serving. Request any overlays using the `pulse_non_linear_cuepoints` parameter and set the time in seconds in the `pulse_max_linear_break_duration` parameter to create or override the insertion policies for linear ad breaks in Pulse. See [Ooyala Pulse Ad Parameters](#) on page 164.
- **Specifying the first video from the playlist.** The new `useFirstVideoFromPlaylist` parameter allows you to specify the playlist's first video set as the page's initial `embedCode`, rather than the actual `embedCode` placed on the page. See [Using Playlists in Player V4](#) on page 136.
- **Controlling the Share screen with skin.json.** On the Share screen, you can now use `skin.json` (`shareScreen.socialContent` property) to configure which buttons to display (Twitter, Facebook, Google+, and Email), and in what order. If none are specified, then the Share tab is hidden. See [Ooyala Player Skin Documentation and JSON Schema](#).
- **Bitmovin Plugin Self-hosting:** If you are using the self-hosting the Bitmovin Video Plugin for DASH and HLS, you now need to host additional files associated with the Bitmovin plugin (see [Hosting Player V4 Resources](#) on page 81).
- **Mapping of Backlot Player UI parameters for V4 Players.** Some of the parameters (for example, tint color) are now recognized by Player V4. However, not all Backlot parameters are supported by Player V4. Additionally, Backlot Player UI settings have a lower priority compared to `skin.json` parameters. If you are including `skin.json` in your page, the settings in this file will override any Backlot settings. If you do not include `skin.json` in your deployment, the Backlot player settings will be used. In a planned upcoming release for Backlot, unsupported parameters for Player V4 will be removed to provide clarity on which specific Backlot player UI settings are fully supported.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS](#), [font](#), and [image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of `skin.json`, download the latest `skin.json` file and merge any changes in this file with your customized `skin.json` file.
- **Core Player (required)**



- <http://player.ooyala.com/static/v4/stable/4.11.13/core.min.js>
- **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.11.13/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.11.13/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.11.13/skin-plugin/html5-skin.min.css>
- **Video Plugins (at least one required)**
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.11.13/video-plugin/bit_wrapper.min.js
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.11.13/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.11.13/video-plugin/osmf_flash.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.11.13/video-plugin/akamaiHD_flash.min.js
 - Plugin for YouTube iFrame (Deprecated): <http://player.ooyala.com/static/v4/stable/4.11.13/video-plugin/youtube.min.js>
- **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.11.13/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.11.13/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.11.13/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.11.13/ad-plugin/freewheel.min.js>
- **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.11.13/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.11.13/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.11.13/analytics-plugin/Nielsen.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.11.13/other-plugin/discovery_api.min.js
- **Playlists Plugin**
 - <http://player.ooyala.com/static/v4/stable/4.11.13/other-plugin/playlists.js>
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.11.13/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include:

- In addition to user-selected Discovery videos, autoplayed Discovery videos are now included in Ooyala Analytics.
- Player V4 now autoplays Discovery videos on mobile browsers.
- On iOS devices, when a video was paused, the default player overlaid the native player play button, which blurred the button appearance.

Important Notes for this Release



- If you are self-hosting skin.json, if you deploy this release (4.11.13), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues in This Release Related to Changes

The following are known issues for this release:

- My Ads does not work on mobile browsers.
- For some browsers on some Windows and Mac OS versions, Freewheel and Vast mid-roll ads might not play.
- For the Bitmovin plugin, Google IMA mid-roll ads might not play intermediately on some Mac OS environments.
- For HLS streams on Windows 10 and Microsoft Edge browsers, you might encounter a "Playback Stream Error" while in Flash.

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)

2016-12-15 Release - Player V4 Version 4.10.6

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Introducing the Ooyala Player Series

The Ooyala Player Series provides instructional videos for the Ooyala Player. Check out our first video, [Ooyala Player Architecture](#), which talks about core Player features, plugins, and what role they have in the player experience. Future videos will address such topics as how to embed the player, how to interact with the Player API, and more.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS](#), [font](#), and [image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.10.6/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.10.6/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.10.6/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.10.6/skin-plugin/html5-skin.min.css>
 - **Video Plugins (at least one required)**
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.10.6/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.10.6/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.10.6/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.10.6/video-plugin/akamaiHD_flash.min.js



- Plugin for YouTube iFrame: <http://player.ooyala.com/static/v4/stable/4.10.6/video-plugin/youtube.min.js>
- **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.10.6/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.10.6/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.10.6/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.10.6/ad-plugin/freewheel.min.js>
- **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.10.6/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.10.6/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.10.6/analytics-plugin/Nielsen.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.10.6/other-plugin/discovery_api.min.js
- **Playlists Plugin**
 - <http://player.ooyala.com/static/v4/stable/4.10.6/other-plugin/playlists.js>
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.10.6/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include:

- **YouTube plugin:** Problems with HTTPS playback. HTTPS is now supported.
- **YouTube plugin:** Problems with switching (before playback starts) between multiple videos on the same page.
- **Bitmovin plugin:** If initialTime was set for a video, and you paused playback, you could not seek to the point prior to the initialTime setting.
- For Google IMA Ads, the adLoadTimeout default setting was 5 seconds. We have increased the default to 15 seconds. See [Configuring Ad Parameters](#) on page 180.

Important Notes for this Release

- If you are self-hosting skin.json, if you deploy this release (4.10.6), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues in This Release Related to Changes

The following are known issues for this release:

- For iOS and Android devices, subsequent videos in Discovery do not autoplay.
- For the Bitmovin plugin, closed captions don't correlate with the video if initialTime is > 0.
- For Safari and iOS, the CC button may appear even though a closed caption file is not present.

Player V4 Resources

- [Player V4 documentation](#)
- [Player V4 JavaScript API Reference](#)
- [Ooyala Player Skin Documentation and JSON Schema](#)



New Features in this Player Release

- **Playlist support.** You can now integrate playlists created with Ooyala Theme Builder. See [Using Playlists in Player V4](#) on page 136.
- **Watermark support.** In addition to custom watermarks (see [Adding a Custom Branding Overlay to Player V4](#) on page 132), Player V4 now supports watermarks configured in skin.json. You can configure the watermark location, transparency, click URL, and scaling. See the [Player V4 JSON Schema](#) documentation.
- **New showAdControls parameter for Google IMA.** Enabling showAdControls displays Ooyala's ad control bar while an IMA ad is playing. For details, see [Google IMA Ad Parameters](#) on page 157.
- **New parameters for VAST ads.** You can now use page-level parameters to specify the position and position type for VAST ads. For details, see [VAST and VPAID Ad Parameters](#) on page 175.
- **Default for adLoadTimeout parameter.** This has been increased from 5 seconds to 25 seconds. If you already overridden the default, consider removing the override or increasing it to around 25 seconds. For details, see [Configuring Ad Parameters](#) on page 180.
- **Discovery and YouTube.** If you use Ooyala Discovery with YouTube assets in Backlot, you can now mix YouTube assets with non-YouTube assets.

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins, CSS, font, and image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.10.4/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.10.4/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.10.4/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.10.4/skin-plugin/html5-skin.min.css>
 - **Video Plugins (at least one required)**
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.10.4/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.10.4/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.10.4/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.10.4/video-plugin/akamaiHD_flash.min.js
 - Plugin for YouTube iFrame: <http://player.ooyala.com/static/v4/stable/4.10.4/video-plugin/youtube.min.js>
 - **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.10.4/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.10.4/ad-plugin/ad_manager_vast.min.js



- Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.10.4/ad-plugin/google_ima.min.js
- FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.10.4/ad-plugin/freewheel.min.js>
- **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.10.4/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.10.4/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.10.4/analytics-plugin/Nielsen.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.10.4/other-plugin/discovery_api.min.js
- **Playlists Plugin**
 - <http://player.ooyala.com/static/v4/stable/4.10.4/other-plugin/playlists.js>
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.10.4/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include an issue in Microsoft Edge with the spinner not disappearing after scrubbing a video.

Known issues from previous Player V4 releases have been fixed if they do not appear on the Known Issues list below.

Important Notes for this Release

- If you are self-hosting skin.json, if you deploy this release (4.10.4), you also need to pull the most recent version of the skin.json file (and apply any of your customizations).
- Device registration for Dash DRM is currently not supported.

