



# Dolby Vision Professional Tools

## User Manual

Version 4.0.0

February 2019

Confidential information

## Copyright

© 2019 Dolby Laboratories. All rights reserved. For information, contact:

Dolby Laboratories, Inc.

1275 Market Street

San Francisco, CA 94103-1410 USA Telephone 415-558-0200

Fax 415-863-1373

<http://www.dolby.com>

## Trademarks

Dolby and the double-D symbol are registered trademarks of Dolby Laboratories. Following are trademarks of Dolby Laboratories:

Dolby®

Dolby Atmos® Dolby Audio™ Dolby Cinema™ Dolby Theatre® Dolby Vision™ Dolby Voice®

Feel Every Dimension in Dolby™ Feel Every Dimension™

Feel Every Dimension in Dolby Atmos™ Dolby Digital Plus™

Dolby Advanced Audio™ Dolby Home Theater® Dialogue Intelligence™

Dolby Digital Plus Home Theater™ MLP Lossless™

Pro Logic® Surround EX™

Dolby Digital Plus Advanced Audio™ Dolby Fidelio™

Dolby AccessLink™ Dolby CaptiView™ Dolby CineAsset™

Dolby CineAsset Player™

All other trademarks remain the property of their respective owners.

## Patents

This product is protected by one or more patents in the United States and elsewhere. For more information, including a specific list of patents protecting this product, please visit <http://www.dolby.com/patents>.

## Confidential information

Confidential information for Dolby Laboratories Licensees only. Unauthorized use, sale, or duplication is prohibited.

## Third-party and open-source software declaration

Several open-source software packages are utilized by the product name. Each is individually covered by its own copyright and license, and they are acknowledged here.

ilmBase 2.1.0: <http://www.openexr.com>

OpenEXR 2.2.0 <http://www.openexr.com>

zlib 1.2.8: [zlib.net](http://zlib.net)

---

# Table of Contents

<b>1</b>	<b>INTRODUCTION</b> .....	<b>5</b>
<b>2</b>	<b>INSTALLING AND USING THE TOOLS</b> .....	<b>6</b>
2.1	INSTALLATION .....	6
2.2	USING THE TOOLS .....	6
<b>3</b>	<b>CM_OFFLINE</b> .....	<b>7</b>
3.1	USAGE.....	7
3.1.1	Command line syntax .....	7
3.1.2	Command line help .....	8
3.1.3	Compatible image formats .....	9
3.1.4	Basic option information.....	9
3.1.5	Advanced option information.....	11
3.1.6	Input and output image sequence syntax.....	12
3.1.7	Preset Targets .....	12
3.1.8	Custom Targets .....	12
3.1.9	Specifying frame ranges to process.....	13
3.1.10	Image Scaling.....	13
3.1.11	Metadata Validation.....	13
<b>4</b>	<b>MEZZINATOR</b> .....	<b>15</b>
4.1	USAGE.....	15
4.1.1	Command line syntax .....	15
4.1.2	Command line help .....	16
4.1.3	Compatible input image formats .....	16
4.1.4	Compatible output image formats .....	16
4.1.5	Basic option information.....	17
4.1.6	Advanced option information.....	18
4.1.7	Input image sequence syntax.....	19
4.1.8	Metadata and image sequence alignment .....	19
4.1.9	Trimming and alignment .....	19
4.1.10	Metadata Validation.....	20
<b>5</b>	<b>METAFIER</b> .....	<b>21</b>
5.1	USAGE.....	21
5.1.1	Command line syntax .....	21
5.1.2	Command line help .....	22
5.1.3	Basic option information.....	23
5.1.4	Advanced option information.....	24
5.1.5	Validating Metadata.....	25
<b>6</b>	<b>CM_ANALYZE</b> .....	<b>29</b>
6.1	REQUIRED INFORMATION AND FILES .....	29
6.2	USAGE.....	30
6.2.1	Command line syntax .....	30
6.2.2	Command line help .....	30
6.2.3	Basic usage option information.....	31
6.2.4	Advanced option information.....	32
6.2.5	Compatible input image formats .....	32
6.2.6	Using Live Mode Analysis.....	32

<b>7</b>	<b>MD_CONFORM</b> .....	<b>34</b>
7.1	CONFORMANCE WORKFLOW.....	34
7.2	USAGE.....	34
7.2.1	Command line syntax.....	34
7.2.2	Basic usage option information.....	35
<b>8</b>	<b>ADVANCED COLOR AND IMAGE FORMAT STRINGS (--HELP-FORMAT)</b> .....	<b>36</b>
	<b>APPENDIX A: MASTERING AND TARGET DISPLAYS FOR CM_OFFLINE AND/OR CM_ANALYZE</b> .....	<b>40</b>
	<b>APPENDIX B: CREATING A LONGPLAY DELIVERABLE FROM MULTIPLE SOURCES</b> ....	<b>41</b>
	<b>APPENDIX C: CREATING A PLAYLIST FILE</b> .....	<b>43</b>
	<b>APPENDIX D: COMMON INPUT AND OUTPUT IMAGE FORMATS EXAMPLES</b> .....	<b>44</b>
	<b>APPENDIX E: METADATA ALIGNMENT EXAMPLES</b> .....	<b>46</b>
	<b>APPENDIX F: SPECIFYING LETTERBOX</b> .....	<b>48</b>
	<b>APPENDIX G: CUSTOM TARGET XML</b> .....	<b>49</b>
	<b>APPENDIX H: OPERATING SYSTEM COMPATIBILITY</b> .....	<b>50</b>

# 1 Introduction

The Dolby Vision Professional Tools are a set of command line tools you use to create and manage a Dolby Vision master and mezzanine deliverable. Additionally, use these tools to extract and update the Dolby Vision metadata from a Dolby Vision deliverable.

Dolby Vision Professional Tools include:

- `cm_offline` - Renders, with color mapping, a Dolby Vision Master to a specified target display
- `mezzinator` - Creates a Dolby Vision mezzanine MXF deliverable
- `metafier` - Dolby Vision metadata utility and validation tool
- `cm_analyze` - Generates Dolby Vision XML Metadata with Global and L1 Analysis Metadata to convert existing HDR10 content to Dolby Vision content
- `md_conform` - Conforms L1 metadata in Dolby Vision bitstreams to reduce artifacts for certain Dolby Vision TVs

These tools are delivered as executables and designed to run on Linux, Mac OS X, or Windows systems. Choose the appropriate package for your operating system.

For support and questions, send a detailed email to: [dolbyvisionmastering@dolby.com](mailto:dolbyvisionmastering@dolby.com)

## 2 Installing and using the tools

### 2.1 Installation

The Dolby Vision Professional Tools are available for Linux, MacOS, and Windows and are provided in a separate installer package. Use the appropriate Professional Tools installer package for your operating system and follow the installation prompts.

By default, the installer packages will install the Professional Tools in the following locations:

#### Linux

```
[current_folder]/Dolby_Vision_Professional_Tools_v4.0.0
```

#### MacOS

##### Command Line Tools:

```
/usr/local/bin/dolby_vision_professional_tools/
```

##### User Manual :

```
/usr/local/share/doc/dolby/dolby_vision_professional_tools/
```

#### Windows

```
C:\Program Files\Dolby\Dolby_Vision_Professional_Tools_v4.0.0\
```

### 2.2 Using the tools

Use the built-in help function to view a list of each tool's options and their descriptions. At the command line, use `-h`, `--help`, `--help-advanced` or `-?` to access help. Each tool has a unique set of options and arguments. We recommend that you refer to the options list for each tool and use this document for informational purposes.

To use each command line tool, enter the tool name at the command line. These tools do not require any additional programs to run.

## 3 CM\_offline

This tool maps a Dolby Vision Master to a specified target display and outputs the appropriate derivative standalone image sequence.

If CUDA is supported on the platform, the `cm_offline*` tool defaults to using CUDA for processing because this is usually faster processing than using the CPU. For CUDA support, the machine is required to have a NVIDIA graphics card. CUDA acceleration only supports a single card. If there are multiple GPUs on the machine, this tool will determine the fastest single GPU card and use its available cores for processing. Multiple GPU cards is not currently supported. To force the system to use CPU processing, use the `--cpu-only` option in the command line.

**NOTE:** In this release, the `cm_offline` tool is only CPU when rendering CMv4 deliverables. CUDA support for CMv4 in `cm_offline` will be supported in a future release.

### 3.1 Usage

When specifying combined options with space delimited values, use quotes (“”) at the beginning and end of the specified string.

See [Advanced Color and Image Format Strings](#) for a list all available functions that are described in the output of the `./cm_offline --help-format` option.

See [Appendix D: Common input and output image format examples](#) for common image format examples.

