

# **EZDRM Configuration**

## Testing Playback

## Table of Contents

<b>EZDRM Universal DRM</b>	<b>1</b>
<b>Testing Playback of Encrypted Content - Universal DRM</b>	<b>3</b>
Overview	3
Widevine over DASH (EZDRM Universal)	4
<i>Widevine Proxy URL</i>	4
<i>PX Value</i>	4
<i>Widevine Proxy URL for AWS, Anevia and Nimble Streamer</i>	6
<i>Widevine Auth URL</i>	6
Custom Data (Optional)	7
Sample Player for Widevine over DASH	7
PlayReady over DASH (EZDRM Universal)	12
<i>Proxy URL</i>	12
<i>PX Value</i>	12
<i>PlayReady Auth URL</i>	13
Custom Data (Optional)	14
Sample Player for PlayReady over DASH	14
Sample Player for PlayReady over Microsoft Smooth Streaming	19
Apple FairPlay Streaming over HLS	20
<i>EZDRM FairPlay Account</i>	20
<i>Passing Custom Data</i>	21
<i>Sample Players for Apple FairPlay Streaming</i>	22

# Testing Playback of Encrypted Content - Universal DRM

## Overview

To test the playback of encrypted content, use the Manifest URL that was created during the encryption process.

<https://yourmanifestURL/manifest.mpd>

To test playback, you will also need your Proxy URL from the DRM generated keys, which will be defined in more detail in the following sections.

## Widevine over DASH (EZDRM Universal)

### *Widevine Proxy URL*

The Widevine Proxy URL has two parts, the first is the base URL with your account PX value.

The base URL is: <https://widevine-dash.ezdrm.com/proxy?px=XXXXXX>


The px value is the last 6 digits highlighted in yellow above.

### *PX Value*

The **pX value** is branded for your account and is account specific. Your PX value is always the same no matter what packager you are using.

It is the last six characters of your EZDRM Profile ID, as shown in this example:

1. Login to your EZDRM.com account
2. Under “My Account” click on the link to **My Widevine Account** under **Widevine DRM**.



[DRM](#)
[DEMOS](#)
[RESOURCES](#)
[TRY FREE](#)
[COMPANY](#)
[CONTACT](#)

MY ACCOUNT

FAIRPLAY DRM

PLAYREADY DRM

WIDEVINE DRM

My Account

Username: [redacted].com  
To change / update your email address, please contact us via email at [CustomerService@ezdrm.com](mailto:CustomerService@ezdrm.com)


Password: xxxxxxxx [Update](#)

My Services: Apple FairPlay DRM: Professional  
Microsoft PlayReady DRM: Universal  
Google Widevine DRM: Universal  
Support: None Chosen [Update](#)

Account Info: NA NA  
NA NA  
Yonkers, NY  
United States of America 10701  
555-555-5555 [Update](#)

Billing Info: Billing Account Number:  
Visa ending with 123  
Expires: 01 2025  
Name on card: NA [Update](#)

3. The last 6 digits of your Widevine Profile ID is your PX Number.



[DRM](#)
[DEMOS](#)
[RESOURCES](#)
[TRY FREE](#)
[COMPANY](#)
[CONTACT](#)

MY ACCOUNT

FAIRPLAY DRM

PLAYREADY DRM

My Widevine Account Settings

Your Widevine Profile ID is 8XXXXXX0-4XXC-4XXC-AAXX-387XXX**BOXBX5**

Your authorization url is <http://wm.ezdrm.com/PlayValue.asp>  
To learn more about what an authorization url is, as well as to update it, please go [here](#).

For help with setting up, please take a look at our [setup documentation](#).

## Widevine Proxy URL for AWS, Anevia and Nimble Streamer

The Widevine authentication URL is the same for all packagers except for AWS, Anevia and Nimble Streamer. They have a different license proxy URL, but the PX value is always the same value assigned to your account.