Known Issues

The following are known issues for this release:

- On iPad devices, closed captions disappear if you pause and play in full screen mode in the native browser.
- On iPhone devices and iOS 10, during playback, if you minimize the browser window from full screen to normal screen, closed captions might be duplicated horizontally or vertically.
- During playback with VPAID ads (both normal and full-screen mode), the forward seek is not work on iOS devices. This is due to a known issue in the Google IMA SDK.
- In the Firefox and Microsoft Edge browsers, device registration may result in a DRM server error.
- For the Bitmovin plugin, on replay, the pre-roll plays, but the main video does not play. Instead, the last frame shows from the previous play, and the Ad control bar shows for the main video.
- For the VAST and VPAID Ad Plugin, the Vast ad manager currently double counts tracking events if the ad is a VPAID 2.0 ad. Non-VPAID ads are not affected.
- Playlist pods do not work in this release.

2016-11-03 RELEASE - VERSION 4.9.8

New Features in this Player Release

- **Ooyala Player Plugin for YouTube iFrame.** Use this new plugin to play YouTube-hosted videos using Player V4 and the Player V4 skin. It allows you to syndicate video content from YouTube into your web



application. Integration involves adding an asset in Backlot that points to the video hosted on YouTube, and specifying the plugin on the page where you embed a player. For details, see [Playing YouTube Videos in Player V4 \(Deprecated\)](#) on page 97.

- **Support for ad frequency cap settings.** In Player V4, ads (pre-roll, mid-roll, and post-roll) now recognize the frequency cap setting in Backlot.
- **Support for ads in Live streams.** The Google IMA, Freewheel, and VAST plugins now support pre-roll and mid-roll ads for live streams. See [Ads and Live Streams](#) on page 184.
- **Ad Manager timeout control.** To assist with ad-fill rate issues related to timeout settings, you can now specify the `adManagerLoadTimeout` (an optional global embedded parameter) to specify the timeout for the loading of the Ad Manager module. For details, see [adManagerLoadTimeout](#).

Important: Browsers are starting to drop default support for Flash. Are you ready? See the [Player V4 Resources page](#) and the [Player Migration FAQ](#) on page 66 for details.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins, CSS, font, and image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.9.8/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.9.8/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.9.8/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.9.8/skin-plugin/html5-skin.min.css>
 - **Video Plugins (at least one required)**
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.9.8/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.9.8/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.9.8/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.9.8/video-plugin/akamaiHD_flash.min.js
 - Plugin for YouTube iFrame: <http://player.ooyala.com/static/v4/stable/4.9.8/video-plugin/youtube.min.js>
 - **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.9.8/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.9.8/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.9.8/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.9.8/ad-plugin/freewheel.min.js>
 - **Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.9.8/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.9.8/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.9.8/analytics-plugin/Nielsen.min.js>



- Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
- comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.9.8/other-plugin/discovery_api.min.js
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.9.8/ad-plugin/ssai_pulse.js

Fixed Issues

Known issues from previous Player V4 releases have been fixed if they do not appear on the Known Issues list below.

Important Note for this Release

- Device registration for Dash DRM is currently not supported.

Known Issues

The following are known issues for this release:

- On iPad devices, closed captions disappear if you pause and play in full screen mode in the native browser.
- On iPhone devices and iOS 10, during playback, if you minimize the browser window from full screen to normal screen, closed captions might be duplicated horizontally or vertically.
- During playback with VPAID ads (both normal and full-screen mode), the forward seek is not work on iOS devices. This is due to a known issue in the Google IMA SDK.
- In the Firefox and Microsoft Edge browsers, device registration may result in a DRM server error.
- For the Bitmovin plugin, on replay, the pre-roll plays, but the main video does not play. Instead, the last frame shows from the previous play, and the Ad control bar shows for the main video.
- For the VAST and VPAID Ad Plugin, the Vast ad manager currently double counts tracking events if the ad is a VPAID 2.0 ad. Non-VPAID ads are not affected.

2016-09-29 RELEASE - VERSION 4.8.5

New Features in this Player Release

- **Flash-Free Player using HTML5 MSE.** The Bitmovin plugin for DASH and HLS (`bit_wrapper.min.js`) plays HLS using the HTML5 standard (and Media Source Extensions), enabling HLS on web browsers without the use of Flash. HTML5 MSE also supports playback of MPEG-DASH (clear and DRM-protected content). If you are using the remote assets setup, you must make sure that your remote stream (manifest/m3u8 and fragment/ts) is served via HTTPS. Contact your Ooyala support representative for details.
- **New Nielsen Analytics Plugin.** You can now use the Player V4 Nielsen Analytics plugin (`Nielsen.min.js`) to track Nielsen analytics for your V4 Ooyala Player. For details, see [Nielsen Analytics Plugin](#) on page 204.
- **Session Extension Support in Ooyala Pulse Ad Plugin.** Session extension is mainly used to insert extra ad breaks in a live stream. For more information, refer to `pull_callbacks` in [Ooyala Pulse Ad Parameters](#) on page 164.
- **Analytics Framework Enhancements:** Increased error message reporting from our player to the Analytics Framework. Refer to the [Analytics Framework API Reference](#) for more information.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS](#), [font](#), and [image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.



- If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.8.5/core.min.js>
 - Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.8.5/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.8.5/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.8.5/skin-plugin/html5-skin.min.css>
 - Video Plugins (at least one required)**
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.8.5/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.8.5/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.8.5/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.8.5/video-plugin/akamaiHD_flash.min.js
 - Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.8.5/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.8.5/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.8.5/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.8.5/ad-plugin/freewheel.min.js>
 - Analytics Plugins**
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.8.5/analytics-plugin/omniture.js>
 - Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.8.5/analytics-plugin/googleAnalytics.min.js>
 - Nielsen Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.8.5/analytics-plugin/Nielsen.min.js>
 - Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
 - Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.8.5/other-plugin/discovery_api.min.js
 - Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.8.5/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include:

- For the Google IMA plugin, the video stopped playing properly after playing a mid-roll ad.
- For the VAST and VPAID ad plugin, the Ad Playback and scrubber bar did not synchronize properly.

Other known issues from previous Player V4 releases have been fixed if they do not appear on the Known Issues list below.

Important Note for this Release

- Device registration for Dash DRM is currently not supported.

Known Issues



The following are known issues for this release:

- For the Bitmovin plugin, on replay, the pre-roll plays, but the main video does not play. Instead, the last frame shows from the previous play, and the Ad control bar shows for the main video.
- For the VAST and VPAID Ad Plugin, the Vast ad manager currently double counts tracking events if the ad is a VPAID 2.0 ad. Non-VPAID ads are not affected.

2016-09-14 RELEASE - VERSION 4.7.9

New Features in this Player Release

- **New Google Analytics Plugin for Player V4.** The Player V4 Google Analytics plugin (`googleAnalytics.min.js`) supports integrating Google Analytics with the Ooyala HTML5 web player versions 4.7.9 and above.
- **Support for Multiple Ad Managers on the Same Page.** You can now load and use multiple ad plugins (Freewheel, VAST, and Google IMA ad plugins) on the same page. If all three were loaded on a page, for example, and you had multiple embedded players on the page, Player A and Player B could both use IMA, Player C could use Freewheel, and Player D could use VAST. For details, see [Integrating Multiple Ad Managers](#) on page 177.
- **Enabling the Google IMA countdown timer.** For the Google IMA plugin, the `useGoogleCountdown` page-level parameter displays the Google IMA countdown timer while still enabling clickthroughs to be triggered by clicks on the video. For details, see [Google IMA-Specific Embedded Parameters](#).

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS, font, and image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of `skin.json`, download the latest `skin.json` file and merge any changes in this file with your customized `skin.json` file.
- **Core Player (required)**
 - <http://player.ooyala.com/static/v4/stable/4.7.9/core.min.js>
 - **Player Skin**
 - <http://player.ooyala.com/static/v4/stable/4.7.9/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.7.9/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.7.9/skin-plugin/html5-skin.min.css>
 - **Video Plugins (at least one required)**
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.7.9/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.7.9/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.7.9/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.7.9/video-plugin/akamaiHD_flash.min.js
 - **Ad Plugins**
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.7.9/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.7.9/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.7.9/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.7.9/ad-plugin/freewheel.min.js>
 - **Analytics Plugins**



- Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.7.9/analytics-plugin/omniture.js>
- Google Analytics Plugin: <https://player.ooyala.com/static/v4/stable/4.7.9/analytics-plugin/googleAnalytics.min.js>
- Player V4 YOUBORA Plugin: <https://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>
- comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- **Discovery Plugin**
 - http://player.ooyala.com/static/v4/stable/4.7.9/other-plugin/discovery_api.min.js
- **Live SSAI Pulse Plugin**
 - http://player.ooyala.com/static/v4/stable/4.7.9/ad-plugin/ssai_pulse.js

Fixed Issues

Fixes in this release include:

- 608/708 captioning was not available with Android HLS in Chrome. Android Chrome does not detect and include the caption in the <video> element textTrack.
- With the Google IMA ad plugin, the player blinked just before displaying an overlay ad.
- With the VAST and VPAID Ad Plugin, if you scrubbed the ad past an overlay, then the mid-roll ad was skipped.
- The initiaBitrate was not behaving consistently across some browsers.
- HLS live streaming was stuttering/freezing on desktop environments.
- DASH + Playready was not playing correctly on Microsoft Edge and Internet Explorer.