#### 3.1.1 Command line syntax

Basic usage

```
./cm_offline [-m <arg>] [--reel <arg>] [-f] [-r <arg>] [--targ <arg>]
[--custom-target <arg>] [--targ-info <arg>] [--frame-offset <arg>] [--
output-format <arg>] [--show-targets] [--j2k-profile <arg>] [--
quicktime-codec <arg>] [--help-format] [--help-advanced] [-h] [-v]
<source> <output>
```

Advanced usage

```
./cm_offline [-m <arg>] [--reel <arg>] [-f] [-r <arg>] [--input-crop
<arg1> <arg2> <arg3> <arg4>] [--resize-filter <arg>] [--resize <arg1>
<arg2>] [--output-letterbox <arg1> <arg2> <arg3> <arg4>] [--output-
image-ar <arg>] [--output-canvas-ar <arg>] [--targ <arg>] [--custom-
target <arg>] [--targ-info <arg>] [--frame-offset <arg>] [--source-
format <arg>] [--output-format <arg>] [--show-targets] [--echo-
metadata] [--disable-validation] [-t <arg>] [--j2k-profile <arg>] [--
quicktime-codec <arg>] [--help-format] [--help-advanced] [-h] [-v] [--
disable-font-color] <source> <output>
```

### Common examples

To deliver a standalone deliverable that is derived from the Dolby Vision Master or Mezzanine, that is, standalone mapped SDR:

```
./cm_offline -m <metadata.xml> --targ 1  
<input_image_sequence_#####.tif> <output_image_sequence_#####.tif>  
./cm_offline --targ 1 <input.mxf> <output_image_sequence_#####.ext>
```

When metadata frame start is not aligned with image frame start:

```
./cm_offline -m <metadata.xml> --targ <arg> --frame-offset <-86400>  
<input_image_sequence_#####.tif> <output_image_sequence_#####.tif>
```

To deliver specific frame ranges from the Dolby Vision Master:

```
./cm_offline -m <metadata.xml> --targ <arg>  
<input_image_sequence_#####.tif> <output_image_sequence_#####.tif>  
-f 86400-86500,95000-95100,100000-100100
```

To output a 1920x1080 SDR from a 3840x2160 master:

```
./cm_offline -m <metadata.xml> --targ 1  
<input_image_sequence_#####.tif> <output_image_sequence_#####.tif> -  
-output-format "1920x1080"
```

**NOTE:** For more information on resizing, refer to [Using the crop, resize and letterbox options](#)

To deliver a 1k-nit Rec.2020 from a 4k-nit P3D65 Master with P3 Trims:

```
./cm_offline -m <metadata.xml> --targ 48  
<input_image_sequence_#####.tif> <output_image_sequence_#####.tif>  
--output-format "pq bt2020"
```

### 3.1.2 Command line help

View basic options by entering:

```
./cm_offline -h | -? | --help
```

View the advanced options by entering:

```
./cm_offline --help-advanced
```



### 3.1.3 Compatible image formats

The `cm_offline` tool is compatible with multiple image formats. To output a selected image format, use one of the available image format extensions at the end of the output filename.

- MXF (.mxf)
- TIFF (.tif)
- OpenEXR (.exr)
- JPEG 2000 (.j2c/.j2k)
- DPX (.dpx)
- ProRes (.mov)

By default, the output format is the same bit-depth as the source format. The `cm_offline` tool also has the capability to change the output image formats and resolution.

To output the images in the desired image format in the same bit-depth as the input, specify one of the compatible image formats in the output file name and the tool outputs the correct image.

The `cm_offline` tools support many image manipulation operations such as image format conversion, image resizing, packing, sampling and encoding.

### 3.1.4 Basic option information

Option Command	Description	Default
<code>-m --metadata &lt;arg&gt;</code>	Input XML Metadata filename	
<code>--reel &lt;arg&gt;</code>	Input Reel TXT information filename	
<code>-f --frames &lt;arg1-arg2&gt;, &lt;arg3-arg4&gt;...</code>	Specifies the frame ranges to process. For multiple frame ranges, use <code>&lt;frame_start#-frame_end#&gt;</code> , <code>&lt;frame_start#-frame_end#&gt;</code>	all
<code>-r --frame-rate &lt;arg&gt;</code>	Sets the frame rate tag for the XML or MXF (24, 23, 24000/1001...)	input
<code>--targ &lt;arg&gt;</code>	Specifies the output target to emulate (ID or name from <code>--show-targets</code> )	1
<code>--custom-target &lt;xml&gt;   &lt;string&gt;</code>	Use a custom target for rendering. Specify a custom target XML or enter the parameters manually: <code>&lt;eof&gt;&lt;minbrightness&gt;,&lt;peakbrightness&gt; &lt;chromaName customChroma&gt;</code>	

<code>--targ-info &lt;arg&gt;</code>	Displays the detailed information for the specified output target ID	false
<code>--frame-offset &lt;arg&gt;</code>	Offsets the metadata start frame number by this frame count to match the start image frame number.  Use <code>auto</code> to guess the frame offset.  Use <code>trim</code> to guess the frame offset and trim the head.	0
<code>--output-format &lt;arg&gt;</code>	Optionally specifies a custom format for the output images.	input
<code>--show-targets &lt;file.xml&gt;</code>	Used in conjunction with XML file and displays a list of all the available target profiles.	false
<code>--j2k-profile &lt;arg&gt;</code>	Sets the JPEG 2000 encoding profile for MXF output (bcp5, bcp7, dci, imf_ml3, imf_ml6, imf_ml7, imf_ml8, imf_lossy_ml3, imf_lossy_ml6, imf_lossy_ml7, imf_lossy_ml8) NOTE: The default is bcp5 for bit depth between 8 and 12-bits or imf_ml6 for 16-bits	
<code>--quicktime-codec &lt;arg&gt;</code>	Sets the Codec for Quicktime MOV output (prores_422_hq, prores_422, prores_422_lt, prores_422_proxy, prores_4444, prores_4444_xq.)	ProRes 4444XQ
<code>--help-format</code>	Displays extended help on specifying image format strings for source, target and color encoding.	
<code>--help-advanced</code>	Displays advanced application arguments and usage.	
<code>-h -? --help</code>	Displays basic application arguments and usage.	
<code>-v --version</code>	Displays program version.	

---

**Advanced option information**

Option Command	Description	Default
<code>--input-crop &lt;arg1&gt; &lt;arg2&gt; &lt;arg3&gt; &lt;arg4&gt;</code>	Crops the original frame before processing (4 integers: left right top bottom).	
<code>--resize &lt;arg1&gt; &lt;arg2&gt;</code>	Resizes the output to the specified image resolution (2 integers: width height).	
<code>--resize-filter &lt;arg&gt;</code>	Specifies the resize filter (mitchell, lanczos7, lanczos5, lanczos3, lanczos2, gaussian, triangle, box, dirac).	mitchell
<code>--output-letterbox &lt;arg1&gt; &lt;arg2&gt; &lt;arg3&gt; &lt;arg4&gt;</code>	Sets the active image aspect ratio using a letterbox instead of a ratio (4 integers: left right top bottom) and pads if necessary with black color.	
<code>--output-image-ar &lt;arg&gt;</code>	Sets the active image aspect ratio for the output image after resizing.	
<code>--output-canvas-ar &lt;arg&gt;</code>	Sets the output canvas image aspect ratio for the output image after resizing.	
<code>--echo-metadata</code>	Displays metadata values on-screen as the image is processed.	
<code>--source-format &lt;arg&gt;</code>	Optionally specifies the source image format (use <code>--help-formats</code> for detailed information).	
<code>--disable-validation</code>	Skip Metadata validation	false
<code>-t --threads &lt;arg&gt;</code>	Specifies the number of CPU threads to use (-1 indicates all available cores).	-1
<code>--cpu-only</code>	Forces CPU processing only when CUDA is available (default: CUDA if available)	
<code>--cuda-device</code>	Uses the best available single GPU device	default
<code>--disable-font-color</code>	Disable Log Font Color	false

### 3.1.5 Input and output image sequence syntax

When specifying the image sequence for the input and output, replace the sequence number with the pound sign (#), as shown in the following example:

```
/path/name_#####.ext
```

If the image frame sequence number has varying length, you can use \*#

```
/path/name_*.ext
```

**Note:** Do not use a space character.

### 3.1.6 Preset targets

For convenience, the `cm_offline` tool contains a set of commonly used targets (as seen in `--show-targets`). To use these presets, use the `--targ <arg>` argument. See [Appendix A: Target displays for cm\\_offline](#) for details, or enter:

```
./cm_offline --show-targets
```

### 3.1.7 Custom targets

Custom targets can only be used on CMv4 projects. CMv2.9 does not support custom targets.

To specify a custom target that is not stored in the preset targets, create a 'custom target XML' or enter the specifications as a string of values. To create a custom target XML, see [Appendix G: Custom Target XML](#) for the Custom Target XML specification and example.

To specify a custom target without using a XML file, the following string should be entered:

```
--custom-target "  
<eof><minbrightness>,<peakbrightness> <chromaName|customChroma> "
```

Valid values for each target parameter need to be referenced from image format strings as described in [Advanced color and image format strings](#).

It is recommended to use a custom target XML file instead of writing the values in the command line to avoid typos. The XML files can then be cataloged and a library of custom targets can be built to support multiple projects easily.

