EZDRM Bento4 with CPIX 2.x

Table of Contents

Version 1.0 (published August 5, 2024)

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 CENC

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 version 2 Web Service - CENC

Call the EZDRM web service in a browser:
https://cpix.ezdrm.com/keygenerator/cpix2.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

The parameters are as follows:

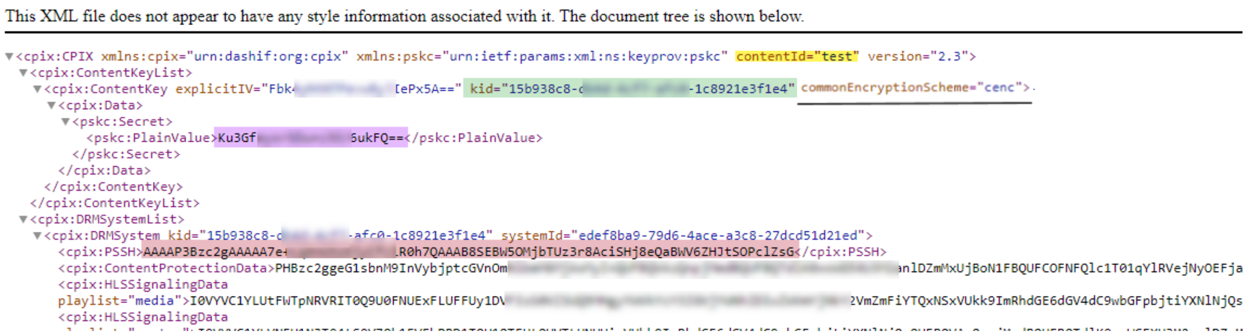
Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password

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: https://www.guidgenerator.com/.

Key Value Definitions

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.

```
▼<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
  ▼<cpix:ContentKeyList>
    ▼<cpix:ContentKey explicitIV="Fbk[redacted]IePx5A==" kid="15b938c8-[redacted]-1c8921e3f1e4" commonEncryptionScheme="cenc">
      ▼<cpix:Data>
        ▼<pskc:Secret>
           <pskc:PlainValue>Ku3Gf[redacted]6ukFQ==</pskc:PlainValue>
         </pskc:Secret>
       </cpix:Data>
     </cpix:ContentKey>
   </cpix:ContentKeyList>
  ▼<cpix:DRMSystemList>
    ▼<cpix:DRMSystem kid="15b938c8-[redacted]-afc0-1c8921e3f1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
       <cpix:PSSH>AAAAP3Bzc2gAAAAA7e[redacted]R0h7QAAAB8SEBW5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH>
       <cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOm[redacted]anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qYlRVejNyOEFja
       <cpix:HLSSignalingData
       playlist="media">I0VYVC1YLUtFWTpNRVRIT0Q9U0FNUExFLUFFUy1DV[redacted]2VmZmFiYTQxNSxVUkk9ImRhdGE6dGV4dC9wbGFpbjtiYXNlNjQs
       <cpix:HLSSignalingData
```

- **contentId** – 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: https://www.guidgenerator.com/.

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4yNtXXXXXXXyJIePx5A==" kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" commonEncryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfeXXXXXXX+z3O/6ukFQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
<cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH>
<cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbmM6MjAxMyI+QUFBQVAzQnpjMmdBQUFBQTdlK0xxWG5XU3M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qYlRVejXXXXXXXXXjhlUWFCV1Y2WkhKdFNPUGNsWnNHPC9wc3NoPg==</cpix:ContentProtectionData>
```

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/cpix2.aspx?k=kid&u=username&p=password&c=drm-001"
```

Widevine and PlayReady CENC 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.

```
python 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:

  15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415

  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) = Ku3GfXXXXXXu+z3O/6ukFQ==

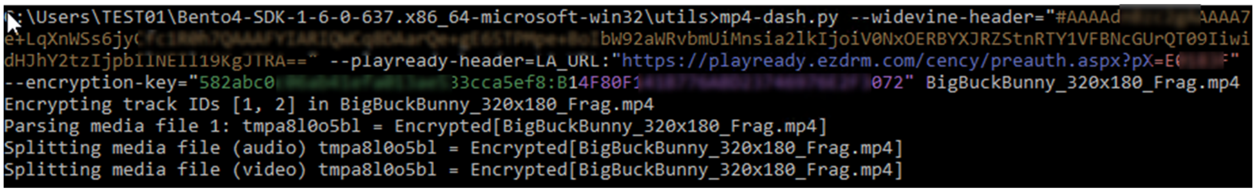
  ⬇

  (KeyHEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

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

Sample command from **utils** folder:

```
python mp4-dash.py --widevine-header=”#AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8e
QaBWV6ZHJtSOPclZsG” --playready-header=LA_URL:”https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXF” -
-encryption-key=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415 BigBuckBunny_320x180_Frag
.mp4
```



```
C:\Users\TEST01\Bento4-SDK-1-6-0-637.x86_64-microsoft-win32\utils>mp4-dash.py --widevine-header="#AAAAd[redacted]AAAA7
e+LqXnWSs6jy[redacted]bW92aWRvbmUiMnsia2lkIjoiV0NxOERBYXJRZStnRTY1VFBNcGUrQT09Iiwi
dHJhY2tzIjpbIlNEIl19KgJTRA==" --playready-header=LA_URL:"https://playready.ezdrm.com/cency/preauth.aspx?pX=E[redacted]F"
--encryption-key="582abc0[redacted]33cca5ef8:B14F80F1[redacted]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/

Widevine Only Example

```
python mp4-dash.py --widevine-header=”#AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8e
QaBWV6ZHJtSOPclZsG” --encryption-key=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415 BigB
uckBunny_320x180_Frag.mp4
```

PlayReady Only Example

```
python mp4-dash.py --playready-header=LA_URL:”https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXF” --
encryption-key=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415 BigBuckBunny_320x180_Frag.
mp4
```

Bento 4 Packager Encryption –Apple FairPlay Streaming HLS

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:
https://cpix.ezdrm.com/keygenerator/cpix2.aspx?**k=kid**&**u=username**&**p=password**&**c=resourcename**

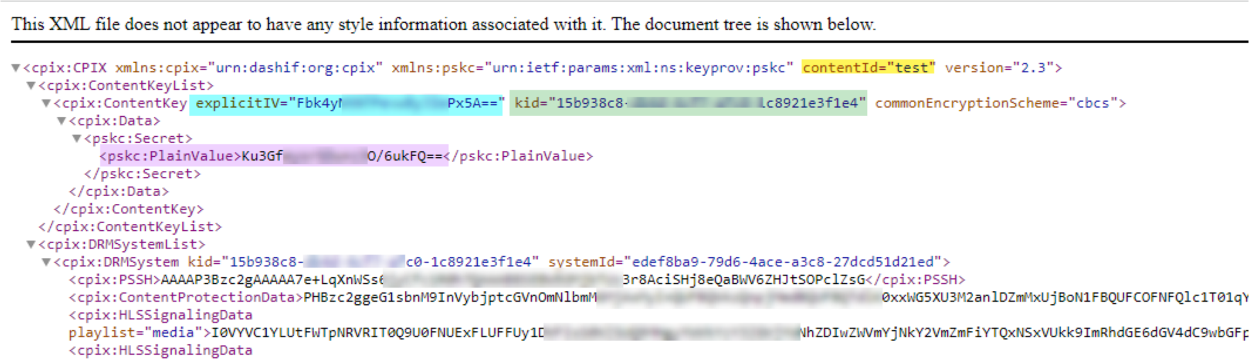
The parameters are as follows:

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: https://www.guidgenerator.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" contentId="test" version="2.3">
  ▼<cpix:ContentKeyList>
    ▼<cpix:ContentKey explicitIV="Fbk4y[redacted]Px5A==" kid="15b938c8-[redacted]c8921e3f1e4" commonEncryptionScheme="cbcs">
      ▼<cpix:Data>
        ▼<pskc:Secret>
            <pskc:PlainValue>Ku3Gf[redacted]O/6ukFQ==</pskc:PlainValue>
          </pskc:Secret>
        </cpix:Data>
      </cpix:ContentKey>
    </cpix:ContentKeyList>
  ▼<cpix:DRMSystemList>
    ▼<cpix:DRMSystem kid="15b938c8-[redacted]c0-1c8921e3f1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
        <cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs[redacted]3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH>
        <cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbmM[redacted]0xxWG5XU3M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qY
        <cpix:HLSSignalingData
        playlist="media">I0VYVC1YLUtFWTpNRVRIT0Q9U0FNUExFLUFFUy1D[redacted]NhZDIwZWVmYjNkY2VmZmFiYTQxNSxVUkk9ImRhdGE6dGV4dC9wbGFp
        <cpix:HLSSignalingData
```

- **contentId** – c value returned, generic resource name/identifier (client generated)

- **explicitIV** – the Apple FairPlay explicit IV value

- **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

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="te
st" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4XXXXXXXXwByJIePx5A==" kid="15b9XXXX-db4d-XXXX-afc0-1c89XXXXf1e4" commonEn
cryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfXXXXXXXXz3O/6ukFQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
```