Other known issues from previous Player V4 releases have been fixed if they do not appear on the Known Issues list below.

Important Note for this Release

- Device registration for Dash DRM is currently not supported.

Known Issues

The following are known issues for this release:

- On some browsers with ad blocker off, sometimes the pre-roll does not play, or it plays but the pre-roll audio overlaps with main video audio.
- VAST-compliant mid-roll ads set up via Backlot are not displaying correctly.
- For some VPAID ads, the loading the ad in the player is slow or times out.
- With the Google IMA ad plugin, the player sometimes displays a black screen instead of the ad.
- For the VAST and VPAID Ad Plugin, if multiple mid-roll ads are queued, only one mid-roll ad plays.
- For the VAST and VPAID Ad Plugin, after playback has completed the first time, if you replay the video, fewer ads appear in the replay than during the initial playpack.
- For the VAST and VPAID Ad Plugin, on iOS devices, you can scrub forward and backward while an ad is playing.
- For the FreeWheel Ad Plugin for Player V4, if you seek the scrubber to the end of the video, the midroll ad does not play.
- For the VAST and VPAID Ad Plugin and the bit_Wrapper.min.js plugin, the 'After Replay' mid-roll ad plays at the start of the video rather than its mid-roll start time.
- On iOS Safari, Pulse ads sometimes stop playing due to an ad timeout.
- Ad audio plays over the main content when the ad times out.
- Dash DRM is not working on Windows 8 and Internet Explorer 11.
- For the Bitmovin plugin, after ad playback, the DASH stream moves to the end on Microsoft Edge on Windows 10 and Internet Explorer 11 on Windows 8 and Windows 10.



- For DASH DRM or DASH CENC VOD assets, on Windows 10 (Microsoft Edge and Internet Explorer 11), the player freezes if a user scrubs forward to the end of the video.

2016-08-01 RELEASE - VERSION 4.6.9

New Features in this Player Release

- Setting the initial bitrate level - You can use the new `initialBitrate` parameter to set the initial minimum bitrate level (immediately after video playback) and to sustain that level for a specific period of time. Once the duration is reached, the bitrate level changes to the video plugin's automatic bitrate level. For details, see [Page-level Parameters for Player V4](#) on page 99.
- Closed captions enhancements:
 - Updated navigation - In Player V4, when you click the CC button, the pop-up allows you to easily enable or disable closed captions. Click **LANGUAGE & CAPTIONS** to change language and closed captions settings. For details, see [Closed Captions in Player V4](#) on page 125.
 - Saved settings - Player V4 now saves closed caption settings automatically in the browser. If a user changes a setting (for example, chooses a different font color), the change remains in effect when they open a new browser tab, browser window, or browser session. There is a new Player V4 event (`OO.EVENTS.SAVE_PLAYER_SETTINGS`) that is triggered when a change is saved.
- Thumbnail-based seeking - Player V4 now displays thumbnails to help users search through a video.
 - For desktop and mobile devices, if you drag the scrubber bar, the player pops up a carousel of thumbnails. The player displays as many as will fit on the screen.
 - For desktop devices (browsers in mobile devices do not support this behavior), if you hover your mouse over the scrubber bar, you will see the nearest thumbnail image associated with that location in the video.

This functionality is available for Ooyala-encoded assets that have associated thumbnails. For details, see [Thumbnail-based seeking](#).

- NPAW Integration with Player V4 - A Player V4 plugin is now available for the Ooyala QoS Solution powered by the NicePeopleAtWork (NPAW) YOUBORA platform: <http://smartplugin.youbora.com/v5/javascript/ooyalav4/stable/sp.min.js>.

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS, font, and image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of `skin.json`, download the latest `skin.json` file and merge any changes in this file with your customized `skin.json` file.
- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.6.9/core.min.js>
 - Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.6.9/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.6.9/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.6.9/skin-plugin/html5-skin.min.css>
 - Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.6.9/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.6.9/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.6.9/video-plugin/bit_wrapper.min.js



- Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.6.9/video-plugin/akamaiHD_flash.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.6.9/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.6.9/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.6.9/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.6.9/ad-plugin/freewheel.min.js>
- Analytics Plugins
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.6.9/analytics-plugin/omniture.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- Discovery Plugin: http://player.ooyala.com/static/v4/stable/4.6.9/other-plugin/discovery_api.min.js
- Live SSAI Pulse Plugin: http://player.ooyala.com/static/v4/stable/4.6.9/ad-plugin/ssai_pulse.js

Known Issues

The following are known issues for this release:

- 608/708 captioning is not available with Android HLS in Chrome. Android Chrome does not detect and include the caption in the <video> element `textTrack`.

2016-07-14 RELEASE - VERSION 4.5.7

New Features in this Player Release

- Support for encoding of Akamai live and remote assets for secure tokens.
- Support for cache busting for ad requests when using Server Side Ad Insertion. URLs with the '[CACHEBUSTING]' macro will be replaced with a random string. The default setting for this feature is set to true and will always do cache busting. To disable cache busting you must set the following variable in the player parameters:

```
{
  "ssai-pulse-ads-manager" :
  {
    "cacheBuster":false
  }
}
```

Default Player Links

To update to the latest Player V4 version from a previous version of Player V4:

1. Download and host all of the new player resources (core plugin, [skin plugins](#), [CSS](#), [font](#), and [image resources](#), [video plugins](#), [ad plugins](#), [analytics plugins](#), and so on) and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.5.7/core.min.js>
 - Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.5.7/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.5.7/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.5.7/skin-plugin/html5-skin.min.css>
 - Video Plugins (at least one required)



- Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.5.7/video-plugin/main_html5.min.js
- OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.5.7/video-plugin/osmf_flash.min.js
- Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.5.7/video-plugin/bit_wrapper.min.js
- Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.5.7/video-plugin/akamaiHD_flash.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.5.7/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.5.7/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.5.7/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.5.7/ad-plugin/freewheel.min.js>
- Analytics Plugins
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.5.7/analytics-plugin/omniture.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- Discovery Plugin: http://player.ooyala.com/static/v4/stable/4.5.7/other-plugin/discovery_api.min.js
- Live SSAI Pulse Plugin: http://player.ooyala.com/static/v4/stable/4.5.7/ad-plugin/ssai_pulse.js

Known Issues

The following are known issues for this release:

- Video Plugin Bugs
 - When using the bit_wrapper.min.js video plugin to play DASH assets on Mac devices with the Chrome and Firefox browsers, the video gets stuck after you try and seek in the video. This is due to a bitmovin issue.
- Discovery Plugin Bugs
 - When using the osmf_flash.min.js video plugin on Internet Explorer 11, the video list in the Discovery screen is not properly aligned.
- DRM Bugs
 - When using DRM with Windows 7 and Mac 10.11 devices and Firefox 47, if the end user starts playing the video then drags the browser tab away from the browser, the video will stop playing and get stuck on the loading spinner.
- Ad Plugin Bugs
 - When playing Google IMA VPAID 2.0 mid-roll ads on iPhone 6 devices using iOS 9.0.1 and iPad Air devices using iOS 8.4, the video starts playing from the beginning (00:00) after the mid-roll ad completes.
 - When using the main_html5.min.js video plugin to play VAST ads on Windows 7 devices with Internet Explorer 11, ads do not play as expected.
 - When playing VPAID ads on iPad Air devices using iOS 9.2, forward seek does not work as expected.
 - When using Nexus 5 devices with Android 5.1.1 and Nexus 9 devices using Android 6, VAST ads and video assets intermittently take an unexpectedly long time to load.
 - On iPhone 6 devices using iOS 8.4 and iPhone 6s devices using iOS 9.2 with Google IMA podded ads, if you let the first ad finish playing then tap the Done button before the second ad starts to play, if you tap Play again, the second ad gets skipped.
 - When using Windows 10 with Edge and Chrome browsers with FreeWheel ads, after clicking on the skip button of the first ad, all podded ads are skipped. The expected behavior is that only the first ad will be skipped.