#### Examples

##### TIFF example

```
./cm_offline -m <file.xml> <source_*.img> <output_#####.tif> --  
target-format "1920x1080x3 u16 interleaved 444 le tight 24fps pq rgb  
computer p3d65" --custom-target "pq(0.0001,800) p3d65"
```

##### ProRes example

```
./cm_offline -m <file.xml> <source_*.img> <output.mov> --target-  
format "1920x1080x3 u10 interleaved 422 le lsb32rev 24fps top_left pq  
ycbcr_bt2020 video bt2020" --quicktime-codec prores_422_hq --custom-  
target " pq(0.0001,800) bt2020"
```

### 3.1.8 Specifying frame ranges to process

Single or multiple groups of frame ranges to be processed are supported by using the `-f --frames` option. Groups of frame ranges can be specified as:

```
-f <frame_start#-frame_end#>,<frame_start#-frame_end#>...
```

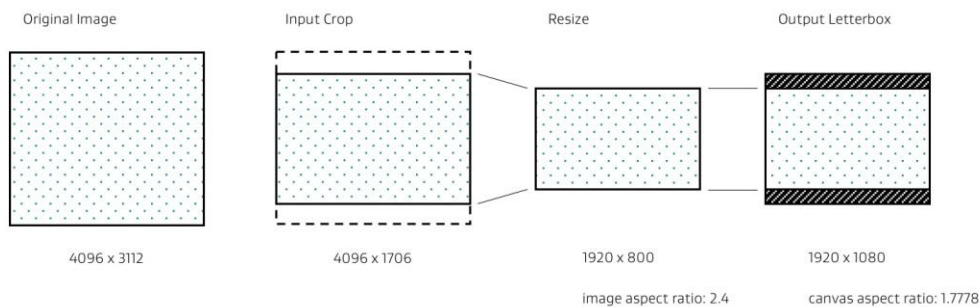
### 3.1.9 Image Scaling

The order of operations for scaling the output image is fixed as: Crop → Resize → Pad. This occurs regardless of the written order in the command line.

For automatic image scaling, use the `--output-format "<horizontal>x<vertical>x3"` option. Using the output-format option will automatically perform the crop, resize and pad based on the input resolution and the specified output resolution.

For advanced control over scaling, crop, resize and pad can be specified manually as shown below. When using the advanced controls, crop, resize and letterbox must all be specified in the command line. Specifying only one function will not work.

For help with this function, use `./cm_offline --help-advanced`



To specify Crop:

```
--input-crop <left_column> <right_column> <top_row> <bottom_row>
```

To specify Resize:

```
--resize <width> <height>
```

You can optionally apply resizing filters while resizing the output image:

```
--resize-filter <filter_name>
```

To specify Output letterbox:

```
--output-letterbox <left_column> <right_column> <top_row>
<bottom_row>
```

### 3.1.10 Metadata validation

When creating a deliverable with `cm_offline`, the metadata is automatically validated. If the metadata contains errors, the errors are displayed, and the images will not be created. The metadata must be fixed using the `./metafier` tool before `cm_offline` continues to process.

To bypass metadata validation checking, use `--disable-validation`. While this bypasses the validation check, corrupt or incorrect metadata may still cause `./cm_offline` to error and not produce a deliverable.

Refer to [Validating metadata](#) section in this document.

## 4 Mezzinator

This tool creates a type of Dolby Vision Mezzanine MXF deliverable that contains the video essence with the interleaved Dolby Vision Metadata track or a MXF with sidecar XML metadata file. This tool does not support any other mezzanine output.

### 4.1 Usage

When specifying combined options with space delimited values, use quotes (“”) at the beginning and end of the specified string.

See [Advanced Color and Image Format Strings](#) for a list all available functions that are described in the output of the `./mezzinator --help-format` option.

See [Appendix D: Common input and output image format examples](#) for common image format examples.

#### 4.1.1 Command line syntax

Basic usage

```
./mezzinator [-m <arg>] [--reel <arg>] [--frame-offset <arg>] [--frame-align <arg>] [--uuid <arg>] [--j2k-profile <arg>] [-r <arg>] [-f] [-b] [--target-bit-depth <arg>] [--disable-passthrough] [--sidecar-xml <arg>] [--help-format] [--help-advanced] [-h] [-v] <input> <output>
```

Advanced usage

```
./mezzinator [-m <arg>] [--reel <arg>] [--frame-offset <arg>] [--frame-align <arg>] [--uuid <arg>] [--j2k-profile <arg>] [-r <arg>] [--source-format <arg>] [-f] [-t <arg>] [-b] [--target-bit-depth <arg>] [--disable-passthrough] [--sidecar-xml <arg>] [--disable-validation] [--disable-font-color] [--help-format] [--help-advanced] [-h] [-v] <input> <output>
```

#### Common examples

Create a standard Dolby Vision MXF from a Dolby Vision Master:

```
./mezzinator -m <metadata.xml> <input_image_sequence_#####.ext> <output.mxf>
```

Create a standard MXF with sidecar XML metadata file:

```
./mezzinator -m <metadata.xml> <input_image_sequence_#####.tif> <output.mxf> --sidecar-xml <output.xml>
```

Create a longplay MXF from multiple reels:

```
./mezzinator --reel <reel.txt> <playlist.play> <output.mxf>
```

**NOTE:** For more information on creating a reel and playlist file, refer to [Appendix B: Creating a longplay deliverable from multiple sources](#) and [Appendix C: Creating a playlist file](#)

### 4.1.2 Command line help

View basic options by entering:

```
./mezzinator -h | -? | --help
```

View the advanced options by entering:

```
./mezzinator --help-advanced
```

### 4.1.3 Compatible input image formats

The mezzinator tool is compatible with the following input image formats:

- MXF (.mxf)
- TIFF (.tif)
- OpenEXR (.exr)
- JPEG 2000 (.j2c/.j2k)
- DPX (.dpx)
- ProRes (.mov)

### 4.1.4 Compatible output image formats

By default, the output format complies with IMF Application #2 Extended and output 12-bit J2C RGB or YCbCr video essence, with the interleaved Dolby Vision metadata within the MXF.

You can select the JPEG 2000 compression profile and select whether to convert the image essence to BT.2020 color space.

If the input is already a MXF with JPEG 2000 images essence, then the mezzinator just passes the image essence through without applying any processing on the images.



### 4.1.5 Basic option information

Option Command	Description	Default
<code>-m --metadata &lt;arg&gt;</code>	Specifies the input XML Metadata filename	
<code>--reel &lt;arg&gt;</code>	Specifies a reel file as the input source (.txt, .reel)	
<code>--frame-offset &lt;arg&gt;</code>	Offsets the metadata start frame number by this frame count to match the start image frame number  Use <code>auto</code> to guess the frame offset.  Use <code>trim</code> to guess the frame offset and trim the head.	0
<code>--frame-align &lt;arg&gt;</code>	Aligns the metadata to the specified image frame number in the sequence, such as 86400.	0
<code>--uuid &lt;arg&gt;</code>	Sets the MXF Asset UUID. The UUID string to set MXF UUID or write in form of:  81a82fd2-d1b7-11e4-8acc-002590ec378e  This field is either manually set with the input option or randomly generated based on input such as time of day or system information.	random
<code>--j2k-profile &lt;arg&gt;</code>	Sets the JPEG 2000 encoding profile for MXF output (bcp5, bcp7, dci, imf_ml3, imf_ml6, imf_ml7, imf_ml8, imf_lossy_ml3, imf_lossy_ml6, imf_lossy_ml7, imf_lossy_ml8)  <b>Note:</b> The default is bcp5 for bit depth between 8 and 12-bits or imf_ml6 for 16-bits	bcp5/imf
<code>-r --frame-rate &lt;arg&gt;</code>	Sets the frame rate tag for the XML or MXF (24, 23, 24000/1001...)	Input frame rate
<code>-f --frames &lt;arg1-arg2&gt;,&lt;arg3-arg4&gt;...</code>	Specifies the frame range(s) to process. For multiple frame ranges, use <frame_start#-frame_end#>,<frame_start#-frame_end#>	all
<code>--b --bt2020</code>	Converts image encoding primaries and white point from P3D65 or supplied source format to BT2020	

<code>--target-bit-depth &lt;arg&gt;</code>	Converts image to specified bit depth (8, 10, 12, 16)	12-bit
<code>--disable-passthrough</code>	Pass-through mode is only used when going from JPEG 2000 (.j2c or .j2k) to MXF (.mxf) or MXF to MXF	
<code>--sidecar &lt;arg&gt;</code>	Saves a sidecar XML file when creating a MXF with embedded metadata	false
<code>--help-format</code>	Displays extended help on specifying image format strings for source, target and color encoding	
<code>--help-advanced</code>	Displays advanced application arguments and usage	
<code>-h -? --help</code>	Displays basic application arguments and usage	
<code>-v --version</code>	Displays program version	

#### 4.1.6 Advanced option information

Option Command	Description	Default
<code>--t threads &lt;arg&gt;</code>	Specify number of CPU threads to use (-1 means all available cores)	-1
<code>--source-format &lt;arg&gt;</code>	Optionally specify the source image format (use --help-formats for detailed information)	
<code>--sidecar-xml</code>	Saves a sidecar XML file and create a MXF without embedded metadata	
<code>--disable-validation</code>	Skip Metadata validation	false
<code>--disable-font-color</code>	Disable Logging Font Color	false

### 4.1.7 Input image sequence syntax

When specifying the image sequence for the input and output, replace the sequence number with the pound sign (#), as shown in the following example:

```
/path/name_#####.ext
```

If the image frame sequence number has varying length, you can use `*#`

```
/path/name_*.ext
```

**Note:** Do not use a space character.

### 4.1.8 Metadata and image sequence alignment

When using the mezzinator tool, make sure the metadata in the XML is aligned with the frame sequence. There are multiple ways to align the metadata with the image sequence.

