

EZDRM DRM Keys Guide

Table of Contents

Table of Contents	2
Universal DRM (CENC-Widevine & PlayReady)	3
<i>Generating Keys</i>	3
<i>Universal DRM Key Value Definitions</i>	7
Apple FairPlay Streaming	9
<i>Generating Keys</i>	9
<i>Apple FairPlay Streaming Key Value Definitions</i>	12
Appendix 1 – Installing ARC Plug-in	13
Appendix 2 – Call for Existing EZDRM Keys	15
<i>Universal DRM</i>	15
<i>Apple FairPlay Streaming</i>	18

Universal DRM (CENC-Widevine & PlayReady)

EZDRM Universal DRM is a combination of Google Widevine Modular with Microsoft PlayReady; both using linked CENC keys over DASH streaming. This enables a content owner to encrypt the media once with CENC keys and deliver either a PlayReady License or a Widevine License depending on the player and platform calling for a license.

Generating Keys

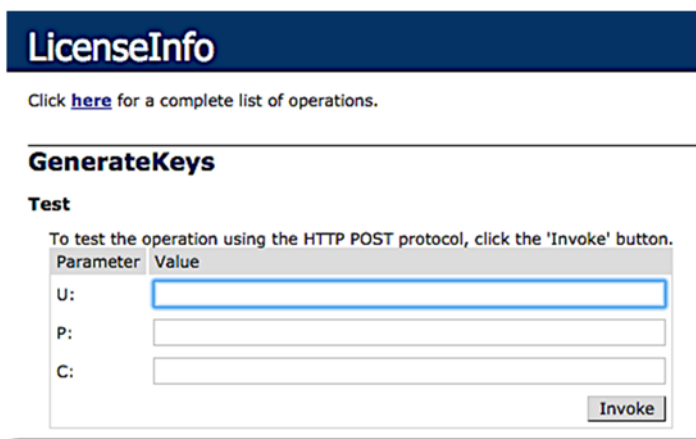
To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Web Service

1. Call the EZDRM web service in a browser:

<https://wvm.ezdrm.com/ws/LicenseInfo.aspx?op=GenerateKeys>

2. Generate Key values by entering the parameters values and clicking “Invoke”.



LicenseInfo

Click [here](#) for a complete list of operations.

GenerateKeys

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
U:	<input type="text"/>
P:	<input type="text"/>
C:	<input type="text"/>

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	Content_ID **optional

Note: The Content_ID is optional. The first time you use this web service it will be blank. For additional calls it can be blank for new keys or use an existing Content_ID. Sending a Content_ID will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you don't send this value, the web service will automatically generate a unique Content_ID. If you call a Content_ID, you will get the back all the DRM key information for that Content_ID. See [Appendix 2](#) for more information on calling existing keys with Content_ID.

3. The response from EZDRM will look like this:

```

<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXlIBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXb24XXXb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXXXXLbg="}
    </ServerGet>
    <ResponseRaw>
      {"status": "OK", "drm": [{"type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed"}], "tracks":
      [{"type": "SD", "key_id": "WVXXXXXXXXlIBEXXw+XXXXX==", "key": "WVXXXXXXXXlIBEXXw+XXXXX==", "pssh":
      [{"drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXXX0=</Checksum>
  </PlayReady>
</EZDRM>

```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C= ""'
```

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password
C	Content_ID **optional, for blank pass ""

Note: Although Content_ID is optional you must pass a "" for blank if you do not specify a Content_ID.

The following is returned from the web service:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6lxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}, {"content_id": "6lxXXx0Z8xXXXXXXXXXXLbg=="}]}</ServerGet>
  </WideVine>
  <ResponseRaw>
    {"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"},"tracks":[{"type":"SD","key_id":" WVXXXXXXXXliBEXXw+XXXXX==","key":" W5XXXXXXXXZHxTjhXXXXvw==","pssh":{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXXX=="}]}]}</ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <LAURL>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX</LAURL>
    <Checksum>1Xq+XXXXXX0=</Checksum>
  </PlayReady>
</EZDRM>
```