- When using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the ad will show as a black screen on Android with VAST ads.
- On Mac 10.11 with Safari, Chrome, and Firefox with VPAID Google IMA pre-roll ads, the ad gets truncated in the control bar portion of the screen.
- When using multiple VAST mid-roll ads with Windows 7 Firefox and Chrome and iPad devices with iOS 8.4 and Nexus 9 devices with Android 6.0, the second mid-roll ad in the set of mid-roll ads does not play.
- When using the main_html5.min.js video plugin with Google IMA ads on Android mobile web with the Chrome browser, when you replay the video it will play the pre-roll ad but will then show a black screen (main video playback fails).

2016-06-30 RELEASE - VERSION 4.5.5

New Features in this Player Release

- Support for encoding once and supporting multiple DRMs with Player V4 through MPEG-DASH common encryption (CENC) and the Bitmovin video plugin, including:
 - Google Widevine Modular (supported for Firefox and Chrome)
 - Apple FairPlay (supported for Safari)
 - Microsoft PlayReady packaged using Azure (supported for Internet Explorer)
- Support for Akamai HD3 via a new Akamai HD video plugin. This feature is available for customers using Akamai delivery (content packaged by Akamai). See [Adding URLs for an Asset](#) for details on how to support remote assets with Akamai HD. There is no change in functionality for how to support VOD assets with Akamai HD.
- The V4 Ooyala Player is now compliant with FCC 708, which allows end users to easily manipulate the look and feel of closed captions. See [Closed Captions in Player V4](#) on page 125 for details.
 - This functionality is available out of the box and enabled for all customers using Player versions 4.5.5 and above.
 - FCC-compliant closed captions provide end users with more control over how captions are rendered (such as font type, font size, colors, opacity, and so on).
 - FCC-compliant closed captions are available on Vod and Live Streams (encoded with CEA-708 captions).
 - The implementation of closed captions features a responsive design, scaling well with different player sizes and browser types.
- Server Side Ad Functionality through the new SSAI Pulse Player V4 Plugin.

Note:

In the Firefox browser, using the latest bit_wrapper.min.js plugin (version 4.5.5) causes an issue with both video and ad playback for SSAI functionality. This is due to incorrect ad URL formatting by the latest bit_wrapper.min.js plugin.

To avoid this issue on Firefox, use the following version of the bit_wrapper.min.js plugin when loading plugins on your page: http://player.ooyala.com/static/v4/stable/4.4.11/video-plugin/bit_wrapper.js. This plugin works across all devices and browsers for SSAI.

We will continue to work with Bitmovin to have this issue fixed.

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.5.5/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.5.5/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.5.5/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.5.5/skin-plugin/html5-skin.min.css>



- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.5.5/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.5.5/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.5.5/video-plugin/bit_wrapper.min.js
 - Akamai HD video plugin: http://player.ooyala.com/static/v4/stable/4.5.5/video-plugin/akamaiHD_flash.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.5.5/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.5.5/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.5.5/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.5.5/ad-plugin/freewheel.min.js>
- Analytics Plugins
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.4.11/analytics-plugin/omniture.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- Discovery Plugin: http://player.ooyala.com/static/v4/stable/4.4.11/other-plugin/discovery_api.min.js
- Live SSAI Pulse Plugin: http://player.ooyala.com/static/v4/stable/4.4.11/ad-plugin/ssai_pulse.js

Known Issues

The following are known issues for this release:

- Video Plugin Bugs
 - When using the bit_wrapper.min.js video plugin to play DASH assets on Mac devices with the Chrome and Firefox browsers, the video gets stuck after you try and seek in the video. This is due to a bitmovin issue.
 - If you play two assets in a row, thumbnails from the first asset may appear on the second asset if the second asset does not have thumbnails.
- Discovery Plugin Bugs
 - When using the osmf_flash.min.js video plugin on Internet Explorer 11, the video list in the Discovery screen is not properly aligned.
- DRM Bugs
 - When using DRM with Windows 7 and Mac 10.11 devices and Firefox 47, if the end user starts playing the video then drags the browser tab away from the browser, the video will stop playing and get stuck on the loading spinner.
- Ad Plugin Bugs
 - When playing Google IMA VPAID 2.0 mid-roll ads on iPhone 6 devices using iOS 9.0.1 and iPad Air devices using iOS 8.4, the video starts playing from the beginning (00:00) after the mid-roll ad completes.
 - When using the main_html5.min.js video plugin to play VAST ads on Windows 7 devices with Internet Explorer 11, ads do not play as expected.
 - When playing VPAID ads on iPad Air devices using iOS 9.2, forward seek does not work as expected.
 - When using Nexus 5 devices with Android 5.1.1 and Nexus 9 devices using Android 6, VAST ads and video assets intermittently take an unexpectedly long time to load.
 - On iPhone 6 devices using iOS 8.4 and iPhone 6s devices using iOS 9.2 with Google IMA podded ads, if you let the first ad finish playing then tap the Done button before the second ad starts to play, if you tap Play again, the second ad gets skipped.



- When using Windows 10 with Edge and Chrome browsers with FreeWheel ads, after clicking on the skip button of the first ad, all podded ads are skipped. The expected behavior is that only the first ad will be skipped.
- When using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the ad will show as a black screen on Android with VAST ads.
- On Mac 10.11 with Safari, Chrome, and Firefox with VPAID Google IMA pre-roll ads, the ad gets truncated in the control bar portion of the screen.
- When using multiple VAST mid-roll ads with Windows 7 Firefox and Chrome and iPad devices with iOS 8.4 and Nexus 9 devices with Android 6.0, the second mid-roll ad in the set of mid-roll ads does not play.
- When using the main_html5.min.js video plugin with Google IMA ads on Android mobile web with the Chrome browser, when you replay the video it will play the pre-roll ad but will then show a black screen (main video playback fails).

2016-06-09 RELEASE - VERSION 4.4.11

Bugs Fixed in This Release

- This release fixes a bug that occurred when using MP4 video assets with the bit_wrapper.min.js video plugin. The player would only display one closed caption language even if video assets had multiple closed caption language files available. This issue is resolved and closed captions function as expected.
- This release fixes a bug that occurred when using the main_html5.min.js video plugin on an Android device with Google IMA pre-roll ads. On replaying the video, playback would not resume after the pre-roll ad. This issue is resolved and playback functions as expected.
- This release implements an automated way to update the Player version number sent to Omniture for reporting.

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.4.11/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.4.11/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.4.11/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.4.11/skin-plugin/html5-skin.min.css>
- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.4.11/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.4.11/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.4.11/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.4.11/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.4.11/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.4.11/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.4.11/ad-plugin/freewheel.min.js>
- Analytics Plugins
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.4.11/analytics-plugin/omniture.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.



- Discovery: http://player.ooyala.com/static/v4/stable/4.4.11/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- Video Plugin Bugs
 - When using the `osmf_flash.min.js` video plugin on Internet Explorer version 11, the video list in the Discovery screen is not properly aligned.
 - When using the `main_html5.min.js` video plugin on Android Nexus devices with Android versions 4.2.1+, the replay button does not work for eHLS assets.
 - When using Lenovo devices with Android 4.2.2, the replay button intermittently does not work.
- Ad Plugin Bugs
 - On iPhone 6 devices using iOS 8.4 and iPhone 6s devices using iOS 9.2 with Google IMA podded ads, if you let the first ad finish playing then tap the Done button before the second ad starts to play, if you tap Play again, the second ad gets skipped.
 - When using Windows 10 with Edge and Chrome browsers with FreeWheel ads, after clicking on the skip button of the first ad, all podded ads are skipped. The expected behavior is that only the first ad will be skipped.
 - When using VAST mid-roll or post-roll ads on Android, be sure to load the `main_html5.min.js` video plugin along with the `bit_wrapper.min.js` plugin. If the `main_html5.min.js` plugin isn't present, the ad will show as a black screen on Android with VAST ads.
 - On Mac 10.11 with Safari, Chrome, and Firefox with VPAID Google IMA pre-roll ads, the ad gets truncated in the control bar portion of the screen.
 - On Nexus 5 devices using Android M and Nexus 10 devices using Android 5.0.2 with the `bit_wrapper.min.js` video plugin and Google IMA overlay ads, the overlay ad does not display properly (is only partially visible).
 - When using multiple VAST mid-roll ads with Windows 7 Firefox and Chrome and iPad devices with iOS 8.4 and Nexus 9 devices with Android 6.0, the second mid-roll ad in the set of mid-roll ads does not play.
 - When using VAST ads with the `bit_wrapper.min.js` plugin or `osmf_flash.min.js` plugin and FreeWheel, Google IMA, and VPAID ads with the `bit_wrapper.min.js` plugin on Mac and Safari, when the pre-roll ad finishes the video does not play.
 - When using the `main_html5.min.js` video plugin with Google IMA ads on Android mobile web with the Chrome browser, when you replay the video it will play the pre-roll ad but will then show a black screen (main video playback fails).