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=test”
```

Apple FairPlay HLS 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:

```
python mp4-dash.py --encryption-key=KID:KeyHEX:IV --hls --encryption-cenc-scheme=cbcs --fairplay-key-uri="
skd://kid:IV" C:\2022\fragmented-bunny.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 the mode **fps** is **KID**:**KeyHEX**:**IV**

  For example:

  15b9XXXXdb4dXXXXafc01c89XXXXf1e4:**2aedXXXXecacXXXXefb3dceXXXXa415**:**15b938c8db4d4cf7afc01c8921e3f1e4**

  The **KID** is the EZDRM **kid** value (no dashes).

  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) = Ku3GfXXXXXXu+z3O/6ukFQ==

  ⬇

  (KeyHEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

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

explicitIV (Base 64) = Fbk4yNtNTPevwByJlePx5A==

↓

IV (HEX no dashes) = 15b938c8db4d4cf7afc01c8921e3f1e4

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

```
--encryption-key=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415:15b938c8db4d4cf7afc01c8921e3f1e4
```

- **KeyURI** - Use the command line option **--fairplay-key-uri** to specify the license URL for encryption. Build by appending the base URL **"skd://"** with the **kid** value (with dashes) and the **IV (HEX no dashes)** seperated by a colon. For example: **skd:// 582de60c-XXXX-XXXX-a013-XXX33cca5ef8: 15b938c8db4d4cf7afc01c8921e3f1e4**

The description for the lines of syntax include:

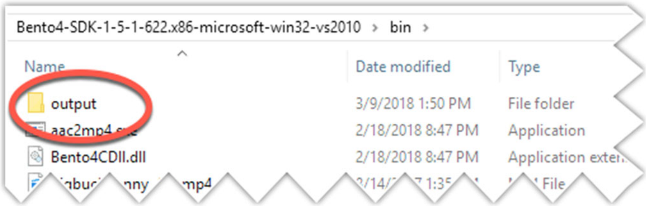
- Use the command line option **--encryption-cenc-scheme=cbcs** to select the **CBCS** encryption mode.

- **video-source.mp4** should be replaced with the name of the source file.

Sample Command from the bin folder:

```
python mp4-dash.py --encryption-key=15b9XXXXdb4d4XXXXfc01c89XXXXf1e4:2aedXXXXecacXXXXeefb3dcXXXXba415:15b9XXXXdb4dXXXXafc01c89XXXXf1e4 --hls --encryption-cenc-scheme=cbcs --fairplay-key-uri="skd://15b9XXXXdb4d4XXXXfc01c89XXXXf1e4:15b9XXXXdb4dXXXXafc01c89XXXXf1e" C:\2022\fragmented-bunny.mp4
```

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



Bento4-SDK-1-5-1-622.x86-microsoft-win32-vs2010 › bin ›

Name	Date modified	Type
output	3/9/2018 1:50 PM	File folder
aac2mp4.exe	2/18/2018 8:47 PM	Application
Bento4CDll.dll	2/18/2018 8:47 PM	Application exten

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/

Bento 4 Packager Encryption – Widevine, PlayReady & Apple FairPlay Streaming CBCS/CMAF

Generating Keys

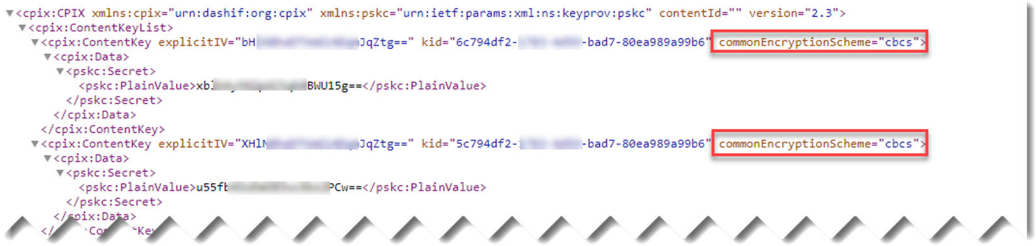
Below are the steps to create the DRM Keys for CBCS encryption for Bento4 (Open Source) including Widevine, PlayReady and Apple FairPlay Streaming.