- Option 1: aligning metadata to image frame number
  - Use the `--frame-align` method to align the metadata to the specified image sequence start frame number that is used in the image filename, such as 86400.
- Option 2: using a frame offset number to align metadata to image sequence
  - Calculate the difference between the metadata start frame number and the image start frame number and use that number in the `--frame-offset` function.
- Option 3: attempt auto alignment of metadata and image sequence
  - Use the `--frame-offset auto` feature to attempt to automatically align the metadata to the image sequence.

See [Appendix E: Metadata alignment examples](#) for further information.

### 4.1.9 Trimming and alignment

To trim the head/tail of the image sequence, there are multiple methods that will trim, depending on your needs.

If you are combining multiple reels and each reel contains a head and tail, the easiest method is to create a reel and playlist file and use those as the input to `cm_offline`. For more information on creating a reel and playlist, refer to [Appendix B: Creating a custom reel.txt file from multiple sources](#) and [Appendix C: Creating a playlist file](#).

To automatically align and trim the head for a single reel, use the `--frame-offset trim` function to automatically align and trim the head. Here, you do not need to specify a frame number or offset. The trim function will automatically trim the head with the best-guess frame offset.

To manually trim a single reel, you can use the `--frame-offset` with the `--f <arg1-arg2>` functions. When specifying the frames, `'-f'`, this is always specified as the image sequence frame numbers.

#### **4.1.10 Metadata validation**

When creating a Dolby Vision MXF, the metadata is automatically checked. If the metadata contains errors, the errors will be displayed and the MXF will not be created. The metadata will need to be fixed before a MXF can be created.

Metadata can be fixed either manually and/or with the metafier tool.

To bypass metadata validation checking, use `--disable-validation`. While this bypasses the validation check, corrupt or incorrect metadata may still cause `./cm_offline` to error and not produce a deliverable.

Refer to the [Validating metadata](#) section in this document.

## 5 Metafier

Metafier is a metadata utility tool that works with Dolby Vision XML metadata and with the Dolby Vision Mezzanine MXF file. Metafier supports extracting, upgrading, checking, validating metadata and other useful functions and displaying information.

### 5.1 Usage

When specifying combined options with space delimited values, use quotes (“”) at the beginning and end of the specified string.

#### 5.1.1 Command line syntax

##### Basic usage

```
./metafier [-o <arg>] [-e <arg>] [--shot-list <arg>] [--shot-list-duration <arg>] [--show-info <arg>] [--show-uuid <arg>] [--reset-displays] [--show-mastering-displays] [--mastering-display <arg>] [--color-encoding <arg>] [--max-fall-cll <arg1> <arg2>] [--aspect-ratios <arg1> <arg2>] [-r <arg>] [--trim-head <arg>] [--trim-tail <arg>] [--insert-black-head <arg>] [--insert-black-tail <arg>] [--start <arg>] [--algorithm-version <arg1> <arg2>] [--validate] [--with-l2-lift] [--save-version <arg>] [-v] [--help-format] [--help-advanced] [-h] <input>
```

##### Advanced usage

```
./metafier [-o <arg>] [-e <arg>] [--force-invalid] [--disable-validation] [--validate-schema <arg>] [--shot-list <arg>] [--shot-list-duration <arg>] [--show-info <arg>] [--show-uuid <arg>] [--reset-displays] [--show-mastering-displays] [--mastering-display <arg>] [--color-encoding <arg>] [--max-fall-cll <arg1> <arg2>] [--aspect-ratios <arg1> <arg2>] [-r <arg>] [--trim-head <arg>] [--trim-tail <arg>] [--insert-black-head <arg>] [--insert-black-tail <arg>] [--start <arg>] [--remove-trim <arg1> <arg2>] [--algorithm-version <arg1> <arg2>] [--validate] [--with-l2-lift] [--fix-dissolves] [--remove-duplicates] [--save-version <arg>] [--swap-gain-gamma] [-v] [--help-format] [--disable-font-color] [--help-advanced] [-h] <input>
```

##### Common examples

Extracting a XML metadata file from an MXF:

```
./metafier -e <metadata.xml> <input.mxf>
```

Validating XML metadata or metadata in a MXF:

```
./metafier --validate <input.xml> or <input.mxf>
```

```
./metafier --validate -o <output_log.txt> <input.xml>
```

```
./metafier --validate --with-l2-lift <input.xml>
```

Upgrading an older version of XML metadata to the current version:

```
./metafier -o <new_metadata.xml> --save-version latest
```

Display the shot list from a XML or MXF:

```
./metafier --shot-list - <input.xml> or <input.mxf>  
./metafier --shot-list <shotlist.txt> <input.xml> or <input.mxf>
```

Display the shot list with shot duration from a XML or MXF:

```
./metafier --shot-list-duration - <input.xml> or <input.mxf>  
./metafier --shot-list-duration <shotlist.txt> <input.xml> or  
<input.mxf>
```

Using metafier to fix simple overlapping shots:

```
./metafier --fix-dissolves <input.xml>
```

**NOTE:** Using metafier does not fix all cases with overlapping shots but is worth trying before manually fixing the shots with overlapping metadata.

### **5.1.2 Command line help**

View command options by entering:

```
./metafier -h, or -help
```

View the advanced options by entering:

```
./mezzinator --help-advanced
```

### 5.1.3 Basic option information

Option Command	Description	Default
<code>-o --output &lt;arg&gt;</code>	Output filename (.xml, .mxf, .txt, .sl)	
<code>-e --extract &lt;arg&gt;</code>	Specifies the XML filename of the extracted metadata from a MXF	
<code>--shot-list &lt;arg&gt;</code>	Creates a shot list and saves it to a .sl or .txt file	
<code>--shot-list-duration &lt;arg&gt;</code>	Creates a shot list with duration and saves it to a .sl or .txt file	
<code>--show-info &lt;arg&gt;</code>	Displays frame size, frame duration and frame rate of the input MXF. Use the dash (-) for stdout.	false
<code>--show-uuid &lt;arg&gt;</code>	Displays the UUID of the input MXF. Use the dash (-) for stdout.	false
<code>--reset-displays</code>	Resets Mastering and Target displays specified in the metadata to default values. If it cannot identify the original Mastering Display, it will set it to: "4000-nit, P3, D65, ST. 2084, Full Range"	false
<code>--show-mastering-displays</code>	Displays a list of supported mastering displays	false
<code>--mastering-display &lt;arg&gt;</code>	Sets the Mastering Display by ID when using the '--reset-displays' flag (Overwrites any default behavior)	
<code>--color-encoding &lt;arg&gt;</code>	Sets the color encoding information	input
<code>--max-fall-cll &lt;arg1&gt; &lt;arg2&gt;</code>	Sets the specified MaxFALL and MaxCLL values (does not calculate values)	0 0
<code>--aspect-ratios &lt;arg1&gt; &lt;arg2&gt;</code>	Sets the aspect ratios (Canvas, Image)	input
<code>-r --frame-rate &lt;arg&gt;</code>	Sets the frame rate tag for the XML or MXF (24, 23, 24000/1001...)	input
<code>--trim-head &lt;arg&gt;</code>	Trims the head of the metadata up to the specified frame number (default: 0)	0

<code>--trim-tail &lt;arg&gt;</code>	Trims the tail of the metadata by the number of specified number of frames	0
<code>--insert-black-head &lt;arg&gt;</code>	Inserts the specified number of black frame metadata for the header	0
<code>--insert-black-tail &lt;arg&gt;</code>	Inserts the specified number of black frame metadata for the tail	0
<code>--start &lt;arg&gt;</code>	Specifies the starting metadata frame number (only works with XML Metadata and not with MXF)	0
<code>--validate</code>	Validates the XML or MXF metadata and displays the results	false
<code>--with-l2-lift</code>	Adds additional check for positive L2 lift values when used with <code>--validate</code> for letterbox content	
<code>--save-version &lt;arg&gt;</code>	Specifies the output Metadata version (input XML/MXF) options: 2.0.5, 4.0.2, latest (v2.0.5 for CMv2.9, v4.0.2 for CMv4.0) NOTE: This function will not upgrade v2.0.5 to v4.0.2	input
<code>-v --version</code>	Displays program version	
<code>--help-format</code>	Show extended help on specifying image format strings	
<code>--help-advanced</code>	Displays advanced application arguments and usage	
<code>-h -? --help</code>	Displays basic application arguments and usage	

#### 5.1.4 Advanced option information

Option Command	Description	Default
<code>--force-invalid</code>	In case of invalid metadata, ignores the errors and continues anyway	false
<code>--disable-validation</code>	Skip Metadata validation	



