# Render codecs in Mistika VR

Here it is a small introduction to the render codecs in Mistika-VR.

Please note that only the most representative codecs are mentioned here, and some technical explanations contain over-simplifications (this document is not intended for advanced users)

Currently Mistika VR render support these tracks:

- 1 Video track, also with support for Stereo3D in split image modes.

- 2 Audio channels. (Just mute the cameras that should not be audio source). The user can choose between audio embedded in a movie file or as a separate file.

The rest of this document is divided in two sections, compressed codecs and uncompressed codecs.

## COMPRESSED CODECS

Compressed codecs produce files with small size and require low bandwidth, but some information is lost on each generation (although there are some “lossless” variants).

In general they are great for final deliveries, and also for images that are not going to suffer more changes in post production. But only the “lossless” variants (explained below) should be used for passing images to other post production applications if they will go beyond basic editing tasks.

Please note that most cameras will have applied a first compression generation, and final consumer deliveries will also produce another one. Adding more compression stages on intermediate renders will degenerate image quality pretty fast.

**NVidia hardware codecs Vs. software codecs:** NVidia codecs can render much faster than software codecs (typically 10 times faster ), while producing similar quality that their software equivalents. However, they need some resources that may not be available:

- A NVidia board of the latest generations (maximum resolution depend on each model). Click here to find the available render resolutions for each GPU

- Windows or Linux OS. (Apple does not support NVidia encoder, at least at the date of this document )

The render speed of NVidia codecs only depend on the GPU. Meanwhile, the speed of all the other codecs only depend on the CPUs (except for the case of uncompressed codecs, which mainly depend on the storage speed).

**444 vs 420:** In few words, the “YUV 420” term means that part of the chroma information is shared between adjacent lines of the image, while RGB 444 keeps all the sensor information. 420 produce smaller files with no significant visual impact, but it removes information that can be important for chroma keys and other VFX tasks. As a result, 420 is an excellent format for previews and for final delivery to end consumers, while RGB is more indicated for intermediate renders and VFX. But it also depends on the quality of the original cameras (if the original cameras were already compressing to 420 then you can keep going in this format. Check your camera specifications to get this information)

**H264 Vs. H265:** H265 provides more quality at the same bitrate, and it also supports higher resolutions up to 8k (h264 only supports levels up to 4k). In the other hand, h265 is slower to decode and to playback than h264.

For those reasons, in general we recommend H264 for HD and 4K. And H265 for 6K and 8K

**GOP factor:** Many codecs have a GOP definable value (you will find it in their settings wheel). Basically, the GOP (Group Of Pictures) defines the number of frames between keyframes.

In few words, the compression on a keyframe do not depend on adjacent frames (think about it as a jpg image), while intermediate frames between keyframe only contain their “differences” with the next (or previous) keyframes. High GOP values produce smaller files and reduced bandwidth needs for streaming applications. But they are not recommended for editing applications, as they will need a lot of processing to jump to a particular frame (and some applications do not even support GOP editing at all). Keep it low for editing and VFX, and increase it for previews and for final deliveries. The suitable GOP also depends on the content, so we can not provide more specific rules.

**Lossless compression Vs. QP**: If the rendered images are going to go trough professional color grading or high end VFX processing then you should use “Lossless” compression whenever it is possible.

In few words (oversimplifying...), “Lossless” codecs don´t change the images, or they only affect information that was already lost due to the signal to noise ratio of the cameras, but that is enough to reduce the file size significantly (typically 3:1 or more).

Meanwhile QP permits user defined compression by setting the **Q**uantization **P**arameter (**lower value = more quality** and bigger size ). Please note that when using QP what is constant is the subjective “quality”, not the bitrate (the bitrate will constantly evolve using more or less bitrate depending on the content, in order to achieve a target quality factor).

As a result, the meaning of the QP value is content dependent, so there are no exact rules to follow. But for average content we could say that values from 10 and below should produce high quality “lossless” results, ideal for intermediate renders to pass the results to other post production applications. In the other direction, values between 11 and 24 can be good for final consumer deliveries providing good visual quality (although not enough data for applying further post production tasks), and higher QP values approaching 25 or more will produce small files but low quality and noticeable artifacts.

**QP compression Vs. Bitrate**:

**The “QP” modes** are suitable as intermediate format, with consistent quality adjustable from technically lossless (QP = 0) to visually lossless (QP = 10) and quality loose beyond that point. Or in other words, high quality (QP around 10) normal quality (QP around 20) all the way down to strongly compressed (QP = 50).

Use low GOP values for intermediate format for better scrubbing performance in other apps., or higher GOP values to produce smaller files at the cost of editing interactivity (and potential quality issues at some point). Options of this mode are: GOP and QP factor.