2016-06-02 RELEASE - VERSION 4.4.8

New Features in this Release

- New third-party analytics plugins for Player V4.
 - The **comScore analytics plugin for Player V4** is developed and maintained by comScore. If you would like to use the comScore plugin with Player V4, please contact your Ooyala account manager for details on how to enable and use the plugin. Any player version starting from 4.3.3 supports the comScore plugin.
 - The **Adobe Analytics (Omniture) analytics plugin for Player V4** is developed and maintained by Ooyala. Any player version starting from 4.4.8 supports the Adobe Analytics (Omniture) plugin. See [Adobe Analytics \(Omniture\) Plugin](#) on page 190 for details.
- Updated default player control bar look and feel, including a completely new set of icons for playback controls and social sharing.
- Ooyala Discovery enhancements, including a more intuitive Discovery algorithm for content recommendation.
- Updated Google IMA plugin.



- The Google IMA plugin now supports a new parameter, `useGoogleAdUI`, that allows you to specify whether or not to use Google's UI for ads. This feature enables you to use the Google ad countdown and learn more button in Google IMA ads. See [Configuring Player V4 Ad Settings with Embedded Parameters](#) for details.
- The Google IMA plugin now supports a new parameter, `vpaidMode`, that allows you to use "insecure" VPAID creatives. See [Configuring Player V4 Ad Settings with Embedded Parameters](#) for details.
- With this release, when using Player V4 on the iPhone mobile web, if you tap Done on the video screen, the pause screen will display. The previous release behavior would show the start screen.
- This release fixes a bug that occurred when using HLS videos on iOS devices using the `main_html5.min.js` video plugin. After seeking past the buffered range, video playback would not continue. This issue is now fixed and seek behavior functions as expected.

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.4.8/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.4.8/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.4.8/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.4.8/skin-plugin/html5-skin.min.css>
- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.4.8/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.4.8/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.4.8/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.4.8/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.4.8/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.4.8/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.4.8/ad-plugin/freewheel.min.js>
- Analytics Plugins
 - Adobe Analytics (Omniture) Plugin: <https://player.ooyala.com/static/v4/stable/4.4.8/analytics-plugin/omniture.js>
 - comScore Analytics Plugin: Please contact your account manager for access to this plugin.
- Discovery: http://player.ooyala.com/static/v4/stable/4.4.8/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- Video Plugin Bugs
 - When using the `osmf_flash.min.js` video plugin on Internet Explorer version 11, the video list in the Discovery screen is not properly aligned.
 - When using the `main_html5.min.js` video plugin on Android Nexus devices with Android versions 4.2.1+, the replay button does not work for eHLS assets.
 - When using Windows 10 with Edge and Chrome and Windows 7 with Chrome, closed captions options do not appear in the closed caption screen.
 - When using Lenovo devices with Android 4.2.2, the replay button intermittently does not work.
- Ad Plugin Bugs



- On iPhone 6 devices using iOS 8.4 and iPhone 6s devices using iOS 9.2 with Google IMA podded ads, if you let the first ad finish playing then tap the Done button before the second ad starts to play, if you tap Play again, the second ad gets skipped.
- When using Windows 10 with Edge and Chrome browsers with FreeWheel ads, after clicking on the skip button of the first ad, all podded ads are skipped. The expected behavior is that only the first ad will be skipped.
- When using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the ad will show as a black screen on Android with VAST ads.
- On Mac 10.11 with Safari, Chrome, and Firefox with VPAID Google IMA pre-roll ads, the ad gets truncated in the control bar portion of the screen.
- On Nexus 5 devices using Android M and Nexus 10 devices using Android 5.0.2 with the bit_wrapper.min.js video plugin and Google IMA overlay ads, the overlay ad does not display properly (is only partially visible).
- When using multiple VAST mid-roll ads with Windows 7 Firefox and Chrome and iPad devices with iOS 8.4 and Nexus 9 devices with Android 6.0, the second mid-roll ad in the set of mid-roll ads does not play.
- When using VAST ads with the bit_wrapper.min.js plugin or osmf_flash.min.js plugin and FreeWheel, Google IMA, and VPAID ads with the bit_wrapper.min.js plugin on Mac and Safari, when the pre-roll ad finishes the video does not play.
- When using the main_html5.min.js video plugin with Google IMA ads on Android mobile web with the Chrome browser, when you replay the video it will play the pre-roll ad but will then show a black screen (main video playback fails).

2016-04-28 RELEASE - VERSION 4.3.3

New Features in this Release

- New Live DVR feature.
 - This feature allows a user, during live streaming (HLS streams), to drag the slider backwards in time and to drag all the way to the right to catch up to present time.
 - This feature requires the bit_wrapper.min.js video plugin. Flash must be enabled in the browser.
 - This feature is supported on the following desktop browsers: Chrome, Firefox, Safari for OS/X, Internet Explorer, and Microsoft Edge.
- New Ooyala Pulse Ad Manager plugin.
 - This plugin allows you to integrate with Ooyala Pulse, an ad server and ad management platform, where publishers are able to take full control over their ad inventory, use targeting based on audience and video content, and raise effective CPM and sell-out rates with realtime simulations, dynamic ad delivery and alternative ad sources.
- Support for HLS on Android web.
- Support for DVR and bitrate selection for HLS playback on Safari.
- Closed Captions rendered by the player skin.
- Player skin updates.
 - The player skin CSS class names now have an "oo-" prefix and use dash format.
 - The player skin value for the minimum width of the player now is restricted to a maximum of 320px.
- VAST Ad Manager plugin updates.
 - The VAST ad manager has been updated to not allow pre-loading of ads (sending the request for XML before the video play begins) to make ad loading universal across ad managers and to ensure that ad impression numbers are accurate.
- ID3 detection.
 - The Ooyala Player V4 will recognize ID3 tag information embedded in an HLS stream.



- When an HLS stream is embedded with ID3, the Ooyala Player will automatically raise an event VC_TAG_FOUND with the value videoTagFound to indicate that the ID3 cues have been encountered.
- With this capability, the Ooyala Player ensures that ID3 cues are easily detectable by web applications for a variety of use cases.

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.3.3/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.3.3/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.3.3/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.3.3/skin-plugin/html5-skin.min.css>
- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.3.3/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.3.3/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.3.3/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - Ooyala Pulse Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.3.3/ad-plugin/pulse.min.js>
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.3.3/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.3.3/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.3.3/ad-plugin/freewheel.min.js>
 - Discovery: http://player.ooyala.com/static/v4/stable/4.3.3/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- Ad Plugin Bugs
 - On iPhone 6 devices using iOS 8.4 and iPhone 6s devices using iOS 9.2 with Google IMA podded ads, if you let the first ad finish playing then tap the Done button before the second ad starts to play, if you tap Play again, the second ad gets skipped.
 - When using the OSMF Flash plugin with FreeWheel ads, when you scrub to the end of a video after a pre-roll ad ends, the post-roll ad plays, but the player freezes.
 - When using VAST ads with Nexus 5 devices using Android 4.4.2, if you seek in the video to an ad, the ad timer shows NaN:NaN until the ad starts playing.
 - If you only specify the bit_wrapper.min.js video plugin when using VAST mid-roll or post-roll ads with Android, the ad will show as a black screen until the user clicks pause then play. We have filed a ticket with Bitmovin to fix this issue. The workaround is, when using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the bug will occur on Android with VAST ads.
 - The ad marquee and ad control bar do not appear for Google IMA and VPAID 2.0 ads to avoid blocking ad interactivity. This was implemented intentionally.
 - When using the bit_wrapper.min.js plugin with Google IMA ads and Ooyala-transcoded DASH assets, post-roll ads do not automatically run when playback of the video concludes. The post-roll plays only if you seek to the end of the video.
 - With the iPhone 6 using iOS 8.4 and iPhone 6s using iOS 9.2 to play a video with a set of Google IMA podded ads, the second ad gets skipped if the end user clicks "done" before the second ad starts to play.