The same last 6 digits of your account will be added as the px value. The base URL is as follows:

<https://widevine-dash.ezdrm.com/widevine-php/widevine-foreignkey.php?pX=XXXXXX>

## Widevine Auth URL

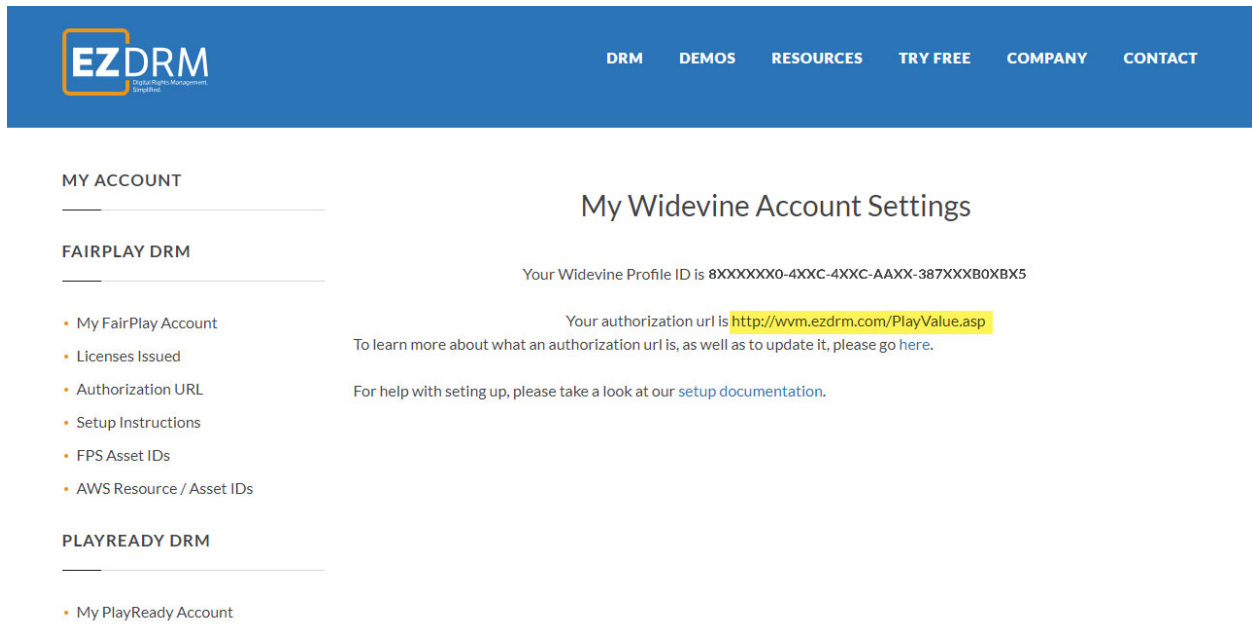
The EZDRM system passes several values by default to your Authorization URL:

- **Client\_ip** of the end client
- **pX** is your EZDRM specific PX value for your account
- **Custom Data** is added to the end of the string (*this will be described in more detail in the next section*)

Here is an example of the EZDRM return post to your Authorization URL:

[https://AUTHURL/PlayValue.asp?p1=1&response={"status":"OK","supported\\_tracks":\["\[\]","internal\\_status":127,"client\\_info":\["\[\]","done"\],"client\\_ip":"46.200.40.005"}&pX=XXXXXX&CustomData=123](https://AUTHURL/PlayValue.asp?p1=1&response={)

You can find and edit your authorization URL by logging into your EZDRM account, under “My Account” click on the link to **My Widevine Account** under **Widevine DRM**.



The screenshot shows the EZDRM website with a blue header containing the EZDRM logo and navigation links: DRM, DEMOS, RESOURCES, TRY FREE, COMPANY, and CONTACT. The main content area is titled "My Widevine Account Settings". On the left, there is a sidebar with sections: "MY ACCOUNT", "FAIRPLAY DRM" (containing links for My FairPlay Account, Licenses Issued, Authorization URL, Setup Instructions, FPS Asset IDs, and AWS Resource / Asset IDs), and "PLAYREADY DRM" (containing My PlayReady Account). The main content area displays: "Your Widevine Profile ID is 8XXXXXX0-4XXC-4XXC-AAXX-387XXXB0XBX5", "Your authorization url is <http://wvm.ezdrm.com/PlayValue.asp>", "To learn more about what an authorization url is, as well as to update it, please go [here](#).", and "For help with setting up, please take a look at our [setup documentation](#)."

## Custom Data (Optional)

This additional configuration to your Widevine Proxy URL will pass additional business logic for license validation.

