EZDRM Bento4 Configuration Open Source

Table of Contents

Version 2.3 / Updated Sept 28, 2021

Prerequisites

Bento Binary Release (SDK with header files, libraries, and command line applications). Download here: https://www.bento4.com/downloads/

Python 3.6 or above is recommended.

To download Python 3.6: https://www.python.org/downloads/

Bento 4 Packager Encryption – Widevine & PlayReady

Generating Keys

Below are the steps to create the DRM Keys for CENC-PlayReady or CENC-Widevine encryption for Bento4 (Open Source).

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 CPIX Web Service

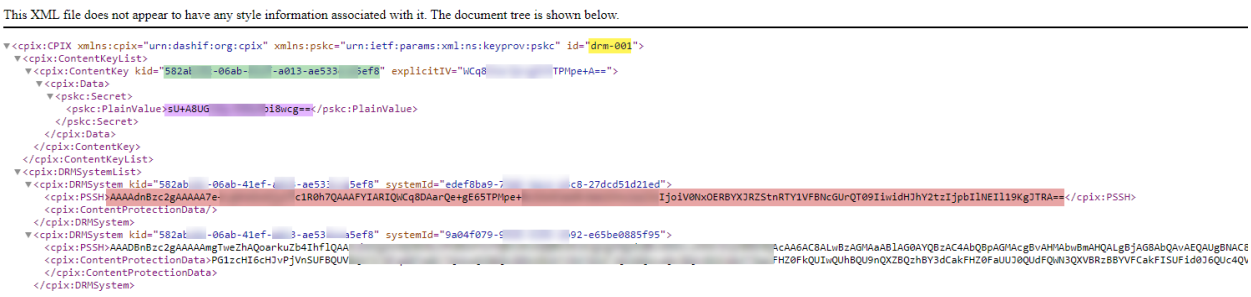
1. Call the EZDRM web service in a browser: https://cpix.ezdrm.com/keygenerator/cpix.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

   The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

2. The response from EZDRM will look like this:



- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value.
- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="drm-001">
<cpix:ContentKeyList>
<cpix:ContentKey kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" explicitIV="WCq8DAXXXXXXE65TPMpe+A==">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>sU+XXXXXXqvSN0aXbXXXXX==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
```

```
<cpix:DRMSystemList>
<cpix:DRMSystem kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" systemId="edefXXXX-79d6-XXXX-a3c8-27dcd5XXXXXd">
<cpix:PSSH>AAAAdnBzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXXXAAFYIARIQWCq8DAarQe+gE65TPMpe+BoIbW92aWRvbmUiMnsia2lkIjo
iV0NxQERBYXJRZStnRTY1VFBNcGUrQT09IXXXXHJhY2tzIjpbIlNEIl19KgJTRA==</cpix:PSSH>
<cpix:ContentProtectionData/>
</cpix:DRMSystem>
```

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 "https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=drm-001"
```

Widevine and PlayReady Encryption

Once you have the DRM values to encrypt the content, you can add them to the Bento4 open-source packager for CENC-PlayReady or CENC-Widevine encryption.

1. First you need to fragment the MP4. Open command prompt and navigate to the MP4 file.

   For this example, we are fragmenting the MP4 file named "**BigBuckBunny_320x180.mp4**". The command would be:

   ```
   mp4fragment.exe inputname.mp4 inputname-frag.mp4
   ```

   Sample command:

   ```
   c:\Users\User\Downloads\Bento4-SDK-1-5-1-622.x86-microsoft-win32-vs2010\bin>
   mp4fragment.exe BigBuckBunny_320x180.mp4 BigBuckBunny_320x180_Frag.mp4
   ```

2. This will result in a file called **BigBuckBunny_320x180_Frag.mp4** in the **bin** folder. Move this file into the **utils** folder.

3. Next you can package the media into the DASH format. Use the following syntax in Command prompt from **utils** folder.

   ```
   mp4-dash.py --widevine-header="#PSSH" --playready-header=LA_URL:https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXX --encryption-key=KID:KEYHEX video-source.mp4
   ```

   Note: Be sure to include the # before the PSSH Data value.