- When using the bit_wrapper.min.js plugin with FreeWheel post-roll ads on Android devices and Mac devices with Chrome, the post-roll ad appears as a black screen and blocks the ad screen.
- Live DVR Bugs
 - On Internet Explorer, if a live DVR HLS stream is paused for longer than the available DVR buffer (if the DVR window passes your current window while you are paused), the stream will not be able to unpause.

2016-04-12 RELEASE - VERSION 4.2.14

This release of Player V4 includes fixes for the following bugs:

- On Mac devices using Safari, the Discovery tray was misaligned. This issue is resolved.
- When using the Safari browser, the Up Next title text was overlapping the description. This issue is resolved.
- On iOS devices using Google IMA ads on a Player enabled for Discovery, after starting a Discovery video, Google IMA ads only played with audio (no video). This issue has been resolved for iPad devices. Discovery is not supported on iPhone devices.
- On iPad devices, when multiple V4 players were embedded on the same page, only one player would play successfully (additional players would not play even after you paused the first video). This issue has been resolved so that you can have multiple players on iPad devices, play the first video, pause the first video (or watch to the end) and successfully play videos from another player afterward. Please note that there is a known limitation that iOS that only allows a single media to play at a given time (you cannot play multiple videos at the same time).

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.2.14/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.2.14/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.2.14/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.2.14/skin-plugin/html5-skin.min.css>
- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.2.14/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.2.14/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.2.14/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.2.14/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.2.14/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.2.14/ad-plugin/freewheel.min.js>
- Discovery: http://player.ooyala.com/static/v4/stable/4.2.14/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- If there are multiple players on the same page, if you play the video from the first player, play the video from the second player, go to fullscreen mode in the second player, go to normal screen mode in the second player, and then go to fullscreen mode in the first player, the second player will overlap the first player (the view of the first player is distorted).
- Bitmovin (bit_wrapper.min.js) Video Plugin Bugs



- When using the bit_wrapper.min.js plugin with DASH video assets in the OSX Chrome environment, Google IMA post-roll ads are skipped and FreeWheel post-roll ads only play the ad audio (no video).
- eHLS assets do not playback completely when played with the bit_wrapper plugin in Chrome. We are working on upgrading and testing the latest version of the Bitmovin SDK to fix this issue for the next player release.
- When using the bit_wrapper.min.js video plugin with DASH videos with Samsung Galaxy S2 - Android 4.1.2, Samsung S2 Android 4.1.2, and Samsung S3 Android 4.3, the main video never plays.
- When using the bit_wrapper.min.js video plugin, VAST 2.0 and 3.0 m3u8 ads do not play with Nexus 5 devices using Android 4.4.2 and Mac 10.11 with Chrome 49 and Firefox 45.
- When using the bit_wrapper.min.js plugin with VPAID 2.0, a black screen appears instead of an ad. The workaround for this bug is to load the main_html5.min.js video plugin before the bit_wrapper.min.js plugin when using VPAID 2.0 ads on Safari.
- When using the bit_wrapper.min.js plugin with Google IMA ads and Ooyala-transcoded DASH assets, post-roll ads do not automatically run when playback of the video concludes. The post-roll plays only if you seek to the end of the video.
- If you only specify the bit_wrapper.min.js video plugin when using VAST mid-roll or post-roll ads with Android, the ad will show as a black screen until the user clicks pause then play. We have filed a ticket with Bitmovin to fix this issue. The workaround is, when using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the bug will occur on Android with VAST ads.
- The bit_wrapper.min.js plugin does not currently support Live closed captions for HLS.
- You cannot use the bit_wrapper.min.js plugin with iOS. You must use main_html5.min.js instead. Please note that the bit_wrapper.min.js plugin won't cause any problems or errors to other plugins if you include it, it just won't function at all with iOS.
- The bit_wrapper.min.js SDK can only play a DASH video that has audio codecs AAC or mp4a.40.2. The bit_wrapper.min.js SDK is not playable with AC3. We have raised an issue with Bitmovin for this issue. If you want to ingest DASH through Ooyala, please set your audio codec to AAC or mp4a.40.2.
- When using Ooyala-transcoded DASH streams with Safari, Chrome, and Firefox, the main video does not play, and a pre-roll with a black screen plays instead.
- Main (main_html5.min.js) Video Plugin Bugs
 - When using the main_html5.min.js plugin and HLS video on Microsoft Edge, the video sometimes does not respond to seek events.
- Ad Plugin Bugs
 - Overlay ads do not work as expected (they have sizing and positioning issues) when the end user replays the video. We are working with FreeWheel on FreeWheel-related issues and are debugging other ad manager issues.
 - If there are more than one VPAID ads in a VAST 2.0 ad pod, then the ads will not be played. This happens only in the VAST and VPAID ad plugin.
 - On Nexus 5 devices using Android 4.4.2, FreeWheel ads intermittently get skipped.
 - On iOS devices, if an IMA VPAID ad is playing and you tap the Done button, the ad does not resume playing when you tap the Play button.
 - On the Firefox browser, using the Pulse ad plugin, when resizing the screen, the dimensions do not properly occupy the space of the player.
- iOS 8 devices have a limitation on very large MP4 bitrate (bitrate 4400) or file sizes (the device itself is not able to play). This is an Apple issue, please refer to Apple documentation for full details on this limitation.
- On Nexus 5 devices using Android 6.0 and Windows 7 with Firefox 37.0, the closed caption text overlaps with the scrubber bar.



- Discovery UpNext videos do not automatically autoplay after the Next Video countdown ends. We are working on fixing this issue. For this release, a workaround is to add this event listener to player param:

```

var assetCount = 0;

onCreate: function(player) {
    player.mb.subscribe('*', 'test', function(event) {
        if (event.match(OO.EVENTS.EMBED_CODE_CHANGED)) {
            assetCount++;
        }
        if (event.match(OO.EVENTS.PLAYBACK_READY)) {
            if (assetCount > 1) {
                player.mb.publish(OO.EVENTS.INITIAL_PLAY);
            }
        }
    });
}

```

2016-04-01 RELEASE - VERSION 4.2.9

This release fixes a bug where the autoplay functionality caused pre-roll ads to malfunction (the pre-roll ad was not playing as expected).

Default Player Links

- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.2.9/core.min.js>
- Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.2.9/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.2.9/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.2.9/skin-plugin/html5-skin.min.css>
- Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.2.9/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.2.9/video-plugin/osmf_flash.min.js
 - Bitmovin plugin for HLS and DASH: http://player.ooyala.com/static/v4/stable/4.2.9/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - VAST and VPAID Ad Plugin: https://player.ooyala.com/static/v4/stable/4.2.9/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.ooyala.com/static/v4/stable/4.2.9/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.ooyala.com/static/v4/stable/4.2.9/ad-plugin/freewheel.min.js>
- Discovery: http://player.ooyala.com/static/v4/stable/4.2.9/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- Bitmovin (bit_wrapper.min.js) Plugin Bugs
 - eHLS assets do not playback completely when played with the bit_wrapper plugin in Chrome. We are working on upgrading and testing the latest version of the Bitmovin SDK to fix this issue for the next player release.
 - When using the bit_wrapper.min.js video plugin with DASH videos with Samsung Galaxy S 2 - Android 4.1.2, Samsung S2 Android 4.1.2, and Samsung S3 Android 4.3, the main video never plays.