As part of the licensing process, the content owner can pass a set of parameters through the EZDRM system in order to use that information within the business logic. These parameters should be attached to the server URL. The server URL will include your PX value and all of the other values should follow:

<https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX&CustomData=123>

## Sample Player for Widevine over DASH

An example playback player for Widevine Universal content is the Shaka Player found here: <https://shaka-player-demo.appspot.com/demo/>

Widevine uses Chrome and Firefox web browsers.

*Note: all calls must be over https.*

1. Select the **(custom asset)** option from the **Asset** dropdown:

## Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:  
Custom manifest:  
Custom license server:  
Custom license certificate URL:  

Load
Un

Angel One (multicodec, multilingual)

Live sim (6s segments)  
Live sim (multi-period)  
**Wowza**  
Big Buck Bunny (Live)  
**Bitcodin**  
Art of Motion (DASH)  
Art of Motion (HLS, TS)  
Sintel (HLS, TS, 4k)  
**Azure Media Services**  
Azure Trailer  
Tears Of Steel (external text)  
**GPAC**  
live profile with five periods  
main profile, single file  
onDemand profile  
**Verizon Digital Media Services**  
Multi DRM - 8 Byte IV  
Multi DRM - MultiPeriod - 8 Byte IV  
Widevine - 16 Byte IV  
Widevine - 16 Byte IV - (mix of encrypted and unencrypted periods)  
**Offline**  
(custom asset)

- To use the player for testing you'll need to edit the following:
  - The **Manifest URL** for the packaged MPD
  - The **Proxy URL** - The Widevine **ServerURL** value from the EZDRM Encryption Key file.



## Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:

Custom manifest:

Custom license server:

Custom license certificate URL:



3. Click the Load button to see the test playback:

# Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming. Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:

Custom manifest:

Custom license server:

Custom license certificate URL:

**Load**

**Unload**



When testing playback, you can also test sending your Custom Data. Enter the Custom Data as part of the Proxy URL as shown in the URL above and the example below:

## Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:	<input type="text" value="(custom asset)"/>
Custom manifest:	<input type="text" value="https://wowza.ezdrm.com/vod/mp4:sample.mp4/r"/>
Custom license server:	<input type="text" value="zdrm.com/proxy?px=XXXXXX&amp;customdata=12345"/>
Custom license certificate URL:	<input type="text"/>

**Load**

**Unload**



## PlayReady over DASH (EZDRM Universal)

### *Proxy URL*

The PlayReady Proxy URL has two parts, the first is the base URL with your account PX value.

The base URL is: <https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX>


The px value is the last 6 digits highlighted in yellow above.

### *PX Value*

The **pX value** is branded for your account and is account specific. Your PX value is always the same no matter what packager you are using.

It is the last six characters of your EZDRM Profile ID, as shown in this example:

1. Login to your EZDRM.com account
2. Under "My Account" click on the link to **My PlayReady Account** under **PlayReady DRM**.
3. The last 6 digits of your PlayReady Profile ID is your PX Number.



[DRM](#)
[DEMOS](#)
[RESOURCES](#)
[TRY FREE](#)
[COMPANY](#)
[CONTACT](#)

MY ACCOUNT

FAIRPLAY DRM

PLAYREADY DRM

My PlayReady Account

Licenses Issued

### My PlayReady Account Settings

Your PlayReady Profile ID is 4XXXXXXE-5XX9-4XX4-8XX1-346593E01XXF

Your authorization url is <https://www.ezdrm.com/demo/SilverLightDRM-OOB-Out-of-Browser/postbackurl.asp>  
 To learn more about what an authorization url is, as well as to update it, please go [here](#).

For help with setting up, please take a look at our [setup documentation](#).