<code>--validate-schema &lt;arg&gt;</code>	Validate XML schema (XML version). Options are: 4.0.2 NOTE: only v4.0.2 and higher XML version may be validated	
<code>--remove-trim &lt;arg1&gt; &lt;arg2&gt;</code>	Remove a trim from the Metadata (Trim Level, Target ID) Options for Trim Level: L2 or L8	false
<code>--fix-dissolves</code>	Attempts to introduce dissolves to fix overlapping shots	false
<code>--remove-duplicates</code>	Attempts to remove any duplicate trim levels found in the metadata	false
<code>--algorithm-version &lt;arg1&gt; &lt;arg2&gt;</code>	Set the Algorithm Version (L1 Version, L2 Version)	empty
<code>--swap-gain-gamma</code>	Swaps the gain and gamma values for old baselight 2.x metadata files (this will also update the metadata to the current version supported by this application)	false
<code>--disable-font-color</code>	Disable Logging Font Color	false

### 5.1.5 Validating metadata

The metafier tool has a feature that can perform basic metadata testing for validating Dolby Vision metadata in XML or MXF. To use the validation feature, use the `--validate` argument with metafier followed by the file.

```
./metafier --validate <md_file.xml>|<file.mxf>
./metafier --validate <reel.txt>
./metafier --validate </fullpath/>
```

The metafier tool checks the metadata for the following information:

- Mastering monitor
  - A single mastering monitor is listed in the metadata
  - The mastering monitor has valid specifications in the metadata
- Target displays
  - Target displays in the metadata are valid
- Color encoding
  - If present, the color encoding information is sanity checked and displayed

- Checks for overlapping shots and displays the shots that have overlap
- Checks for any gaps in metadata and displays the pair of shots that has the gap
- Checks each shot for negative duration
- Validity checks for L1 and L2 are valid ranges (-1,1)
- L1 values are compliant with:  $\text{min} \leq \text{mid} \leq \text{max}$
- Checks for same L1 values across consecutive shots and will output a warning
- Outputs the following useful information:
  - Frame Range
  - Aspect Ratio
  - XML Version number
  - Number of shots in the XML
  - Frame Range of entire XML
  - TargetID with trim count
  - Color Encoding values in XML
- Checks for trims with positive L2 lift values:
  - `--validate --with-l2-lift` will check and report shots that contain lift-trim values greater than zero if the aspect ratio, `canvas ≠ image`. If the `canvas=image`, then using `--with-l2-lift` will not perform the check

When using the metafier tool to validate a reel file, the metafier tool will validate each XML as described above as well as cross compare each XML file specified in the reel file. If metadata mismatches one or more specified XML files, then the metafier tool will summarize what is mismatching per XML file. Cross-comparing XML files is only performed when using a reel file as the input.

When validating all the XML files in a folder, the metafier tool will validate each XML individually and display the validation report.

#### 5.1.5.1 Validation example for XML v2.0.5

```
$ ./metafier --validate <file.xml>
01/29/2019/10:37:11.407652027 Metadata: INFO Validation Summary:
=====
Metadata version: "2.0.5"
Aspect Ratios (Canvas - Image): 1.77778 - 1.77778
Frame Rate: 24fps
Mastering Monitor: 4000-nit, P3, D65, ST.2084, Full (ID 7): Bit Depth:
16bit - Diagonal: 42in - Color Info: "pq(0.005,4000) rgb computer
p3d65"
Mastering Monitor Validation Test: PASS
Target Displays Validation Test: PASS
Level6 (MaxFALL - MaxCLL): 0 - 0
Color Encoding: "u16 444 pq(0,10000) rgb computer p3d65"
Color Encoding Validation Test: PASS
```

Overlapping Shots Validation Test: **PASS**  
Gap between Shots Validation Test: **PASS**  
Negative Shot duration Validation Test: **PASS**  
Number of Shots: 2  
Frame Range: 86400-86455  
L2 Trim Count (TargetID, count): (1, 26)  
L1 Metadata Validation Test: **PASS**  
L2 Metadata Validation Test: **PASS**  
=====

01/29/2019/10:37:11.407891643 metafier: **INFO** No issues found in the Metadata

### 5.1.5.2 Validation example for XML v4.0.2

```
$ ./metafier --validate <file.xml>
01/29/2019/10:33:19.170772029 Metadata: INFO Validation Summary:
=====
Metadata version: "4.0.2"
Aspect Ratios (Canvas - Image): 1.77778 - 1.77778
Frame Rate: 24fps
Mastering Monitor: 4000-nit, P3, D65, ST.2084, Full (ID 7): Bit Depth:
16bit - Diagonal: 42in - Color Info: "pq(0.005,4000) rgb computer
p3d65"
Mastering Monitor Validation Test: PASS
Target Displays Validation Test: PASS
Level6 (MaxFALL - MaxCLL): 0 - 0
Color Encoding: "u16 444 pq(0,10000) rgb computer p3d65"
Color Encoding Validation Test: PASS
Overlapping Shots Validation Test: PASS
Gap between Shots Validation Test: PASS
Negative Shot duration Validation Test: PASS
Number of Shots: 2
Frame Range: 86400-86455
L2 Trim Count (TargetID, count): (1, 26)
L8 Trim Count (TargetID, count): (1, 26)
L1 Metadata Validation Test: PASS
L2 Metadata Validation Test: PASS
L3 Metadata Validation Test: PASS
L8 Metadata Validation Test: PASS
L9 Metadata Validation Test: PASS
=====
```

01/29/2019/10:33:19.171177404 metafier: **INFO** No issues found in the Metadata

### 5.1.5.3 Example with gaps between shots example

If shots are not contiguous or overlapping, the pair of shots, or multiple shots with errors, will be displayed that contains the error.

```
12/16/2016/12:06:47.879302843 metadata: ERROR Gap between Shots:  
UUID = b850de18-92c0-4be7-9049-49ac722203bb @{86441+29@24/1}  
UUID = c139651d-e544-47c8-a443-a071f7a24e2f @{86471+19@24/1}  
UUID = b14bb736-6d8d-4345-8669-2db52a11260f @{86610+18@24/1}  
UUID = 6abdd0b0-e3ab-410d-890b-6f32878b35a4 @{86629+38@24/1}  
12/16/2016/12:06:47.879564730 metafier: ERROR Invalid Metadata
```

## 6 CM\_analyze

The `cm_analyze` tool is primarily converts existing HDR10 content to Dolby Vision content by automatically generating Dolby Vision L1 Analysis metadata for the HDR10 source. This metadata and original HDR10 source is then used to create a Dolby Vision Mezzanine deliverable to be encoded and distributed.

This tool is not intended for generating a backward-compatible SDR version. `cm_analyze` maintains high quality and artistic intent by enabling playback through Dolby Vision enabled playback devices.

If CUDA is supported on the platform, the `cm_analyze` tool defaults to using CUDA for processing because this is usually faster processing than using the CPU. For CUDA support, the machine requires to have a NVIDIA graphics card. CUDA acceleration only supports a single card. If there are multiple GPUs on the machine, this tool will determine the fastest single GPU card and use its available cores for processing. Multiple GPU cards is not currently supported. To force the system to use CPU processing, use the `--cpu-only` option in the command line.

The `cm_analyze` tool is compatible with the following image formats. To output a selected image format, use one of the available image format extensions at the end of the output filename.

- MXF (.mxf)
- TIFF (.tif)
- OpenEXR (.exr)
- JPEG 2000 (.j2c/.j2k)
- DPX (.dpx)
- ProRes (.mov)

### 6.1 Required information and files

- Conformed HDR10 image sequence that is encoded in SMPTE ST.2084 (PQ)
- Shot list with shot type for the entire sequence (.txt, .sl, .edl). Also works with a Dolby Vision XML
  - If the shot list is not available, use the optional "live mode" to generate frame-by-frame metadata
- Aspect ratios for the sequence

## 6.2 Usage

Cm\_analyze requires the following information to be specified. Color Encoding metadata will be set based on the mastering display used.

- Source Format: Bit Depth, channel packing, sub-sampling, layout, endianness, chromacities, eof, video range
- Mastering Display ID
- Aspect Ratio or Letterbox Information
- Frame Rate

When specifying combined options with space delimited values, use quotes (“”) at the beginning and end of the specified string.

See [Advanced Color and Image Format Strings](#) for a list all available functions that are described in the output of the `./cm_offline --help-format` option.

See [Appendix D: Common input and output image format examples](#) for common image format examples.