- When using the bit_wrapper.min.js video plugin, VAST 2.0 and 3.0 m3u8 ads do not play with Nexus 5 devices using Android 4.4.2 and Mac 10.11 with Chrome 49 and Firefox 45.
- When using the bit_wrapper.min.js plugin with VPAID 2.0, a black screen appears instead of an ad. The workaround for this bug is to load the main_html5.min.js video plugin before the bit_wrapper.min.js plugin when using VPAID 2.0 ads on Safari.
- When using the bit_wrapper.min.js plugin with Google IMA ads and Ooyala-transcoded DASH assets, post-roll ads do not automatically run when playback of the video concludes. The post-roll plays only if you seek to the end of the video.
- If you only specify the bit_wrapper.min.js video plugin when using VAST mid-roll or post-roll ads with Android, the ad will show as a black screen until the user clicks pause then play. We have filed a ticket with Bitmovin to fix this issue. The workaround is, when using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the bug will occur on Android with VAST ads.
- The bit_wrapper.min.js plugin does not currently support Live closed captions for HLS.
- You cannot use the bit_wrapper.min.js plugin with iOS. You must use main_html5.min.js instead. Please note that the bit_wrapper.min.js plugin won't cause any problems or errors to other plugins if you include it, it just won't function at all with iOS.
- The bit_wrapper.min.js SDK can only play a DASH video that has audio codecs AAC or mp4a.40.2. The bit_wrapper.min.js SDK is not playable with AC3. We have raised an issue with Bitmovin for this issue. If you want to ingest DASH through Ooyala, please set your audio codec to AAC or mp4a.40.2.
- When using Ooyala-transcoded DASH streams with Safari, Chrome, and Firefox, the main video does not play, and a pre-roll with a black screen plays instead.
- Main (main_html5.min.js) Plugin Bugs
 - When playing HLS assets on Safari with the main_html5.min.js video plugin, on Safari browsers, after you replay the video the main video does not respond to seeking.
 - When using the main_html5.min.js plugin and HLS video on Microsoft Edge, the video sometimes does not respond to seek events.
- Ad Plugin Bugs
 - Overlay ads do not work as expected (they have sizing and positioning issues) when the end user replays the video. We are working with FreeWheel on FreeWheel-related issues and are debugging other ad manager issues.
 - If there are more than one VPAID ads in a VAST 2.0 ad pod, then the ads will not be played. This happens only in the VAST and VPAID ad plugin.
 - On Nexus 5 devices using Android 4.4.2, FreeWheel ads intermittently get skipped.
 - On iOS devices, if an IMA VPAID ad is playing and you tap the Done button, the ad does not resume playing when you tap the Play button.
 - On the Firefox browser, using the Pulse ad plugin, when resizing the screen, the dimensions do not properly occupy the space of the player.
 - iOS 8 devices have a limitation on very large MP4 bitrate (bitrate 4400) or file sizes (the device itself is not able to play). This is an Apple issue, please refer to Apple documentation for full details on this limitation.
 - On Nexus 5 devices using Android 6.0 and Windows 7 with Firefox 37.0, the closed caption text overlaps with the scrubber bar.
 - Discovery UpNext videos do not automatically autoplay after the Next Video countdown ends. We are working on fixing this issue. For this release, a workaround is to add this event listener to player param:

```

var assetCount = 0;

onCreate: function(player) {
    player.mb.subscribe('*', 'test', function(event) {
        if (event.match(OO.EVENTS.EMBED_CODE_CHANGED)) {
            assetCount++;
        }
    })
}

```



```

        if (event.match(OO.EVENTS.PLAYBACK_READY)) {
            if (assetCount > 1) {
                player.mb.publish(OO.EVENTS.INITIAL_PLAY);
            }
        });
    }
}

```

2016-03-28 RELEASE - VERSION 4.2.7

New Features in this Release

- New responsive design functionality (by default the player will resize dynamically when the size of the container of the player is changed). For details on how to override this default behavior, see [Player V4 FAQ](#)
- New bitrate selection feature.
 - Adaptive Bitrate (ABR) is only available with HLS in Safari (native ABR supported and managed by Apple) and Microsoft Edge.
 - Bitrate selection is only available with HDS, HLS, and DASH encodings.
- New Bitmovin [video plugin](#) to support DASH and HLS playback, specifically:
 - Clear DASH on all desktop browsers.
 - Clear DASH on Android mobile web (Android 4.1+).
- Note:** The bit_wrapper.js can only play DASH video that uses the audio codecs AAC and mp4a.40.2. The bit_wrapper.js is not playable with AC3 (Dolby Digital). If you want to ingest a DASH video with Ooyala, please use AAC or mp4a.40.2 audio codecs for this release.
- Updates to existing Ad support, including:
 - New support for VPAID 2.0 to allow monetization on mobile devices using the HTML5 player. See [VAST and VPAID Ad Plugin](#) on page 171 for details.
 - A new Analytics Framework to allow third-party analytics providers to integrate with the Ooyala Player. See [Analytics Framework](#) on page 210 for details.
 - Enhancements to the Google IMA and DFP plugin including support for VPAID 1.0 and 2.0. See [Google IMA Ad Plugin](#) on page 147 for details.
 - Enhanced VAST 3.0 support, including support for VMAP. See [VAST and VPAID Ad Plugin](#) on page 171 for details.

Default Player Links

To update to the latest Player V4 version from a previous version:

1. Download and host all of the new plugins and update your resources to point to these new files.
 2. If you've customized your own Player skin using your own version of skin.json, download the latest skin.json file and merge any changes in this file with your customized skin.json file.
- Core Player (required): <http://player.ooyala.com/static/v4/stable/4.2.7/core.min.js>
 - Player Skin:
 - <http://player.ooyala.com/static/v4/stable/4.2.7/skin-plugin/html5-skin.min.js>
 - <http://player.ooyala.com/static/v4/stable/4.2.7/skin-plugin/skin.json>
 - <http://player.ooyala.com/static/v4/stable/4.2.7/skin-plugin/html5-skin.min.css>
 - Video Plugins (at least one required)
 - Main video plugin for HLS and MP4: http://player.ooyala.com/static/v4/stable/4.2.7/video-plugin/main_html5.min.js
 - OSMF Flash plugin for HDS: http://player.ooyala.com/static/v4/stable/4.2.7/video-plugin/osmf_flash.min.js



- Bitmovin plugin for HLS and DASH: http://player.oyala.com/static/v4/stable/4.2.7/video-plugin/bit_wrapper.min.js
- Ad Plugins
 - VAST and VPAID Ad Plugin: https://player.oyala.com/static/v4/stable/4.2.7/ad-plugin/ad_manager_vast.min.js
 - Google IMA Ad Plugin: https://player.oyala.com/static/v4/stable/4.2.7/ad-plugin/google_ima.min.js
 - FreeWheel Ad Plugin: <https://player.oyala.com/static/v4/stable/4.2.7/ad-plugin/freewheel.min.js>
- Discovery: http://player.oyala.com/static/v4/stable/4.2.7/other-plugin/discovery_api.min.js

Known Issues

The following are known issues for this release:

- Bitmovin (bit_wrapper.min.js) Plugin Bugs
 - When using the bit_wrapper.min.js video plugin with DASH videos with Samsung Galaxy S2 - Android 4.1.2, Samsung S2 Android 4.1.2, and Samsung S3 Android 4.3, the main video never plays.
 - When using the bit_wrapper.min.js video plugin, VAST 2.0 and 3.0 m3u8 ads do not play with Nexus 5 devices using Android 4.4.2 and Mac 10.11 with Chrome 49 and Firefox 45.
 - When using the bit_wrapper.min.js plugin with VPAID 2.0, a black screen appears instead of an ad. The workaround for this bug is to load the main_html5.min.js video plugin before the bit_wrapper.min.js plugin when using VPAID 2.0 ads on Safari.
 - When using the bit_wrapper.min.js plugin with Google IMA ads and Ooyala-transcoded DASH assets, post-roll ads do not automatically run when playback of the video concludes. The post-roll plays only if you seek to the end of the video.
 - If you only specify the bit_wrapper.min.js video plugin when using VAST mid-roll or post-roll ads with Android, the ad will show as a black screen until the user clicks pause then play. We have filed a ticket with Bitmovin to fix this issue. The workaround is, when using VAST mid-roll or post-roll ads on Android, be sure to load the main_html5.min.js video plugin along with the bit_wrapper.min.js plugin. If the main_html5.min.js plugin isn't present, the bug will occur on Android with VAST ads.
 - The bit_wrapper.min.js plugin does not currently support Live closed captions for HLS.
 - You cannot use the bit_wrapper.min.js plugin with iOS. You must use main_html5.min.js instead. Please note that the bit_wrapper.min.js plugin won't cause any problems or errors to other plugins if you include it, it just won't function at all with iOS.
 - The bit_wrapper.min.js SDK can only play a DASH video that has audio codecs AAC or mp4a.40.2. The bit_wrapper.min.js SDK is not playable with AC3. We have raised an issue with Bitmovin for this issue. If you want to ingest DASH through Ooyala, please set your audio codec to AAC or mp4a.40.2.
 - When using the bit_wrapper.min.js plugin with HLS FreeWheel ads, when you replay the ad, post-roll ads intermittently show a black screen and do not transition to the end screen.
 - When using Ooyala-transcoded DASH streams with Safari, Chrome, and Firefox, the main video does not play, and a pre-roll with a black screen plays instead.
- Main (main_html5.min.js) Plugin Bugs
 - When playing HLS assets on Safari with the main_html5.min.js video plugin, on Safari browsers, after you replay the video the main video does not respond to seeking.
 - When using the main_html5.min.js plugin and HLS video on Microsoft Edge, the video sometimes does not respond to seek events.
- Ad Plugin Bugs
 - Overlay ads do not work as expected (they have sizing and positioning issues) when the end user replays the video. We are working with FreeWheel on FreeWheel-related issues and are debugging other ad manager issues.
 - If there are more than one VPAID ads in a VAST 2.0 ad pod, then the ads will not be played. This happens only in the VAST and VPAID ad plugin.