## PlayReady Auth URL

The EZDRM system passes several values by default to your authorization URL:

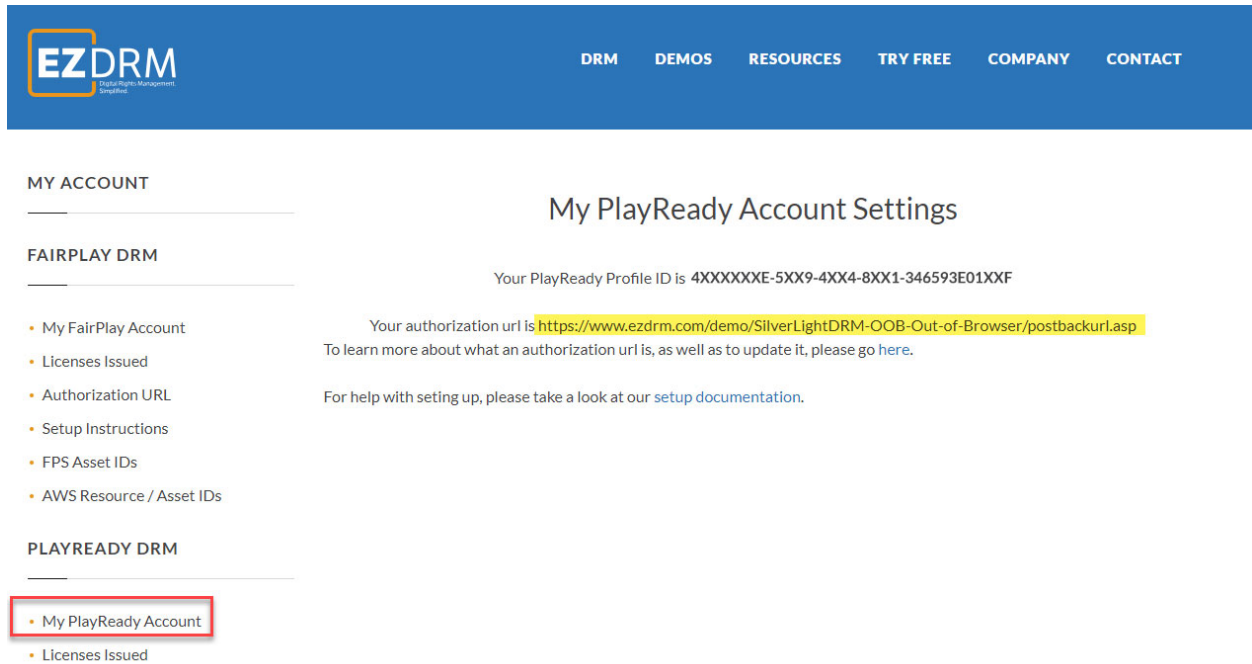
- The IP address of the end client
- EZDRM TOKEN, this is for EZDRM use only
- PX value, this is your EZDRM specific PX value for your account
- Custom Data is added to the end of the string (see next section)

Here is an example of the EZDRM return post to your auth URL:

[https://Your\\_URL.com/auth?IP=10.10.10.10&token=5060ac27-52a7-0151-2bc4-da9055cb0afb&CustomData=samplecustomdata](https://Your_URL.com/auth?IP=10.10.10.10&token=5060ac27-52a7-0151-2bc4-da9055cb0afb&CustomData=samplecustomdata)  
 pX=123456

**Note:** Custom Data is discussed in more detail in the next section. There is a line break after Custom Data where the PX value is returned.

You can find and edit your authorization URL by logging into your EZDRM account, under “My Account” click on the link to **My PlayReady Account** under **PlayReady DRM**.



The screenshot shows the EZDRM website's 'My PlayReady Account Settings' page. The top navigation bar includes links for DRM, DEMOS, RESOURCES, TRY FREE, COMPANY, and CONTACT. The left sidebar has sections for 'MY ACCOUNT' and 'FAIRPLAY DRM'. Under 'FAIRPLAY DRM', there is a list of links: 'My FairPlay Account', 'Licenses Issued', 'Authorization URL', 'Setup Instructions', 'FPS Asset IDs', and 'AWS Resource / Asset IDs'. The 'PLAYREADY DRM' section is also visible, with 'My PlayReady Account' highlighted by a red box. The main content area displays the 'My PlayReady Account Settings' title, the user's PlayReady Profile ID (4XXXXXXE-5XX9-4XX4-8XX1-346593E01XXF), and the authorization URL (https://www.ezdrm.com/demo/SilverLightDRM-OOB-Out-of-Browser/postbackurl.asp). It also provides instructions on how to learn more about the authorization URL and how to set up the account.

## Custom Data (Optional)

This additional configuration to your PlayReady Proxy URL will pass additional business logic for license validation.