### 6.2.1 Command line syntax

#### Basic Usage

```
./cm_analyze [-s <arg>] [-m <arg>] [--frame-offset <arg>] [-f] [-r <arg>] [--letterbox <arg1> <arg2> <arg3> <arg4>] [--aspect-ratios <arg1> <arg2>] [-b] [-l] [--source-format <arg>] [--show-mastering-displays] [--help-format] [--help-advanced] [-h] [-v] <input> <output XML>
```

#### Advanced Usage

```
./cm_analyze [-s <arg>] [-m <arg>] [--frame-offset <arg>] [-f] [-r <arg>] [--letterbox <arg1> <arg2> <arg3> <arg4>] [--aspect-ratios <arg1> <arg2>] [-b] [--analysis-version <arg>] [-l] [--source-format <arg>] [--show-mastering-displays] [-t <arg>] [--help-format] [--help-advanced] [-h] [-v] [--disable-font-color] <input> <output XML>
```

### 6.2.2 Command line help

#### Basic usage

```
./cm_analyze -h, or -help.
```

#### Advanced usage

```
./cm_analyze --help-advanced
```

#### Common example

```
./cm_analyze -s <shotlist.txt> -m 7 -r 24000/1001 --source-format "u16 i444 tight computer pq bt2020" --aspect-ratios 1.77778 1.77778
```

```
<input_hdr10_image_sequence_#####.tif>
<output_dolby_vision_metadata.xml>
```

### 6.2.3 Basic usage option information

Option Command	Description	Default
<code>-s --shot-list &lt;arg&gt;</code>	Specifies input shot list filename (.edl, .txt, .sl, .xml)	
<code>-m   --mastering-display &lt;arg&gt;</code>	Sets the Mastering Display by ID when using the '--reset-displays' flag (Overwrites any default behavior).	
<code>--frame-offset &lt;arg&gt;</code>	Offsets the metadata start frame number by this frame count to match the start image frame number	0
<code>-f   --frames</code>	Specifies the frame range(s) to process. For multiple frame ranges, use <frame_start#-frame_end#,frame_start#-frame_end#,single_frame,...> or <frame_start#> <frame_end#>	
<code>-r --frame-rate &lt;arg&gt;</code>	Sets the frame rate tag for the XML or MXF (24, 23, 24000/1001...).	input
<code>--letterbox &lt;arg1&gt; &lt;arg2&gt; &lt;arg3&gt; &lt;arg4&gt;</code>	Specifies the area to ignore for processing (left right top bottom).	0 0 0 0
<code>--aspect-ratios &lt;arg1&gt; &lt;arg2&gt;</code>	Sets the aspect ratios (Canvas, Image).	input
<code>-b --bda</code>	Computes MaxFALL, MaxCLL as per the Blu-ray Disc document and records the values in the output XML.	false
<code>-l   --live-mode</code>	Use to replicate live broadcast with frame-based metadata	false
<code>--source-format &lt;arg&gt;</code>	Optionally specifies the source image format (use --help-formats for detailed information).	input
<code>--show-mastering-displays</code>	Displays a list of supported mastering displays.	false
<code>--help-format</code>	Displays extended help on specifying image format strings for source, target and color encoding.	

<code>--help-advanced</code>	Displays advanced application arguments and usage.	
<code>-h -? --help</code>	Displays basic application arguments and usage.	
<code>-v --version</code>	Displays the program version.	

## 6.2.4 Advanced option information

Option Command	Description	Default
<code>--analysis-version &lt;arg&gt;</code>	Specifies the analysis algorithm version used to create L1 metadata. (use 2 for version 2.9, 4 for version 4.0)	Live mode = 4 Shot based = 2
<code>-t --threads &lt;arg&gt;</code>	Specifies the number of CPU threads to use (-1 indicates all available cores)	-1
<code>--disable-font-color</code>	Disable Log Font Color	false

## 6.2.5 Compatible input image formats

The `cm_analyze` tool is compatible with these image formats and the format using `--source-format` should be specified for each file format used.

When specifying combined options with space delimited values, use quotes (“”) at the beginning and end of the specified string.

See [Advanced Color and Image Format Strings](#) for a list of all available functions that are described in the output of the `./cm_analyze --help-format` option.

See [Appendix D: Common input and output image format examples](#) for common image format examples.

## 6.2.6 Using Live Mode analysis

Live Mode is useful when there is no shot list or EDL for the HDR10 deliverable that is to be converted for Dolby Vision delivery. In Live-Mode, `cm_analyze` creates per-frame analysis metadata instead of shot-by-shot metadata. Using `--live-mode` with `cm_analyze`, the tool will analyze every frame of the asset and apply a temporal filter between frames to smooth transitions between cuts or abrupt transitions.

The resulting XML is one shot with per-frame metadata for the entire sequence, carrying L1 metadata per each frame.

Example of Live Mode Usage with ProRes 422:



```
./cm_analyze --source-format "u10 i422 lsb32rev le ybcr_bt2020 pq  
bt2020 video" --aspect-ratios 1.77778 2.4 --mastering-display 21 --  
frame-rate 24000/1001 <input_prores_422.mov> <output.xml> --live-mode
```

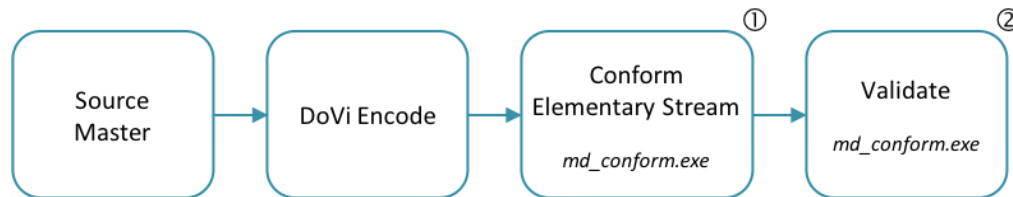
## 7 MD\_conform

Md\_conform is a bitstream metadata utility tool that works with Dolby Vision v2.9 bitstream files. This tool conforms L1 Analysis Metadata in Dolby Vision bitstream files to address luminance irregularities during playback for certain Dolby Vision TVs. This tool does not require a re-encode of the content.

The md\_conform tool is applied to bitstreams right after encoding.

### 7.1 Conformance Workflow

This is the workflow for conforming L1 metadata within the existing implementation of the Dolby Vision Encoder. In the future, the Dolby Vision Encoder will be updated, and this will no longer be required.



#### Example CLI

① `./md_conform --conform-L1 <source_el_bitstream.265> <conformed_el_bitstream.265>`

② `./md_conform --validate <conformed_bitstream.265>`

**NOTE:** For BD Profile 7, conform L1 using the EL (Enhancement Layer)

### 7.2 Usage

#### 7.2.1 Command line syntax

Basic usage

```
./md_conform [--validate] [--conform-L1] [-h|--help] [-v|--version] [-  
-disable-font-color] <source> <output>
```

Example

Conforming a bitstream file:

```
./md_conform --conform-L1 <source_bitstream.file>  
<conformed_bitstream.file>
```

## 7.2.2 Basic usage option information

Option Command	Description	Default
--validate	Checks the source file if conformance is required	
--conform-L1	Checks the source file and conforms L1 metadata	
--disable-font-color	Disable Log Font Color	
-v   --version	Displays program version	
-h   --help	Displays application arguments and usage	

## 8 Advanced color and image format strings (`--help-format`)

You can specify a space-separated string of the following tags to inform the sequence reader / writer about the images you are attempting to read or produce.

Option	Description and examples
Image size:	Specifies the size of the image. This is primarily for raw formats such as yuv in which there is no header. Usually it is read from the image file format header. The basic format is WIDTHxHEIGHT, with an optional number of image channels, 1920x1080 1920x1080x3
Raw image structure tags - Using these tags can to force the interpretation of the bits to a specific value if they cannot automatically be queried from the file itself. This is primarily for raw formats such as yuv.	
Image bit depth	u8   8bit   8bits u10   10bit   10bits u12   12bit   12bits u14   14bit   14bits u16   16bit   16bits u32   32bit   32bits f16   half f32   float f64   double
Orientation (pixels)	top_left top_right bottom_left bottom_right left_top right_top left_bottom right_bottom
Endianness	be (big endian) le (little endian) ne (native endian)
Layout	msb32 lsb32

	msb32_reverse   msb32rev ** usually for new dpx where things are stored lsb32_reverse   lsb32rev ** BGR (or component 2,1,0 for 4:2:2 YCbCr) msb16 lsb16 tight
Channel packing	planar interleaved
Sub-sampling	444 422 420
Color image state tags: These tags can be specified to define how the values in the image should be interpreted (such as color space)	
Encoding	All encodings allow specification of luminance range that defines scaling to nits except for absolute spaces such as PQ. Usage is: <encoding>(0.0,100.0) <encoding> is one of the following encoding names. The first argument is the minimum luminance; the second is the maximum luminance in nits. linear pq gamma_bt1886   gamma1886 gamma_bt709   gamma709 gamma_bt2020   gamma2020 gamma_srgb   gammasrgb gamma_dci   gammadci log slog slog2 logc aceslog