**The “Bitrate” modes** allow maximum bitrate of 135.000 kb/s. This is too low for a 6k image for example, and can produce cyclically dropping quality with each GOP (good quality on keyframes, the progressively lower quality between them). Use Bitrate mode only if defining a specific bitrate is important (and even with that is not very precise and considered obsolete for modern formats), otherwise it is always better to use QP. Options of the Bitrate mode are: GOP and Bitrate.

**Inject spatial metadata:**

This setting add some extra metadata to tell the player applications that the media is an VR360 file. If not present, general purpose players may fail to switch to 360 mode and play the images as they are, even if they support 360.

**Main lossless codecs:**

> **NVidia HEVC 444 10b Lossless**: No useful information is lost, and it provides significant file size reduction compared to uncompressed codecs. In the other hand, they can be slow to decode, and not all applications support HEVC
>
> **NVidia HEVC 420 8b Lossless**: Produce smaller size, but being 420 8bit means that it is only “lossless“ for low cost cameras that are also based on YUV420

**Main general purpose codecs:**

**NVidia mp4 H264/H265(HEVC) QP**: This is probably the best codec for general purpose deliveries. Provides best compression and speed for a same bitrate.

**H264** provides great compatibility and it is fast to decode, ideal for real time playbacks up to 4K. **If you are not sure about what codec to use then this should be the one to use by default**

**H265/HEVC** supports **8K** and provides additional quality at the same bitrate, but it can be much slower to decode.

Note: In **QP** formats, Quality is user defined by defining the **Q**uantization **P**arameter (lower means better quality & bigger size.

**Apple Prores:** It is the standard family of codecs for Mac computers, but it also works in other platforms. In general provides similar quality and size as the equivalent H264 variants, but it is slower to render due to the lack of GPU hardware encoding. Prores encoding is based on CPU, so this component is critical for good render speed. However it is not limited in resolution, and with the latest optimizations from v8.8.10 it can be a good alternative.

**JPG 8b image sequence**. Being an image sequence it is not so efficient in terms of compression because it can not compress across time, and it is also slower to render. But it is still an interesting format due to one advantage: It is very compatible and most media applications support it, even the very old.

Enumerated formats also have other advantages, but for those aspects the EXR DWA is generally superior, so they are commented in the EXR point..

**EXR DWA**. This is an image format developed by ILM and mainly used as an intermediate format for VFX workflows. Probably the most popular variant is DWA (created by DreamWorks Animation). It is an image sequence (one frame per file), which makes it ideal for many VFX workflows, and in fact most VFX applications support this format.

A difference to movie files it can be rendered in parallel (several render nodes working together to render the same clip) with render managers like Smedge, Deadline, or Mistika Ultima BatchManager. Also it can be partially “patched” when needed (as a difference, “movie” files need to be rewritten completely).

This permits an interesting workflow for Mistika VR users: You can render a first stitch for initial review (for example with Mistika Boutique or Mistika Review products) and then adjust and re-render only the parts containing fixes. To do that just put the edit marks in the area to fix and render with the same name, so Mistika will only

overwrite the selected frames . (obviously this technique only works as long as you do not need to move the horizon or other geometry settings anymore)

Another advantage of the EXR format is that it supports 16bit per channel, which makes it ideal for HDR workflows and for high end cameras in general.

EXR DWA provides definable compression, with the default compression value (45) it is considered a “Lossless” format, even for high end cameras.

The EXR DWA format is CPU intensive, but as a difference to jpg this format is fast to encode and to playback, being possible to playback in Mistika applications at 4K when having decent CPUs.

Another EXR variant of common usage is **EXR ZIP.** This is made by compressing the original image to ZIP format internally ( the ZIP compression is not only lossless but also fully reversible). But It is much slower and less efficient in terms of file sizes when compared to DWA, so it is only recommended when explicitly required by an VFX department.

## UNCOMPRESSED CODECS

These formats do not apply any compression (apart from their nominal bit depth and sampling limitations). They are ideal for sending media to high end VFX and other professional post production tasks. In general they are bigger than “Lossless” codecs, but also more compatible and faster to decode, (as long as the storage is fast enough). In the other hand, **they require big and fast storage volumes**.

### Uncompressed Movie Formats:

**Mistika .js:** It is the only uncompressed “Movie” format available (all images of the clip contained in the same file). It is designed for perfect memory alignment and optimal real time playback performance (up to 8K 60p uncompressed in real time). It also supports parallel render with Mistika Ultima Totem tool, and It avoids file system fragmentation problems. In the other hand only Mistika applications support this format, but if you plan to use it on them then it is probably the optimal format.

### Uncompressed Image Sequences:

Apart from maintaining the best quality, being enumerated sequences they can be rendered in parallel with Smedge, Deadline, or Mistika Ultima BatchManager. And they can also be patched by parts.

**DPX RGB 10b:** It is probably the most standard format in the VFX industry. Also, it does not have resolution limits in itself.

**Tiff 16b:** Used to produce masters coming from very high end cameras with a lot of extra HDR range. It produce extremely big files with no real benefit in most cases. Only use it if you are absolutely sure that you need it.