Universal DRM Key Value Definitions

Here are the descriptions of the key values returned by EZDRM:

```

<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXHxTjhXXXXXvW==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXlibEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXb24XXXb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [ {"type": "SD"} ], "content_id": "6IxXXx0Z8xXXXXXXXXXXLbg==" }
    </ServerGet>
    <ResponseRaw>
      {"status": "OK", "drm": [{"type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed"}], "tracks": [{"type": "SD", "key_id": "WVXXXXXXXXlibEXXw+XXXXX==", "key": "WVXXXXXXXXlibEXXw+XXXXX==", "pssh": [{"drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXXX=="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXHxTjhXXXXXvW==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXXX0=</Checksum>
  </PlayReady>
</EZDRM>

```

Widevine:

- **ContentID:** an internal ContentID to the DRM system. Sending a Content_ID will allow you to encrypt content with the same DRM values as other content and have that content share one license.
- **Key:** The DRM content encryption key (128 bit key); Base 64 encoded.
- **KeyHEX:** The DRM content encryption key; coded in Hex format.
- **KeyID:** The media key ID used for encryption (also known as KID); Base 64 encoded.
- **KeyIDGUID:** The media key ID used for encryption; coded in GUID format.
- **KeyIDHEX:** The media key ID used for encryption; coded in Hex format.
- **PSSH:** The Widevine modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

- **ServerURL:** The Widevine proxy URL unique to your account that will be used for playback.

PlayReady:

- **Key:** This value is the same as the Widevine Key above, because the DRM is using common encryption with a shared key base; Base 64 encoded.
- **KeyHEX:** This value is the same as the Widevine KeyHEX above; coded in Hex format.
- **KeyIDGUID:** This value is the same as the Widevine KeyIDGUID above; coded in GUID format.
- **LAURL:** The license acquisition URL for PlayReady unique to your account that will be used for playback.
- **Checksum:** some encoders require this value for the PlayReady encryption Key and KeyID.

Apple FairPlay Streaming

EZDRM Apple FairPlay DRM is a hosted Apple FairPlay Streaming (DRM). This enables a content owner to encrypt the media with Apple FPS DRM keys and deliver content Apple devices with native support MAC Safari browser via HTML 5 player or iOS via native App or Safari 11.3.

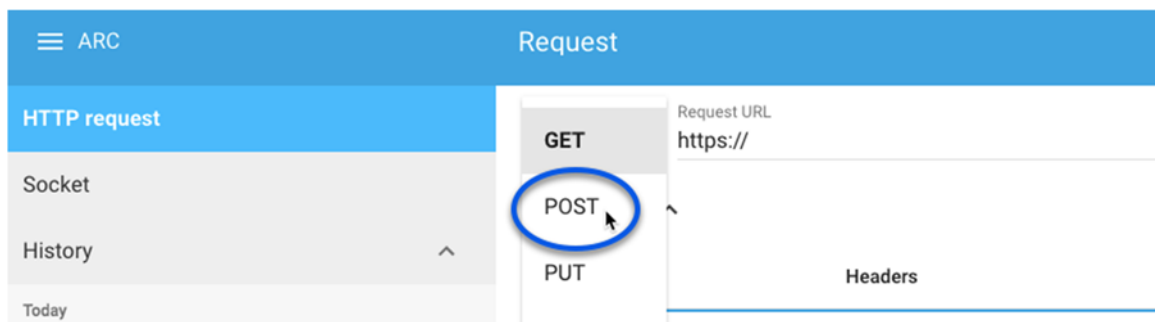
The packaging process encrypts the media. This is accomplished via a secure web call to the EZDRM Key Servers API. The Key Server API will return an XML response with the DRM key structure.

Generating Keys

To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM Key Servers API, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Key Servers API

1. To request the DRM keys through Advanced REST client (ARC) API, open a session and select HTTP Request. If you do not have the ARC plug-in for Chrome, see [Appendix 1](#) for instructions.
2. Change the Method dropdown to **POST**.



