

# Ingesting and transcoding HDR h.265 (HEVC) content

## 1. Add HDR metadata to your production master object

Look at your master in the fabric browser and get the library ID, the object ID, and the name of the file within it that contains your video stream.

(Make sure you are looking at the **master** and not the **mezzanine**.)

Example info for a production master:

```
Library ID: ilib3j9xhdqvzkGvzk5cGSBsD4mk12TG
Object ID: iq__2WQzK4QUXnvZJ4rAJf2yQ2eg8WD
Filename: test_reel.mxf
```

Download and set up the **add\_hdr\_metadata** script:

```
git clone https://github.com/eluv-io/add_hdr_metadata
cd add_hdr_metadata
npm install
```

Set environment variables PRIVATE\_KEY and FABRIC\_CONFIG\_URL:

```
export PRIVATE_KEY=(your_fabric_private_key)
export FABRIC_CONFIG_URL="(the_fabric_config_url)"
```

Example values for FABRIC\_CONFIG\_URL:

Network	Config URL
Test	https://main.net955210.contentfabric.io/config
Demo	https://demov3.net955210.contentfabric.io/config
Production	https://main.net955305.contentfabric.io/config

Place your HDR information into a JSON file - if you wish, edit the presupplied `sample_hdr.json` file:

```
{
  "master_display": "G(13250,34500)B(7500,3000)R(34000,16000)WP(15635,16450)L(40000000,50)",
  "max_cll": "1514,172"
}
```

Execute the script:

```
node add_hdr_metadata.js LIBRARY_ID OBJECT_ID FILENAME JSON_FILE
```

example:

```
node add_hdr_metadata.js ilib3j9xhdqvzkGvzk5cGSBsD4mk12TG iq__2WQzK4QUXnvZJ4rAJf2yQ2eg8WD test_reel.mxf sample_hdr.json
```

Once the script finishes, it will print out the hash for the revised master object. You will need this has to create your mezzanine.

## 2. Create the mezzanine object using your desired profile

When you create your mezzanine, an ABR Profile JSON file specifies the desired encoding, bit depth, encoder preset, and bitrate or CRF (constant rate factor) to use. There are two sample profiles included with the **add\_hdr\_metadata.js** script:

**abr\_profile\_hdr\_4k\_crf16\_veryslow\_clear.json** uses CRF 16 to set mezzanine quality level.

**abr\_profile\_hdr\_4k\_50mbps\_veryslow\_clear.json** uses a target bitrate (50 megabits/sec) to set mezzanine quality level.

Both files specify a bit depth of 10 (to use HDR10) and encoder preset 'veryslow' (which produces higher quality at the expense of taking longer to prepare the mezzanine).

These files specify at 16:9 aspect ratio with 4k resolution (3840x2160). If your master has a different aspect ratio or resolution then you will need to adjust the following numbers in the file:

Line	Change
" {\"media_type\":\"video\", \"aspect_ratio_height\":9, \"aspect_ratio_width\":16}: {	<b>9 and 16</b> to new height / width of aspect ratio - use lowest common denominator whole numbers, e.g. 3 and 4, not 1 and 1.333 or 30 and 40
\"height\": <b>2160</b> ,	<b>2160</b> to the height of your master
\"width\": <b>3840</b>	<b>3840</b> to the width of your master

Once you have selected which approach to take (CRF or target bit rate) and have made any needed adjustments to the ABR Profile JSON file, you will then run the CreateABRMezzanine.js script to create the HDR mezzanine.

If you do not have elv-client-js installed already, do the following:

```
git clone https://github.com/eluv-io/elv-client-js
cd elv-client-js
npm install
```

Find the ID of your mezzanine library (click on it in the fabric browser, then click on the Library Info tab)

Find the ID of your title Content Type (click on Content Types in the fabric browser, find the one named "*YOUR\_COMPANY\_NAME* Title" or "ABR Master", click on it, and copy its Object ID)

Determine your fabric configuration URL from the "Example values for FABRIC\_CONFIG\_URL" table above.

If the source file(s) for your master are stored in S3, you will need to set the following environment variables:

```
export AWS_REGION=(the region)
export AWS_BUCKET=YOUR_BUCKET_NAME
export AWS_KEY=YOUR_AWS_KEY
export AWS_SECRET=YOUR_AWS_SECRET
```

If not still set, you will also need to set PRIVATE\_KEY:

```
export PRIVATE_KEY=(your_fabric_private_key)
```

Run the following:

```
node ./testScripts/CreateABRMezzanine.js \
--config-url "YOUR_FABRIC_CONFIG_URL" \
--title "YOUR_MEZZANINE_TITLE" \
--type YOUR_CONTENT_TYPE_ID \
--masterHash YOUR_MASTER_HASH_FROM_STEP_1 \
--library YOUR_MEZZANINE_LIBRARY_ID \
--abr-profile YOUR_ABR_PROFILE_JSON_FILE_NAME
```

Once the mezzanine transcode is started, the script will print out the object ID of your new mezzanine. Use this value to check on the status of your mezzanine:

```
node ./testScripts/MezzanineStatus.js \  
  --config-url "YOUR_FABRIC_CONFIG_URL" \  
  --objectId THE_NEW_MEZZANINE_OBJECT_ID
```

Once the mezzanine has completed, run the same command with the --finalize option:

```
node ./testScripts/MezzanineStatus.js \  
  --config-url "YOUR_FABRIC_CONFIG_URL" \  
  --objectId THE_NEW_MEZZANINE_OBJECT_ID --finalize
```

(It may take several minutes after the finalize command finishes for the object status to update in the fabric browser)