You can pass custom logic to your authentication URL by adding your own values to the end of your PlayReady LAURL, as follows:

<https://playready.ezdrm.com/ww/preauth.aspx?pX=XXXXXX&CustomData=123&CustomData2=&CustomData3=>

Depending on your DASH Player, such as the Bitmovin player that has a Custom Data value that uses a Microsoft Custom Data element, you can either use this object or pass custom data via adding values to your PlayReady License Acquisition URL (LA\_URL). This is the PlayReady LAURL value that is returned in your packaging XML license values.

## Sample Player for PlayReady over DASH

An example playback player for PlayReady DASH content is the Shaka Player found here: <https://shaka-player-demo.appspot.com/demo/>

Microsoft PlayReady uses IE and Edge web browsers.

*Note: all calls must be over https.*

1. Select the **(custom asset)** option from the **Asset** dropdown:

## Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.  
Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset: Angel One (multicodec, multilingual)

Custom manifest: Live sim (6s segments)

Custom license server: Wowza

Custom license certificate URL: Big Buck Bunny (Live)

**Load** **Un**

Bitcodin

Art of Motion (DASH)

Art of Motion (HLS, TS)

Sintel (HLS, TS, 4k)

**Azure Media Services**

Azure Trailer

Tears Of Steel (external text)

**GPAC**

live profile with five periods

main profile, single file

onDemand profile

**Verizon Digital Media Services**

Multi DRM - 8 Byte IV

Multi DRM - MultiPeriod - 8 Byte IV

Widevine - 16 Byte IV

Widevine - 16 Byte IV - (mix of encrypted and unencrypted periods)

Offline

(custom asset)

2. To use the player for testing you'll need to edit the following:
  - The **Manifest URL** for the packaged MPD
  - The **Proxy URL** - The PlayReady **LAURL** value from the EZDRM Encryption Key file.

## Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:

Custom manifest:

Custom license server:

Custom license certificate URL:



3. Click the Load button to see the test playback:



## Shaka Player v2.4.6-debug

uncompiled | compiled (debug) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:

Custom manifest:

Custom license server:

Custom license certificate URL:

**Load**

**Unload**



When testing playback, you can also test sending your Custom Data. Enter the Custom Data as part of the Proxy URL as shown in the URL above and the example below:

## Shaka Player v2.4.6-debug

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:	<input type="text" value="(custom asset)"/>
Custom manifest:	<input type="text" value="https://wowza.ezdrm.com/vod/mp4:sample.mp4/"/>
Custom license server:	<input type="text" value="ency/preauth.aspx?pX=E01XXF&amp;CustomData=1"/>
Custom license certificate URL:	<input type="text"/>
<div><div>Load</div><div>Unload</div></div>	



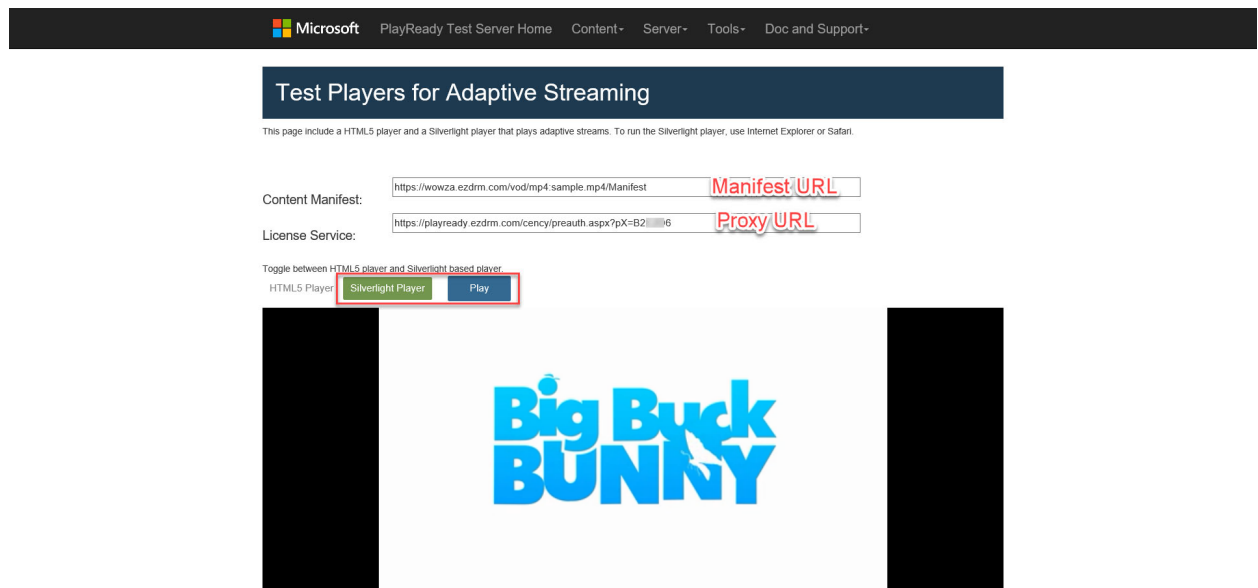
## Sample Player for PlayReady over Microsoft Smooth Streaming