The description for the lines of syntax include:

- Use the command line option **--widevine-header** to specify the Widevine **PSSH** Data value with a **#** in front of the value.

- Use the command line option --**playready-header** to specify the EZDRM PlayReady **license_acquisition_url** provided by EZDRM including your **PX Value**.

  PlayReady:
  https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXX

  Note: Your PlayReady **PX value** is the last six characters of your PlayReady Profile ID. The appropriate one is required for all packagers you use. For more details on how to find your PX value refer to the **EZDRM Testing Playback** guide at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation.**

- Use the command line option --**encryption-key** to specify the EZDRM Key. Bento4 refers to encryption-key as **KID**:**KeyHEX**.

  For example:

  582dXXXX06abXXXXa013XXXX3cca5ef8:B14FXXXX41B7XXXXD237XXXX6E2XXXX2

  The **KID** is the EZDRM **kid** value (no dashes) and for the **KeyHEX** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

  pskc:Secret key (Base 64) = sU+A8UXXXXvSN0aXXXXwcg==

  (KeyHEX) = B14FXXXX41B7XXXXD237XXXX6E2XXXX2

- **video-source.mp4** is the source file.

Sample command from **utils** folder:

```
mp4-dash.py --widevine-header=#AAAAdnBzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXX7QAAAFYIARIQWCq8DAarQe+gE65IPMoe+Bo
IbW92aWRvbmUiMnsia2lkIjoiV0NxOERBYXJRZStnRTY1XXXXXXXXX09IiwidHJhY2tzIjpbIlNEIl19KgJTRA== --playready-heade
r=LA_URL:https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXF --encryption-key=582dXXXX06abXXXXa013XXX
X3cca5ef8:B14FXXXX41B7XXXXD237XXXX6E2XXXX2 BigBuckBunny_320x180_Frag.mp4
```

```
N:\Users\TEST01\Bento4-SDK-1-6-0-637.x86_64-microsoft-win32\utils>mp4-dash.py --widevine-header="#AAAAd      AAAA7
e+LqXnWSs6jy(      bW92aWRvbmUiMnsia2lkIjoiV0NxOERBYXJRZStnRTY1VFBNcGUrQT09Iiwi
dHJhY2tzIjpbIlNEIl19KgJTRA==" --playready-header=LA_URL:"https://playready.ezdrm.com/cency/preauth.aspx?pX=E      "
--encryption-key="582abc0      33cca5ef8:B14F80F1      072" BigBuckBunny_320x180_Frag.mp4
Encrypting track IDs [1, 2] in BigBuckBunny_320x180_Frag.mp4
Parsing media file 1: tmpa8l0o5bl = Encrypted[BigBuckBunny_320x180_Frag.mp4]
Splitting media file (audio) tmpa8l0o5bl = Encrypted[BigBuckBunny_320x180_Frag.mp4]
Splitting media file (video) tmpa8l0o5bl = Encrypted[BigBuckBunny_320x180_Frag.mp4]
```

You can also script this process with curl or other web service calls.

For updated details of the syntax please refer to:
https://www.bento4.com/developers/dash/encryption-and-drm/

Bento 4 Packager Encryption – 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.

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

Option 1: Request DRM keys using EZDRM CPIX Web Service

3. Call the EZDRM web service in a browser: https://cpix.ezdrm.com/keygenerator/cpix.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

   The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

   * To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Key Value Definitions

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

- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value
- **explicitIV** – the Apple FairPlay explicit IV value

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="hyb-001">
<cpix:ContentKeyList>
<cpix:ContentKey kid="582de60c-XXXX-XXXX-a013-XXX33cca5ef8" explicitIV="WCXXXXXXXX+gE65TXXXe+A==">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>sU+A8XXXXXXXXXaXbi8wcg==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
```

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 "https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=hyb-001"
```

Apple FairPlay Encryption

Once you have the DRM values to encrypt the content, you can add them to the Bento4 open source packager for Apple FairPlay encryption.