3. Enter the Request URL below updated with your username and password:

<http://fps.ezdrm.com/api/keys?u=USERNAME&p=PASSWORD>

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password

Request

Method: POST Request URL: <http://fps.ezdrm.com/api/keys/AssetID?u=USERNAME&p=PASSWORD> SEND

Parameters ^

Headers Body Variables

Toggle source mode + Insert headers set

Header name Header value

ADD HEADER

Headers are valid Headers size: bytes

200 OK 214.10 ms DETAILS

There was an error parsing XML data

```

^ <FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>

```

4. The following is an example of the response:

```

<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>

```

5. A unique **AssetID** is assigned the first time you call the API, to get existing keys see [Appendix 2](#).

Option 2: Request DRM keys with curl

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -X POST 'http://fps.ezdrm.com/api/keys?U=Username&P=Password' -d ''
```

Note: there is a space between the single quotes at the end of the line.

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password

The following is returned:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrj/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

Apple FairPlay Streaming Key Value Definitions

The following values are used for EZDRM Apple FairPlay encryption.

- **SAMPLE-AES** – The EZDRM encryption method used for Apple FairPlay. The individual audio and video samples that make up an Apple HLS chunk are encrypted using AES-128 encryption. This method is supported directly on devices running iOS 6.0 and greater.
- **Key:** The 128-bit content encryption key.
- **Initialization Vector (IV):** The initialization vector (IV) used for encryption.

Here are the descriptions of the key values returned by EZDRM for Apple FairPlay:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

- **AssetID:** the internal EZDRM ContentID assigned the first time you call the API. Sending a AssetID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. For more information on returning existing keys see [Appendix 2](#).
- **KeyHEX:** the DRM encryption key. The first 32 characters are the **key** value and the last 32 characters are the **IV** value; combined in Hex format.

Sample KeyHEX:

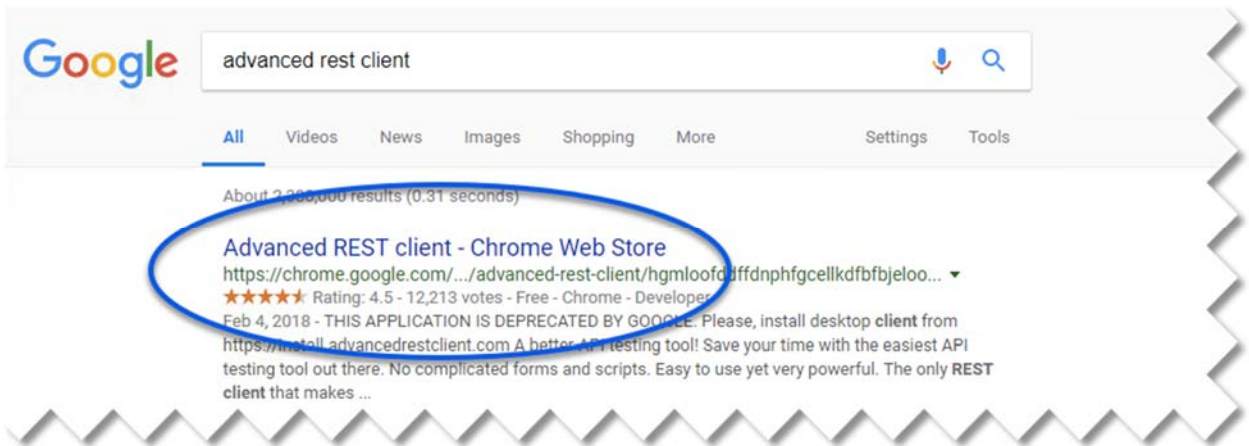
```