CBCS encryption uses PlayReady 4.3 Header. When this encryption is required, an additional flag is available to return when added to the end of the query string, shown in the example as follows*:

https://cpix.ezdrm.com/KeyGenerator/cpix2.aspx?k=kid&u=username&p=password&c=resourcename&EncryptionScheme=cbcs

**CBCS is needed for CMAF*

A query with no EncryptionScheme flag returns CENC using PlayReady 4.0 Header by default.



```
▼<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="" version="2.3">
  ▼<cpix:ContentKeyList>
    ▼<cpix:ContentKey explicitIV="bH[redacted]JqZtg==" kid="6c794df2-[redacted]-bad7-80ea989a99b6" commonEncryptionScheme="cbcs">
      ▼<cpix:Data>
        ▼<pskc:Secret>
           <pskc:PlainValue>xbl[redacted]BWU15g==</pskc:PlainValue>
         </pskc:Secret>
       </cpix:Data>
     </cpix:ContentKey>
    ▼<cpix:ContentKey explicitIV="XHlN[redacted]JqZtg==" kid="5c794df2-[redacted]-bad7-80ea989a99b6" commonEncryptionScheme="cbcs">
      ▼<cpix:Data>
        ▼<pskc:Secret>
           <pskc:PlainValue>u55fb[redacted]PCw==</pskc:PlainValue>
         </pskc:Secret>
       </cpix:Data>
     </cpix:ContentKey>
```

Option 1: Request DRM keys using EZDRM version 2 Web Service - CBCS

Call the EZDRM web service in a browser:
https://cpix.ezdrm.com/KeyGenerator/cpix2.aspx?k=kid&u=username&p=password&c=resourcename&EncryptionScheme=cbcs

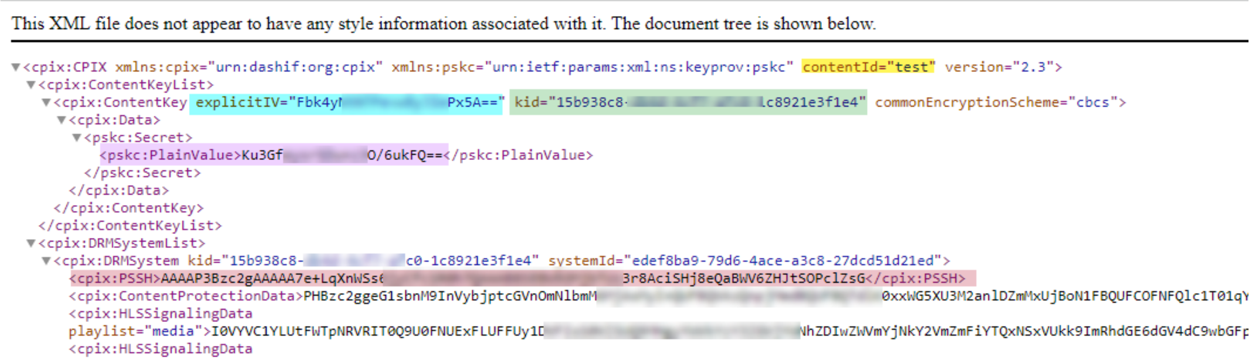
The parameters are as follows:

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
EncryptionScheme	Encryption using CBCS – PlayReady 4.3 Header

* To generate a GUID for the k value, you can use a GUID generator like the one found here: https://www.guidgenerator.com/.

Key Value Definitions

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.

```
▼<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
  ▼<cpix:ContentKeyList>
    ▼<cpix:ContentKey explicitIV="Fbk4y[redacted]Px5A==" kid="15b938c8-[redacted]c8921e3f1e4" commonEncryptionScheme="cbcs">
      ▼<cpix:Data>
        ▼<pskc:Secret>
           <pskc:PlainValue>Ku3Gf[redacted]O/6ukFQ==</pskc:PlainValue>
         </pskc:Secret>
       </cpix:Data>
     </cpix:ContentKey>
   </cpix:ContentKeyList>
  ▼<cpix:DRMSystemList>
    ▼<cpix:DRMSystem kid="15b938c8-[redacted]c0-1c8921e3f1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
       <cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs[redacted]3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH>
       <cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbm[redacted]0xxWG5XU3M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01q
       <cpix:HLSSignalingData
       playlist="media">I0VYVC1YLUtFWTpNRVRIT0Q9U0FNUExFLUFFUy1[redacted]NhZDIwZWVmYjNkY2VmZmFiYTQxNSxVUkk9ImRhdGE6dGV4dC9wbGF
       <cpix:HLSSignalingData
```

- **contentId** – c value returned, generic resource name/identifier (client generated)

- **explicitIV** – the Apple FairPlay explicit IV value

- **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: https://www.guidgenerator.com/.