An example playback player for PlayReady over Microsoft Smooth Streaming can be found here: <https://test.playready.microsoft.com/Tool/PlayerHAS>

Microsoft PlayReady uses IE and Edge web browsers.

*Note: all calls must be over https.*

1. To use the player for testing you'll need to edit the following:
  - **Content Manifest:** enter your **Manifest URL**
  - **License Service:** The **Proxy URL** - The PlayReady **LAURL** value from the EZDRM Encryption Key file
2. Select **HTML5** or **Silverlight Player** and click the **Play** button.



## Apple FairPlay Streaming over HLS

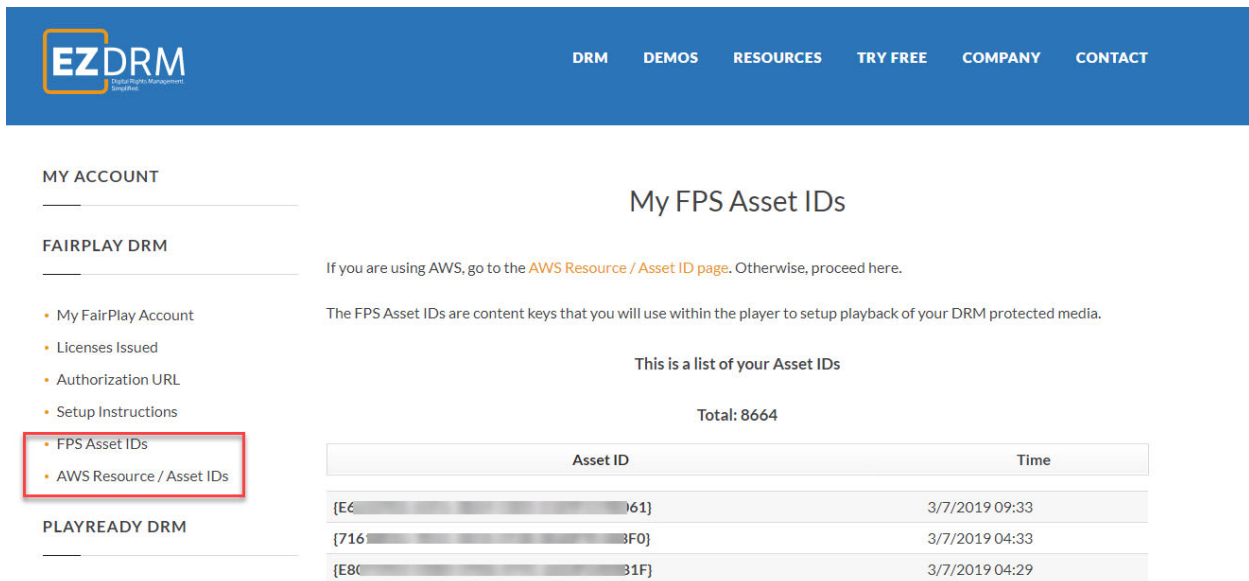
Apple FairPlay uses SPCPath. This the base URL is:

<https://fps.ezdrm.com/api/licenses/AssetID>

## EZDRM FairPlay Account

### Asset ID

To access your FPS asset IDs, log in to your EZDRM account. Under **FairPlay DRM**, select **FPS Asset IDs**, the IDs will be listed by date created. For AWS and Anevia asset IDs, select **AWS Resource / Asset IDs**.