Colorspace	<p>Specifies the current color space of the image. Some of these values represent fixed transforms, from one back to RGB (or to one), such as ycbcr_bt709. Others imply a combination of a 3x3 matrix transforms to and from XYZ.</p> <p>rgb  ycbcr_bt601   ycbcr601  ycbcr_bt709   ycbcr709   ycbcr_r709   ycbcr_rec709  ycbcr_bt2020   ycbcr2020   ycbcr_r2020   ycbcr_rec2020  xyz  ictcp</p>
Signal range	<p>Specifies the range of values. Full range is considered unscaled, video is the typical 16 - 235 (when in 8-bits) values. Signal is the SDI 'valid' range of 4-1019 in 10-bits (and scaled, not inserted)</p> <p>signal   sdi  video   legal   narrow   smpte  computer   full   data  plus   smpteplus   smpte_plus</p>
Chromaticities and Whitepoint	<p>Specifies the color coordinates represented by 0 - 1 in normalized values. These values automatically compute the 3x3 transforms between XYZ and RGB, for example.</p> <p>p3d65  bt709   r709   rec709  bt2020   r2020   rec2020  bt601ntsc   ntsc  bt601pal   pal  dci  aces  sgamut3  sgamut3cine   sgamut3_cine  custom (red.x, red.y, green.x, green.y, blue.x, blue.y, white.x, white.y)</p>
Framerates	<p>Specifies the frame rate of the material by having the frame rate followed immediately (no space) by <b>fps</b>, as shown in the following values.</p> <p>23.976fps  24fps  29.97fps  30fps  60fps</p>

<p>Frame range</p>	<p>Specifies the frame range of the sequence. If this is not specified, the sequence attempts to scan the directory and discover the frame range (or for yuv files, look at the file size and image size to compute the range). This allows the specification of a shorter range.</p> <p>0-100</p>
<p>Shortcuts</p>	<p>444p   p444: 444 planar  422p   p422: 422 planar  420p   p420: 420 planar  444i   i444: 444 interleaved  422i   i422: 422 interleaved  420i   i420: 420 interleaved</p>
<p>Presets</p>	<p>These simple tags imply a collection of the above color information tags.</p> <p>bt601_ntsc_ycbr: gamma_bt709(0,100) ycbr_bt601 video bt601ntsc  bt601_pal_ycbr: gamma_bt709(0,100) ycbr_bt601 video bt601pal  bt709_rgb: gamma_bt709(0,100) rgb video bt709  bt709_ycbr: gamma_bt709(0,100) ycbr_bt709 video bt709  bthdr: pq rgb video bt2020  bthdrfull: pq rgb computer bt2020  computer_srgb: gamma_srgb(0,100) rgb computer bt709  dcinema: gamma_dci(0.024,48) xyz computer dci  exr_linear: linear(0,100) rgb computer bt709  jpeg: gamma_srgb(0,100) ycbr_bt601 computer bt709  prm: pq rgb signal p3d65  pulsar: pq rgb computer p3d65  sony_cinema: slog3(0,48) rgb computer sgamut3_cine  sony_video: slog3(0,100) rgb video sgamut3</p>

## Appendix A: Mastering and target displays for cm\_offline and/or cm\_analyze

This appendix describes the available target displays for the cm\_offline tool and mastering displays for the cm\_analyze tool.

The cm\_offline tool synchronizes targets with the latest software version of the Dolby Vision Content Mapping Unit (CMU).

For the list of the latest mastering and target displays, always refer to the target help or the help menu for the third-party tool that is connected to the CMU.

```
./cm_analyze --show-mastering-displays  
./cm_offline --show-targets
```

### Mastering Displays

ID	Name
7	4000-nit, P3/D65, ST.2084, Full
8	4000-nit, Rec. 2020, ST.2084, Full
20	1000-nit, P3/D65, ST.2084, Full
21	1000-nit, Rec. 2020, ST.2084, Full
30	2000-nit, P3/D65, ST.2084, Full
31:	2000-nit, Rec. 2020, ST.2084, Full
Default (7)	4000-nit, P3/D65, ST.2084, Full

### Target Displays

ID	Name
1	100-nit, Rec. 709, BT.1886, Full
27	600-nit, P3/D65, ST.2084, Full
37	2000-nit, P3/D65, ST.2084, Full
38	2000-nit, Rec. 2020, ST.2084, Full
48	1000-nit, P3/D65, ST.2084, Full
49	1000-nit, Rec. 2020, ST.2084, Full
Default (1)	100-nit, Rec. 709, BT.1886, Full



## Appendix B: Creating a longplay deliverable from multiple sources

This function creates a single long play mezzanine or long play XML from multiple reels or multiple XMLs.

**Warning:** Do not use this function to cut a reel. For cutting reels, use the `[-f|--frames <arg1> <arg2>]` function available in the `cm_offline` and `mezzinator` tools.

When creating a custom `reel.txt` file, specify the XML associated with the image sequence and the frame-start and frame-end from the frame numbers specified in the XML.

Create a plain text, `reel.txt` file with the following information:

```
<metadata_file1.xml> <first_frame_sequence_#> <last_frame_sequence_#>
<metadata_file2.xml> <first_frame_ sequence_#> <last_frame_sequence_#>
```

**Note:** The last frame number is defined as  $last\_frame\_start + last\_shot\_length - 1$

### Finding frame numbers

When the `frame_start` number is not equal to zero, you must obtain the frame numbers from within the XML file as shown in the following example. To print shots in the XML, use the `curl` command in Terminal to easily find the frame starts and duration.

#### Curl Example

```
$ curl file:///<fullpath_filename.xml> | grep In
<Source> <ParentID>0</ParentID> <In>0</In> </Source>
<Record> <In>86400</In> <Duration>48</Duration> </Record>
<Source> <ParentID>0</ParentID> <In>0</In> </Source>
<Record> <In>86448</In> <Duration>72</Duration> </Record>
```

In this example, the first frame number is **86400** and the last frame number is **86448 + 72 - 1**

When `frame_start` number is zero, you can alternatively use the `metafier` tool on the XML file to extract the frame number of each shot contained in the XML, as shown in the following example:

```
./metafier --shot-list - <filename.xml>
```

#### Output:

Start frame of each shot

When using the mezzinator tool with a `reel.txt` file, the `--reel <arg>` replaces the `-m | --metadata <arg>` command. Generally, you use a `playlist.play` file as the input source. For more information, see [Appendix C: Creating a playlist file](#).

The following example shows using the mezzinator with reels to create a longplay MXF.

```
./mezzinator --reel <reel_filename.txt> <input_filename.play>  
<output_filename.mxf>
```

## **Creating a longplay XML only**

The following example shows using metafier to create a longplay XML using reels.

```
./metafier -o <output.xml> <reel_filename.txt>
```

## **Creating a longplay XML with additional black frame header and tails**

To create a longplay XML in which the image sequence has black frame headers and tails that were added without metadata, use the `--insert-black-head <arg>` and `--insert-black-tail <arg>` function in the metafier tool. Specify the number of black frames to be captured in the metadata for head and tail.

The following example shows adding 24 black frames to the header and tail of the longplay image sequence:

```
./metafier -o <output.xml> <reel_filename.txt> --insert-black-  
head 24 --insert-black-tail 24
```

## Appendix C: Creating a playlist file

Use a playlist file in combination with the `reel.txt` file to create a single long play mezzanine from multiple reels.

A `playlist.play` file contains the input frame list from multiple sources or paths.

Create a plain text file that has the `.play` file extension (`<filename.play>`), and contains the following information:

```
<image_file_#####.ext> <first_frame_sequence_#>-  
<last_frame_sequence_#> <format_preset_string>
```

**Note:** For the playlist file, the frame sequence numbers are the numbers used in the image sequence filename.

For the `<format_preset_string>`, use the following presets or the customized values described in [Appendix D: Common input and output image formats examples](#).

```
bt601_ntsc_ycbcr | "gamma_bt709(0,100) ycbcr_bt601 video  
bt601ntsc"  
bt601_pal_ycbcr | "gamma_bt709(0,100) ycbcr_bt601 video bt601pal"  
bt709_rgb | "gamma_bt709(0,100) rgb video bt709"  
bt709_ycbcr | "gamma_bt709(0,100) ycbcr_bt709 video bt709"  
computer_srgb | "gamma_srgb(0,100) rgb computer bt709"  
dcinema | "gamma_dci(0.024,48) xyz computer dci"  
exr_linear | "linear(0,100) rgb computer bt709"  
jpeg | "gamma_srgb(0,100) ycbcr_bt601 computer bt709"  
prm | "pq(0.005,600) rgb signal p3d65"  
pulsar | "pq(0.005,4000) rgb signal p3d65"
```

The mezzinator tool can optionally use a `playlist.play` file as the input file instead of using the image sequence file.

The following example shows using the mezzinator tool with a playlist file.

```
./mezzinator --reel <reel_filename.txt> <input_playlist.play>  
<output_filename.mxf>
```

## Appendix D: Common input and output image formats examples

By default, `cm_offline` and `mezzinator` tools check the image format header and the metadata file header for the color encoding space.

**Note: The `--source-format` and `--output-format` functions do not need to be specified unless image conversion is required.**

The general format string to specify the input and output format is:

```
"<horizontal_pixels>x<vertical_pixels> <bit_depth> <channel_packing>  
<color_subsampling> <endianness> <layout> <frame_rate> <orientation>  
<encoding(min_luminance, max_luminance)> <color_space> <value_range>  
<chromaticities_whitepoint>"
```

**Note:** When specifying combined options with space delimited values, use quotes (""") at the beginning and the end of the specified string.

Generally, you do not specify the output image format unless you want to change the container or override the image.

