

# **EZDRM Bento4 Configuration**

## **Open Source**

---

## Table of Contents

<b>Prerequisites .....</b>	<b>3</b>
<b>Bento 4 Packager Encryption – Widevine &amp; PlayReady .....</b>	<b>3</b>
<i>Generating Keys.....</i>	<i>3</i>
<i>Widevine and PlayReady Encryption .....</i>	<i>6</i>
<b>Bento 4 Packager Encryption – Apple FairPlay Streaming .....</b>	<b>9</b>
<i>Generating Keys.....</i>	<i>9</i>
<i>Key Value Definitions.....</i>	<i>10</i>
<i>Apple FairPlay Encryption .....</i>	<i>12</i>
<b>Additional Information .....</b>	<b>14</b>

Version 2.0 / Updated January 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:  
<http://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=resourceName>

The parameters are as follows:

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

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<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="S82ab-06ab-41ef-3-ae53-5ef8" explicitIV="uQq8 TPHe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>SUHABUGb18wcg=</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSystemList>
    <cpix:DRMSystem kid="S82ab-06ab-41ef-3-ae53-5ef8" systemId="odef8ba9-7-c8-27dc51d21ed">
      <cpix:PSSH>AAAAAAdBzc2GAAAAA7e-c1R0h7QAAAFYIARIQcQ8DAarQe+gE65TPfpe+IjoIV0NxEERBYXJR2stnRTY1VFBNGURQT09IiIdHjY2tzIjpbIINEI19kgjTRA=</cpix:PSSH>
      <cpix:ContentProtectionData/>
    </cpix:DRMSystem>
  </cpix:DRMSystemList>
  <cpix:DRMSystem kid="S82ab-06ab-41ef-3-ae53-5ef8" systemId="9a04f079-5-92-e65be0885f95">
    <cpix:PSSH>AAADBNBzc2GAAAAAngTueZhaQoarKuZb4Hf1QAAACAAGACBALwBzAGHAAAB1AG0AYQ8zAC4ABQ8pAGHAcgBVAAHAAwBMAHQALgBfAG8ABQAVAEQAUGBNAC6FHZ0fKQIwQUHQU9nQXZBzHBY3dCakFHZ0faUJ3QJdFQnN3QXVBRzBBYVFCakFISUFId0j6QUC4QV
    <cpix:ContentProtectionData>PG1zchI6chVpJvN5UFBQVU...
  </cpix:DRMSystem>
</cpix:CPiX>
```

- **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>SUXXXXXXqvSN0aXbXXXXX==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" systemId="edefXXXX-79d6-XXXX-a3c8-27dcd5XXXXd">
<cpix:PSSH>AAAAdnBzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXXXAAFYIARIQWCa8DAarOe+gE65TPMpe+BoIbW92aWRvbmUiMnsia2lkIjo
iV0NxOERBYXJRZStnRTY1VFBncGUrOT09TXXXXHJhY2tzIjpbIjNEIj19KeJTRA==</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 http://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="https://playready.ezdrm.com/cency/preauth.a  
spx?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 visit: [https://www.ezdrm.com/Documentation/EZDRM\\_Testing\\_Playback\\_v2.pdf](https://www.ezdrm.com/Documentation/EZDRM_Testing_Playback_v2.pdf).

- 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: <http://guid-convert.appspot.com>

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="#AAAAdnBzc2eAAAAA7e+XXXXXXXXXXXXXXXXXXXXX70AAAFYIARIQWCq8DAarOe+eF65TPMpe+B
oTbw92aWRvbmUiMnsia2lkIjoiv0NxoERBYXJRZStnRTY1XXXXXXXXXX09IiwidHJhY2tzIjpbIjNEI119KgJTRA==" --playready-hea
der="https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXE" --encryption-key="582dXXX06abXXXa013XXX3
cca5ef8:B14FXXX41B7XXXD237XXX6E2XXX2" BigBuckBunny_320x180_Frag.mp4
```

```
C:\Users\TEST01\Bento4-SDK-1-6-0-637.x86_64-microsoft-win32\utils>mp4-dash.py --widevine-header="#AAAAdnBzc2eAAAAA7e+LqXnWSst
RIQWCq8DAarOe+gEe iMnsia2lkIjoiv0NxoERBYXJR NcGUrQT09
IiwidHJhY2tzIjpbIjNEI119KgJTRA==" --playready-header="https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXE" --
-encryption-key="582dXXX06abXXXa013XXX3cca5ef8:B14F8E2F3072" BigBuckBunny_320x180_Frag.mp4

Encrypting track IDs [1, 2] in BigBuckBunny_320x180_Frag.mp4
Parsing media file 1: tmpyu9opod1 = Encrypted[BigBuckBunny_320x180_Frag.mp4]
Splitting media file (audio) tmpyu9opod1 = Encrypted[BigBuckBunny_320x180_Frag.mp4]
Splitting media file (video) tmpyu9opod1 = Encrypted[BigBuckBunny_320x180_Frag.mp4]

C:\Users\TEST01\Bento4-SDK-1-6-0-637.x86_64-microsoft-win32\utils>
```

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

- Call the EZDRM web service in a browser:  
<http://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=resourceName>

The parameters are as follows:

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

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<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="000ae000-533cca5ef8" explicitIV="AArgA...?Mpe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>q4sp<GmTQ==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysList>
    <cpix:DRMSysSystem kid="000ae000-533cca5ef8" systemId="edef8ba9-79d...-27dcd51d21ed">
      <cpix:PSSH>AAAAAdnBzc2...XnW5s6jyCfc1R0h7QAAAFYIARIQAArgAAarQe+...oIbW92awRvbmUimnsia2IkIjoiQUFYZ0FBYXJRZStnRTY1VFBNcGUrQ
      <cpix:ContentProtectionData/>
    </cpix:DRMSysSystem>
  </cpix:DRMSysList>
</cpix:CPIX>
```

- **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+pE65TXXXe+A==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>SU+A8XXXXXXXXXXaXbi8wcg==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
</cpix:CPIX>
```

## 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://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=encyrptionkey --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. Bento4 refers to encryption-key for FairPlay as the **kid** and **IV** (**explicitIV** decoded from Base 64 to HEX) combined.

Decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at: <http://guid-convert.appspot.com/>

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



**encryption iv (HEX no dashes) = 582aXXXXXXXX41efXXXae533ccaXXXX**

The first 32 characters are used in the key file for “kid” and the last 32 characters are used for “iv”.

582aXXXXXXXX41efXXXae533ccaXXXX582aXXXXXXXX41efXXXae533ccaXXXX

- **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:



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](mailto:simplify@ezdrm.com)