**EZDRM** Digital Rights Management. Simplified.

DRM DEMOS RESOURCES TRY FREE COMPANY CONTACT

MY ACCOUNT

FAIRPLAY DRM

- My FairPlay Account
- Licenses Issued
- Authorization URL
- Setup Instructions
- FPS Asset IDs**
- AWS Resource / Asset IDs

PLAYREADY DRM

### My FPS Asset IDs

If you are using AWS, go to the [AWS Resource / Asset ID page](#). Otherwise, proceed here.

The FPS Asset IDs are content keys that you will use within the player to setup playback of your DRM protected media.

This is a list of your Asset IDs

Total: 8664

Asset ID	Time
{E6...}61}	3/7/2019 09:33
{716...}3F0}	3/7/2019 04:33
{E8C...}31F}	3/7/2019 04:29

### Authorization URL

To access your authorization URL for Apple FairPlay, log in to your EZDRM account. Under **My Account**, Select **My FPS Authorization URL**.

## MY ACCOUNT

### FAIRPLAY DRM

- My FairPlay Account
- Licenses Issued
- Authorization URL
- Setup Instructions
- FPS Asset IDs
- AWS Resource / Asset IDs

## My FPS Authorization URL

In order to issue out licenses for PlayReady protected content, we need to know your authorization URL. Please make sure to provide the full URL, including the "http://".

Your authorization URL:

UPDATE

## Passing Custom Data

As part of the licensing process, the content owner can pass a set of parameters through the EZDRM system in order to use that information within the business logic. These parameters should be attached to the server URL provided above.

This is an example in the HTML player:

```
var serverProcessSPCPath = '<<FPS PATH TO LICENSE SERVER>>/api/licenses/<<ASSET ID>>?<<KEY1>>=<<VALUE1>>&<<KEY2>>=<<VALUE2>>';
var serverProcessSPCPath = 'http://fps.ezdrm.com/api/licenses/09cc0377-6dd4-40cb-b09d-XXXXXXX?CustomValue=Value1?CustomValue2=Value2';
```

The EZDRM system passes several values by default to your Authorization URL:

- The FPS License Server Path
- **AssetID** – changes per the asset; the AssetID is generated during DRM key generation. Asset IDs can be found under FairPlay DRM in your EZDRM account by date created.
- **Custom Data** values
- The **Client IP** of the end client is added to the end of the string

Here is an example of the EZDRM return post to your Authorization URL:

<https://fps.ezdrm.com/api/licenses/09cc0377-6dd4-40cb-b09d-XXXXXXX?customdata=123?customdata=345&ClientIP=12.34.567.89>

## Sample Players for Apple FairPlay Streaming

For an iOS offline example, you can download using this link:

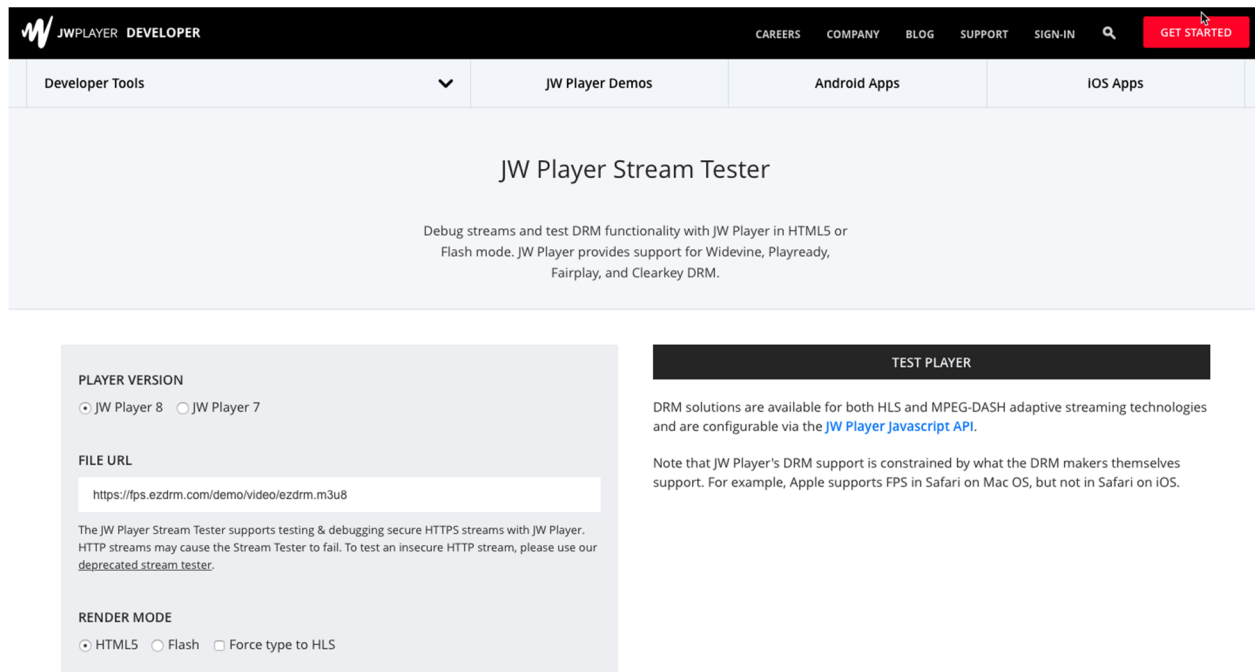
<https://www.ezdrm.com/downloads/fps-ios-demo-updates-3.zip>