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId="test" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="Fbk4yNtXXXXXXXXyJIePx5A==" kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" commonEncryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>Ku3GfeXXXXXXX+z3O/6ukFO==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="15b938c8-XXXX-4cf7-XXXX-1c89XXXXf1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
<cpix:PSSH>AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8eQaBWV6ZHJtSOPclZsG</cpix:PSSH>
<cpix:ContentProtectionData>PHBzc2ggeG1sbnM9InVybjptcGVnOmNlbmM6MjAxMyI+QUFBQVAzQnpjMmdBQUFBQTdlK0xxWG5XU3M2anlDZmMxUjBoN1FBQUFCOFNFQlc1T01qYlRVejXXXXXXXXXXjhlUWFCV1Y2WkhKdFNPUGNsWnNHPC9wc3NoPg==</cpix:ContentProtectionData>
```

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/cpix2.aspx?k=kid&u=username&p=password&c=test&EncryptionScheme=cbcs”
```

Widevine, PlayReady and Apple FairPlay Streaming CBCS/CMAF Encryption

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

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

```
python mp4-dash.py --widevine-header=”#PSSH” --encryption-key=KID:KeyHEX:IV --hls --encryption-cenc-scheme=cbcs --fairplay-key-uri="skd://kid:IV" --playready-header=LA_URL:”https://playready.ezdrm.com/cency/preauth.aspx?pX=EXXXXF” --playready-version=4.3 C:\2022\fragmented-bunny.mp4
```

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

- 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 --**encryption-key** to specify the EZDRM Key. Bento4 refers to encryption-key for the mode **fps** is **KID**:**KeyHEX**:**IV**

  For example:

  15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415:15b938c8db4d4cf7afc01c8921e3f1e4

  The **KID** is the EZDRM **kid** value (no dashes).

  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) = Ku3GfXXXXXXu+z3O/6ukFQ==

(KeyHEX) = 2aedXXXXecacXXXXefb3dceXXXXa415

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

explicitIV (Base 64) = Fbk4yNtNTPevwByJlePx5A==

IV (HEX no dashes) = 15b938c8db4d4cf7afc01c8921e3f1e4

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

```
--encryption-key=15b9XXXXdb4dXXXXafc01c89XXXXf1e4:2aedXXXXecacXXXXefb3dceXXXXa415:15b938c8db4d4cf7
afc01c8921e3f1e4
```

- **KeyURI** - Use the command line option **--fairplay-key-uri** to specify the license URL for encryption. Build by appending the base URL **"skd://"** with the **kid** value (with dashes) and the **IV (HEX no dashes)** seperated by a colon. For example:
  skd:// 582de60c-XXXX-XXXX-a013-XXX33cca5ef8: 15b938c8db4d4cf7afc01c8921e3f1e4

- 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.**

The description for the lines of syntax include:

- Use the command line option **--encryption-cenc-scheme=cbcs** to select the **CBCS** encryption mode.

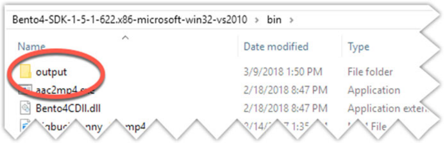
- Use the command line option **--hls**

- **video-source.mp4** should be replaced with the name of the source file.

Sample Command from the bin folder:

```
python mp4-dash.py --widevine-header=”#AAAAP3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7XXXXXXXXXX5OMjbTUz3r8AciSHj8e
OaBWV6ZHJtSOPclZsG”  --encryption-key=15b9XXXXdb4d4XXXXfc01c89XXXXf1e4:2aedXXXXecacXXXXeefb3dcXXXXba415:15
b9XXXXdb4dXXXXafc01c89XXXXf1e4 --hls --encryption-cenc-scheme=cbcs --fairplay-key-uri="skd://582de60c-XXXX
-XXXX-a013-XXX33cca5ef8:15b938c8db4d4cf7afc01c8921e3f1e4" --playready-header=LA_URL:”https://playready.ezd
rm.com/cency/preauth.aspx?pX=EXXXXF” --playready-version=4.3 C:\2022\fragmented-bunny.mp4
```

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



Bento4-SDK-1-5-1-622.x86-microsoft-win32-vs2010 › bin ›

Name	Date modified	Type
output	3/9/2018 1:50 PM	File folder
aac2mp4.exe	2/18/2018 8:47 PM	Application
Bento4CDll.dll	2/18/2018 8:47 PM	Application exten…

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

Additional Information

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