Use the following syntax in Command prompt from the **bin** folder:

```
mp4hls --hls-version 7 --output-single-file --encryption-mode=SAMPLE-AES --encryption-key=KeyHEXIV --encryption-iv-mode=
fps --encryption-key-format=com.apple.streamingkeydelivery --encryption-key-uri=KeyUri --encryption-key-format-versions=1
video-source.mp4
```

Here are the descriptions of the keys returned from the EZDRM Key servers API:

- Use the command line option --**encryption-key** to specify the EZDRM Key. When the mode is 'fps', the encryption key must be 32 bytes: 16 bytes for the **KeyHEX** followed by 16 bytes for the **IV**.

  These are the decoded values from the **pskc:Secret key** and **explicitIV** combined (see instructions below for decoding the values).

  For the **KeyHEX** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at:
  https://tomeko.net/online_tools/base64.php?lang=en

  pskc:Secret key (Base 64) = sU+A8UXXXXvSN0aXXXXwcg==

  (KeyHEX) = b14fXXXX41b7XXXXd2374XXXXe2f3072

  Next, for the **IV** value, decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at:
  https://tomeko.net/online_tools/base64.php?lang=en

explicitIV (Base 64) = WCq8DXXXXX+gE65TXXXX+A==

IV (HEX no dashes) = 582aXXXXXXXX41efXXXae533ccaXXXX

*For example, the --**encryption-key** would look like this:*

```
--encryption-key=b14fXXXX41b7XXXXd2374XXXXe2f3072582aXXXXXXXX41efXXXae533ccaXXXX
```

- **KeyURI** - Use the command line option **--encryption-key-uri** to specify the license URL for encryption. Build by appending the **kid** value to base URL "skd://fps.ezdrm.com/;" for example: skd://fps.ezdrm.com/;582de60c-XXXX-XXXX-a013-XXX33cca5ef8

The description for the lines of syntax include:

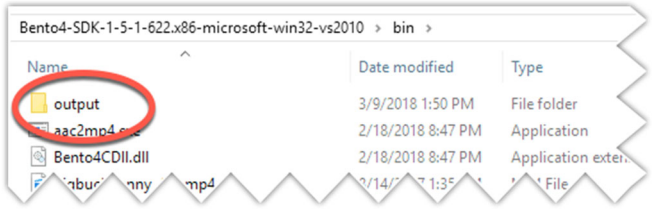
- Use the command line option **--encryption-mode=SAMPLE-AES** to select the SAMPLE-AES encryption mode.
- Use the command line option **--encryption-iv-mode=fps** to specify that the IV is delivered with the key.
- Use the command line option **--encryption-key-format=com.apple.streamingkeydelivery** to specify that FairPlay will be used to deliver the key and IV.
- **video-source.mp4** should be replaced with the name of the source file.

Sample Command from the bin folder:

```
mp4hls --hls-version 7 --output-single-file --encryption-mode=SAMPLE-AES --encryption-key=b14fXXXX41b7XXXX
d2374XXXXe2f3072582aXXXXXXXX41efXXXae533ccaXXXX --encryption-iv-mode=fps --encryption-key-format=com.apple
.streamingkeydelivery --encryption-key-uri=skd://fps.ezdrm.com/;582aXXXXXXXX41efXXXae533ccaXXXX --encrypti
on-key-format-versions=1 BigBuckBunny_450.mp4
```

```
C:\Users\TEST01\Bento4-SDK-1-6-0-637.x86_64-microsoft-win32\bin>mp4hls --hls-version 7 --output-sin
gle-file --encryption-mode=SAMPLE-AES --encryption-key=b141[redacted]072582abc0c06ab
4[redacted]f8 --encryption-iv-mode=fps --encryption-key-format=com.apple.streamingkeydeliver
y --encryption-key-uri=skd://fps.ezdrm.com/;582[redacted]a5ef8 --encryption-key-f
ormat-versions=1 BigBuckBunny_320x180.mp4
Parsing media file BigBuckBunny_320x180.mp4
```

You will now have an **output** folder in the **bin** file with the encrypted files.



You can also script this process with curl or other web service calls.

For updated details on the Bento4 Apple FairPlay syntax please refer to this link: https://www.bento4.com/developers/hls/

Additional Information

For additional questions and comments please contact: simplify@ezdrm.com