To use the player for testing in Safari you'll need to:

- Edit the Cert file
- Edit the M3U8

Another player example can be found using this link:

<https://developer.jwplayer.com/tools/stream-tester/>



The screenshot shows the JW Player Developer website's 'Stream Tester' tool. The header includes navigation links like CAREERS, COMPANY, BLOG, SUPPORT, SIGN-IN, and a GET STARTED button. The main content area is titled 'JW Player Stream Tester' and describes its purpose: 'Debug streams and test DRM functionality with JW Player in HTML5 or Flash mode. JW Player provides support for Widevine, Playready, Fairplay, and Clearkey DRM.'

The interface is divided into two main sections:

- PLAYER VERSION:** Radio buttons for 'JW Player 8' (selected) and 'JW Player 7'.
- FILE URL:** A text input field containing 'https://fps.ezdrm.com/demo/video/ezdrm.m3u8'.
- RENDER MODE:** Radio buttons for 'HTML5' (selected), 'Flash', and a checkbox for 'Force type to HLS'.

Below the input fields, there is a note: 'The JW Player Stream Tester supports testing & debugging secure HTTPS streams with JW Player. HTTP streams may cause the Stream Tester to fail. To test an insecure HTTP stream, please use our [deprecated stream tester](#).'

On the right side, there is a 'TEST PLAYER' section with the following text: 'DRM solutions are available for both HLS and MPEG-DASH adaptive streaming technologies and are configurable via the [JW Player javascript API](#). Note that JW Player's DRM support is constrained by what the DRM makers themselves support. For example, Apple supports FPS in Safari on Mac OS, but not in Safari on iOS.'

Enter the following parameters to test playback:

**PLAYER VERSION**

☒ JW Player 8 ☐ JW Player 7

**FILE URL**

The JW Player Stream Tester supports testing & debugging secure HTTPS streams with JW Player. HTTP streams may cause the Stream Tester to fail. To test an insecure HTTP stream, please use our [deprecated stream tester](#).

**RENDER MODE**

☒ HTML5 ☐ Flash ☐ Force type to HLS

**DRM**

☐ None

☐ Widevine

☐ Playready

☒ Fairplay

**Certificate URL:**

t.cer


**Process SPC URL:**

7-6dd4-4 d-b5 e

☐ Clearkey

**TEST STREAM**

**TEST PLAYER**



DRM solutions are available for both HLS and MPEG-DASH adaptive streaming technologies and are configurable via the [JW Player javascript API](#).

Note that JW Player's DRM support is constrained by what the DRM makers themselves support. For example, Apple supports FPS in Safari on Mac OS, but not in Safari on iOS.

- **FILE URL** – this is the stream URL to test
- **Certificate URL** – the Certificate URL for your stream
- **Process SPC URL** – this is the SPC Base URL with your AssetID.

Enter the parameters and click the **Test Stream** button.