- On Nexus 5 devices using Android 4.4.2, FreeWheel ads intermittently get skipped.
- On iOS devices, if an IMA VPAID ad is playing and you tap the Done button, the ad does not resume playing when you tap the Play button.
- On the Firefox browser, using the Pulse ad plugin, when resizing the screen, the dimensions do not properly occupy the space of the player.
- iOS 8 devices have a limitation on very large MP4 bitrate (bitrate 4400) or file sizes (the device itself is not able to play). This is an Apple issue, please refer to Apple documentation for full details on this limitation.
- On Nexus 5 devices using Android 6.0 and Windows 7 with Firefox 37.0, the closed caption text overlaps with the scrubber bar.
- Discovery UpNext videos do not automatically autoplay after the Next Video countdown ends. We are working on fixing this issue. For this release, a workaround is to add this event listener to player param:

```

var assetCount = 0;

onCreate: function(player) {
    player.mb.subscribe('*', 'test', function(event) {
        if (event.match(OO.EVENTS.EMBED_CODE_CHANGED)) {
            assetCount++;
        }
        if (event.match(OO.EVENTS.PLAYBACK_READY)) {
            if (assetCount > 1) {
                player.mb.publish(OO.EVENTS.INITIAL_PLAY);
            }
        }
    });
}

```

2016-02-24 RELEASE - VERSION 4.1.4

New Features in this Release

- New basic closed caption functionality.
- Updates to existing Ad support, including:
 - New support for linear, podded ads, parsing, and error tracking with VAST 3.0, along with existing support for VAST 2.0.
 - Enhancements to the FreeWheel plugin.
 - Enhancements to the Google IMA plugin.

Known Issues

The following are known issues for this release:

- For videos with an aspect ratio where the height of the video is greater than the width, we recommend that you use our default responsive design setting (aspectRatio:auto). If you need to use a container with a fixed width, we recommend that you set the aspectRatio to a value that matches the aspect ratio of the video. For this release, for fixed width containers, if you set an aspectRatio that does not match the aspect ratio of the video, for HTML5 videos you may see pillar boxing or letter boxing, and for Flash HDS videos you may see the video stretch or shrink to get forcefully fit into the fixed frame.
- For Flash HDS videos with an aspect ratio where the width is much greater than the height (short and wide videos), we recommend that you use our default responsive design setting (aspectRatio:auto) or set the aspectRatio to a value that matches the aspect ratio of the video. If you set an aspectRatio that does not match the aspect ratio of the video, with a fixed width and height, your closed captions may not appear on the video.



- If you are using Google IMA on mobile devices, you need to disable the Ad Marquee, as it is currently blocking the learn more button that allows clickthrough ads. Video clickthroughs are disabled on mobile by Google IMA's design. We will implement a more permanent solution in a future release.
- On Nexus 5 devices using Android 6.0 and Windows 7 with Firefox 37.0, the closed caption text overlaps with the scrubber bar.
- For Google IMA ads on Windows 7 and Mac 10.10 devices using Firefox and Chrome, the end user is unable to click the Back button on the gallery overlay in fullscreen mode.
- For Google IMA overlay ads on Windows 10, Mac 10.11, iPhone 6 and iPhone 6+, the overlay ad does not display properly.
- FreeWheel overlay ads do not work with Flash video playback.

2016-01-29 RELEASE - VERSION 4.0.7

New Features in this Release

- New Ooyala Video Tech Controller (VTC) plugins: main_html5.js and osmf_flash.js that allow:
 - HTML5 to play HLS and MP4.
 - HTML5 to play AES HLS.
 - HTML5 with Flash playback capabilities to play HDS and Flash videos.
- Updates to the existing ads support, including:
 - FreeWheel.
 - VAST.
 - Google IMA and DFP.
- New support for Ooyala IQ Analytics.
- New content-aware controls for VoD, Live, and ads.

Known Issues

- The following is an iOS bug that Ooyala does not have control over. When you seek around quickly on iOS, sometimes you can get the video player stuck in a paused state and the play button does not appear for you to unpause the video.
- On iOS devices, if you play a video with more than one ad and tap the done button more than one time during the playback session to close the ads, the player can get stuck on the spinner screen and the video will not load.
- On the iPad Air and iPhones, if you replay a video, the mid-roll ads and overlay ad that should get played during the mid-roll ad slot incorrectly get played at the end of the video.
- Video content is not centered when playing on the iPad device. This issue is caused by an underlying browser (iOS Safari) issue where the browser is not maintaining the correct video position.

2015-10-30 RELEASE

New Features in this Release

- **Ooyala HTML5 Player V4** (or simply Player V4) — Web-based player that runs on popular browsers that features a new architecture and new player skin.
- **New Ooyala Player Skin for iOS** — Native app that runs on iOS 7+.
- **Ooyala HTML5 Player V4** (or simply Player V4) — Web-based player that runs on popular browsers that features a new architecture and new player skin.
- **New Ooyala Player Skin for iOS** — Native app that runs on iOS 7+.
- Unified look and feel across platforms (HTML5 and iOS for this release).
- Highly *configurable user interface* for a customized playback experience — see *Customizing the Player V4 Appearance* on page 116.
- For a list of supported functionality, see *Player V4* on page 6.



Known Issues

- You can use Player V4 only with V4 routes, and Player V3 only with V3 routes. If you attempt to use a Player V3 with a V4 route, or a Player V4 with a non-V4 route, you will receive an error message in the console log and your player will not work.
- You can load an html5-skin only with Player V4. If you try to load an html5-skin onto an earlier Ooyala player version, you will receive an error message in the console log.
- There are no volume controls shown for iOS devices. This is due to a limitation set by Apple where the volume property is not settable in JavaScript because the audio level is physically controlled by the user with iOS devices. See for more details.
- Ad playback controls do not work on the iPhone in all modes, or on the iPad in fullscreen mode. The controls that do not work for iPhone (all modes) and iPad (fullscreen mode) include:
 - Ad marquee
 - Learn more button
 - Skip button
 - Ad clickthrough
- Certain UI elements may appear distorted on IE 10.
- When you click the share button on the scrubber bar, switch to fullscreen mode, and click the Email subtab, the email tab does not display properly on the Android screen.
- FreeWheel overlay ads do not align properly when you switch to fullscreen view on an HTML5 player.
- When you switch to fullscreen mode, click the more options button, click the share button, click the Email subtab, click in a text field, and hit the spacebar, video controls are lost after the keyboard pops up for Android devices.
- On iOS devices, during playback, if you click on the scrubber bar to move the video forward, it advances directly to the end regardless of where you clicked.
- The play/pause button state does not appear after resuming ad playback from the paused state for iPad Air, Nexus 5 (Android 5.0 and 4.4.2), Mac 10.10 for Firefox and Safari, and Mac 10.9 for Chrome, Firefox, and Safari.
- With FreeWheel, the player UI covers portions of the screen, causing the user from being unable to click on the overlay to open a clickthrough url. The close overlay button will not display on a FreeWheel overlay because FreeWheel renders all overlays and they don't provide a close button. The FreeWheel API does not allow our player to control this functionality directly. We are working to resolve this issue in a future release.
- For podded video ads, it's a known issue with the FreeWheel SDK that if a user pauses any of the video ads after the first one, it will skip that ad. FreeWheel will fix this bug.
- For FreeWheel ads, if you have a podded ad, there is no way to only skip a single ad in the pod, so we disable skip ad button for all FreeWheel podded ads even if the user sets it to display. This is a FreeWheel SDK issue that FreeWheel has no plans to fix.
- Video playback gets stuck if the end user unplugs their headphones during playback. This bug is due to underlying iOS default behavior.