                                FPS "key"                                FPS "IV"
<KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
```

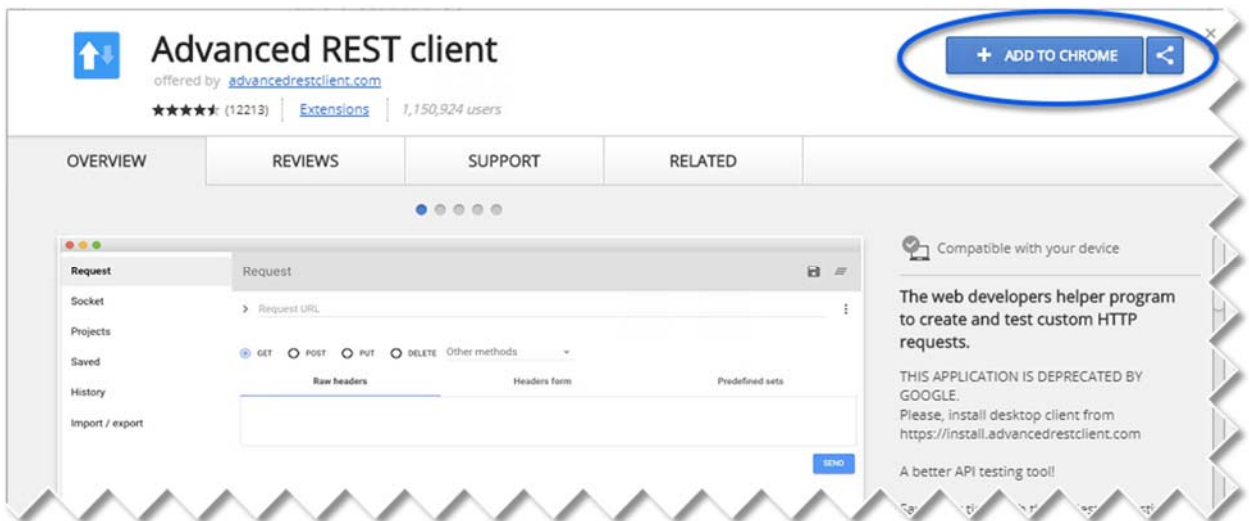
- **KeyUri** the License URL used for the license key call.

Appendix 1 – Installing ARC Plug-in

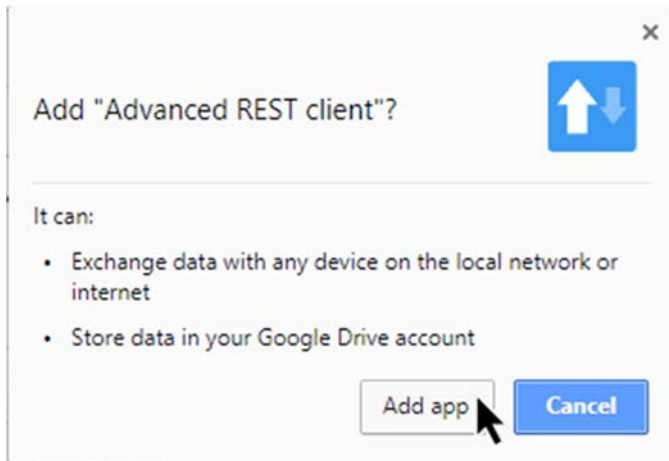
- a. To install the Advanced REST client (ARC) plug-in for Chrome in order to call the EZDRM Key Servers API, search for “Advanced REST client” using the Google search engine.
- b. Click the “Advanced REST client – Chrome Web Store” link.



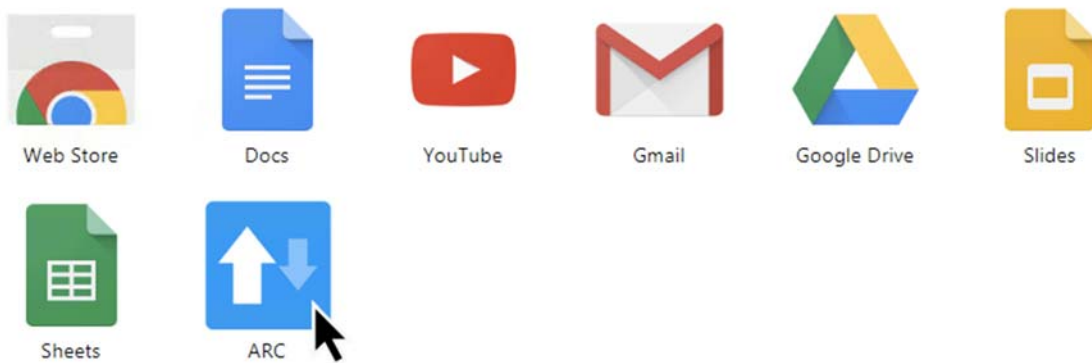
- c. Click on the **+Add to Chrome** button to install the plug-in.



- d. Click the **Add app** button to confirm installation.



e. Open the plug-in by clicking the ARC app button.



Return to [Apple FairPlay Streaming](#) section to continue instructions for calling the EZDRM Key servers API.

Appendix 2 – Call for Existing EZDRM Keys

Universal DRM

Sending a Content_ID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call a Content_ID you will get the back all the DRM key information for that Content_ID.

Option 1: Request existing DRM keys using EZDRM Web Service

1. Call the EZDRM web service in a browser:

<https://wvm.ezdrm.com/ws/LicenseInfo.aspx?op=GenerateKeys>

2. Return Key values by entering the parameter values including the existing Content_ID and click “Invoke”.

The screenshot shows the 'LicenseInfo' section of the EZDRM web service. Below the title, there is a link to a complete list of operations. The 'GenerateKeys' section includes a 'Test' area with instructions to use the HTTP POST protocol. A form with three input fields is shown: 'U:' for username, 'P:' for password, and 'C:' for Content ID. The 'C:' field contains the value '6ixXXx0Z8xXXXXXXXXXXLbg=='. An 'Invoke' button is located at the bottom right of the form.

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	<u>Content ID</u>

- Your return results will always return the existing DRM keys that are tied to the Content_ID.