If you want to override the output image, the `cm_offline` tool performs the conversion on the output image, therefore changing the output image to the format that you specified.

Additionally, you do not usually use the `<encoding>`, `<color_space>` and `<chromaticities_whitepoint>` arguments.

---

**Common format examples**

<b>Image format</b>	<b>Example</b>
8-bit TIF/TIFF	"u8 interleaved tight" <output_filename.tif>
10-bit DPX	"1920x1080x3 <b>u10 interleaved</b> 444 ne <b>msb32</b> " <output_filename.dpx> <b>Note:</b> The important arguments are marked in <b>BOLD</b>
12-DPX	"u12 interleaved lsb16" <output_filename.dpx>
12-bit MXF JPEG 2000 standalone, without metadata	"u12 interleaved lsb16" <output_filename.mxf>
12-bit JPEG 2000	"u12 interleaved lsb16" <output_filename,j2k[j2c]>
16-bit TIF/TIFF	"u16 i444 tight" <output_filename.tif[tiff]>
16-bit OpenEXR	"f16 tight" <output_filename.exr>
ProRes 4444/4444XQ	"u16 i444 be rgb tight"
10-bit ProRes422*	"u10 i422 lsb32rev le ycbcr_bt2020 video"
8-bit ProRes 422*	"u8 i422 le tight ycbcr_rec709 video"

## Appendix E: Metadata alignment examples

The metadata and image essence is always reset to zero when creating a MXF package.

### Using the `--frame-offset` function to align metadata

The frame offset function uses the following equation:

$$\text{offset} = (0 - \text{start\_frame\_of\_sequence})$$

To specify a frame range, it uses:

$$\text{offset} = (0 - \text{start\_frame\_of\_range})$$

The following examples show the mezzinator tool's automatic alignment function.

#### Use Case 1

Aligned metadata [86400, 86500] and frame sequence [86400, 86500]

The metadata is realigned to zero, and becomes [0, 100]. The offset is estimated to be -86400.

$$f = 86400, \text{ we have } (f = 86400 - 86400); \text{ or } f = 0$$

$$f = 86500, \text{ we have } (f = 86500 - 86400); \text{ or } f = 100$$

#### Use Case 2

Unaligned metadata [300, 400] and frame sequence [86400, 86500]

The metadata is realigned to zero and becomes [0, 100]. The offset is estimated to be -86400.

$$f = 86400 \text{ we have } (f = 86400 - 86400); \text{ or } f = 0$$

$$f = 86500 \text{ we have } (f = 86500 - 86400); \text{ or } f = 100$$

#### Use Case 3

Unaligned metadata [90000, 90100] and frame sequence [300, 400]

The metadata is realigned to zero so becomes [0, 100] offset is estimated to be -300.

$$f = 300 \text{ we have } (f = 300 - 300); \text{ or } f = 0$$

$$f = 400 \text{ we have } (f = 400 - 300); \text{ or } f = 100$$

#### Use Case 4

Aligned case metadata [86400, 86500] and frame sequence [86400, 86500] with frame range [86450, 86455]

The metadata is realigned to zero and becomes [0, 100]. The offset is estimated to be -86450.

f = 86450 we have f = 86450 - 86450; or f = 0

f = 86455 we have f = 86455 - 86450; or f = 5

**Note:** This estimate is incorrect. In this case, you must impose an offset of -86400.

f = 86450 we have f = 86450 - 86400; or f = 50

f = 86455 we have f = 86455 - 86400; or f = 55

#### Using the `--frame-align` function to align metadata

Alternatively, the frame align function aligns the metadata to the specified frame number used in the image sequence filename and reset those to zero accordingly when creating the resulting MXF.

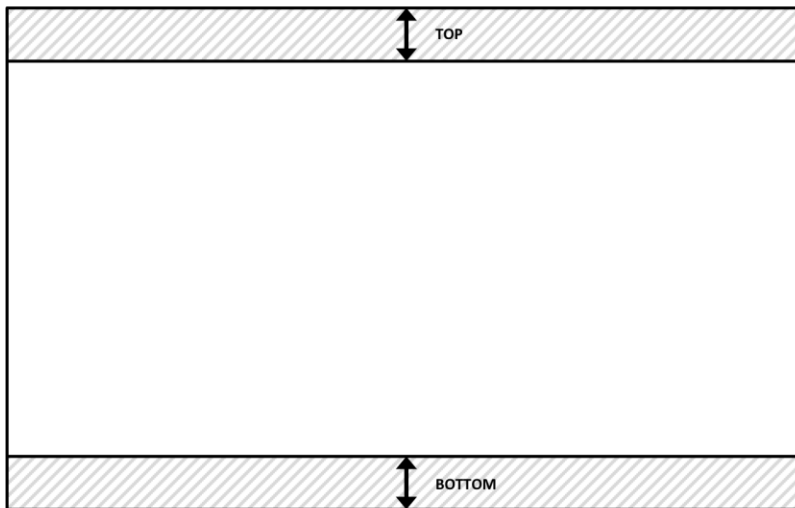
## Appendix F: Specifying Letterbox

When specifying the letterbox area with `cm_offline` and `cm_analyze`, the size used for the letterbox command is the letterbox area and not the active image area as shown in the below diagram. The sequence for the letterbox values are left, right, top, bottom.

```
--output-letterbox <left> <right> <top> <bottom>
```

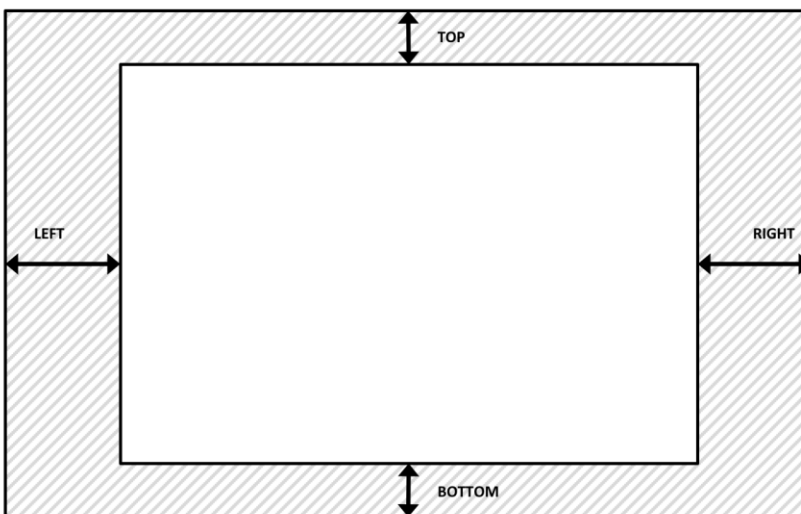
Example 1:

```
--output-letterbox 0 0 274 274
```



Example 2:

```
--output-letterbox 360 360 274 274
```





## Appendix G: Custom Target XML

### Custom Target XML Structure:

```
<?xml version="1.0" encoding="utf-8"?>
<CustomTarget>
  <Version>1.0</Version>
  <Name>Name of Custom Target</Name>
  <Primaries>
    <Red>0.68,0.32</Red>
    <Green>0.265,0.69</Green>
    <Blue>0.15,0.06</Blue>
  </Primaries>
  <WhitePoint>0.3127,0.329</WhitePoint>
  <MinimumBrightness>0</MinimumBrightness>
  <PeakBrightness>2000</PeakBrightness>
  <EOTF>pq</EOTF>
</CustomTarget>
```

### Valid values for each target parameter:

TargetDisplay ID = 255

Always use Target ID 255 when creating a custom target

TargetDisplay NAME = Use a descriptive name for the target display

color\_primaries = Rx Ry, Gx Gy, Bx By

up to 4-decimal places in xy coordinates for Red, Green and Blue

white\_point = x y

up to 4-decimal places in xy coordinates

peakbrightness = whole number nit values

such as 1000, 1010, 550

minbrightness = up to 6-decimal places in nits, less than 1.0 nits

such as 0.0001, 0.000005

EOTF options = pq | gamma\_bt1886 | gamma\_dci | gamma\_22 | gamma\_24 | linear

DiagonalSize = Fixed at 42

## Appendix H: Operating system compatibility

This appendix describes the compatible operating systems on which the Dolby Vision Professional Tools have been tested. They may function on other operating systems that have not been tested by Dolby Laboratories. We recommend that you only run the tools on these supported operating system versions.

Compatible or tested operating system	Supported versions
<b>Linux</b>	CentOS Linux 6.5+, 7.x
<b>Mac OS X*</b>	Mac OS X 10.9.5+
<b>Microsoft Windows</b>	Windows 7.x, Windows 10 <b>Note:</b> Windows also requires Visual C++ Redistributable for Visual Studio 2012.
<b>GPU Support</b>	Most NVIDIA GPUs that support CUDA

\*CUDA is only supported on Mac systems that are using the NVIDIA-published version of GPU drivers and the CUDA runtime. The standard Mac graphic drivers are not compatible.

**Note:** CUDA is only compatible with NVIDIA GPUs. Systems with non-NVIDIA GPUs will only be able to use CPU processing.