```

▼<EZDRM xmlns="">
  ▼<WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHXtjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb</KeyIDHEX>
    ▼<PSSH>
      EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    ▼<ServerGet>
      request={"policy": "", "tracks": [{"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXXXXLbg==" }
    </ServerGet>
    ▼<ResponseRaw>
      {"status": "OK", "drm": [{"type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed"}], "tracks": [{"type": "SD", "key_id": "WVXXXXXXXXliBEXXw+XXXX==", "key": "WVXXXXXXXXliBEXXw+XXXX==", "pssh": [{"drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX="}]}]}
    </ResponseRaw>
  </WideVine>
  ▼<PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHXtjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXXXXXX56bf </KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    ▼<LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXX0=</Checksum>
  </PlayReady>
</EZDRM>

```

Option 2: Request existing DRM keys with curl

The second option to request existing DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service with the existing DRM keys based on the Content_ID.

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=CONTENT_ID'
```

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password
C	<u>Content ID</u>

The web service will always return the existing DRM keys that are tied to the Content_ID.

Sample script with Content_ID:

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=6lxXXx0Z8xXXXXXXXXXXLbg=='
```

Existing DRM Keys returned:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6lxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?px=XXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [{"type": "SD"}], "content_id": "6lxXXx0Z8xXXXXXXXXXXLbg=="}</ServerGet>
  et>
  <ResponseRaw>
    {"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"},"tracks":[{"type":"SD","key_id":"WVXXXXXXXXliBEXXw+XXXXX==","key":"W5XXXXXXXXZHxTjhXXXXvw==","pssh":{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX=="}}]}</ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>https://playready.ezdrm.com/cency/preauth.aspx?px=XXXXX</LAURL>
    <Checksum>1Xq+XXXXX0=</Checksum>
  </PlayReady>
</EZDRM>
```

Apple FairPlay Streaming

Sending a AssetID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call an AssetID you will get the back all the DRM key information for that AssetID.

Option 1: Request existing DRM keys using EZDRM Key Servers API

1. To request existing DRM keys through Advanced REST client (ARC) API, open a session and select HTTP Request. If you do not have the ARC plug-in for Chrome, see [Appendix 1](#) for instructions.
2. Change the Method dropdown to **GET**.



3. Enter the Request URL below updated with your AssetID, username, and password:

<http://fps.ezdrm.com/api/keys/AssetID?u=Username&p=Password>

The parameters are as follows:

Parameter	Description
AssetID	Existing FairPlay AssetID
u	EZDRM username
p	EZDRM password



The GET will always return the existing DRM keys that are tied to the **AssetID**.

Option 2: Request DRM keys with curl

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -X POST 'http://fps.ezdrm.com/api/keys/AssetID?U=Username&P=Password' -d ''
```

Note: there is a space between the single quotes at the end of the line.

The parameters are as follows:

Parameter	Description
-----------	-------------

AssetID	Existing FairPlay AssetID
U	EZDRM username
P	EZDRM password

The response will always return the existing DRM keys that are tied to the **AssetID**.

Sample curl script with AssetID:

```
curl -X POST 'http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25?U=Username&P=Password' -d ''
```

Existing DRM Keys returned:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```