White Paper

Visual Technologies
Multi-device Enabling

# Unlocking the Power of Intel® Deep Link

## Part Four: Accelerating Video Encoding, Decoding and Transcoding Using Intel® GPUs


**This paper is part four in a series of whitepapers designed to provide details regarding openly available development tools that can be used to take full advantage of Intel® Deep Link Technology.**

Together these papers will introduce and demonstrate some of the tools and processes that can be used to leverage Deep Link and allow developers to build better performing and more efficient applications. Real world use cases are included to showcase the full potential of Deep Link.

This fourth paper focuses on applying Deep Link concepts for improving compute performance of video transcoding applications.



### Authors

**Jonas Kollberg**
Strategic Account Manager, CDRM, CCG

**Artem Galin**
Software Enabling and Optimization Engineer, CPIE, CCG


### Table of Contents



## Introduction

Deep Link is Intel®'s innovative platform that combines the power of both the integrated and the discrete GPUs. With the release of our 12th Generation processor, Deep Link has been made even more powerful thanks to the upgrade to Intel®'s Iris® X$^e$ graphics processor and the all-new Intel® Arc™ graphics processor.

With these improvements, developers now have even more ability to strategically apply computing power that was previously unavailable, and to assign tasks to parts of the machine which would otherwise just lie dormant.

**That is the power of Deep Link.**


**Figure 1.** Intel® Deep Link combines a 12th Generation processor or later with an integrated Iris® X$^e$ GPU and a discrete Intel® Arc™ GPU, and managing the use of multiple GPUs simultaneously.

## Deep Link Technology

At its core, Intel® Deep Link Technology is a reimagining and reshaping of the way that a machine's Central Processing Unit (CPU) and Graphics Processing Units (GPUs) interact. Already available on a number of devices, Deep Link offers the ability to combine the computing power of a discrete GPU with that of a powerful integrated GPU.

Using Deep Link, complex workloads and pipelines can be constructed which use multiple computing elements with a high degree of code re-use, thereby simplifying code development and reducing overall effort. This approach offers significant gains in both performance and efficiency, boosting the functional capabilities of a given application by offering expanded computing capabilities and processing options.

**Deep Link Puts Multiple GPUs to Work**

By partitioning computational elements into logical segments, Deep Link GPUs can equitably share the workload - each one working independently from and concurrently with the other. The Intel® Iris® X$^e$ graphics processor (also referred to as the integrated, or iGPU) and the all-new Intel® Arc™ graphics processor (also referred to as the discrete, or dGPU) can be used together to equitably split tasks, often allowing a workload to be completed in roughly half the time.

**Deep Link Gets the Most out of Intel® Hardware**

Deep Link enables Intel® computing components to work together at a level of speed and efficiency that was not available before by combining the computing power of multiple GPUs with similar characteristics. Because the two graphics processors use the same kernel code and have similar computing power and performance characteristics, there is little additional overhead introduced when partitioning tasks between the two.

One of the most common compute-heavy use cases that could benefit from Deep Link is video encoding.

## Managing Video Sizes and Formats

According to multiple web sources, approximately 80% of internet traffic is used for video streaming. For this reason, creating accessible video content has become a priority for everyone who wants to have their message heard - from large corporations to small businesses to individual content creators.

Creating accessible content online typically means making it available to wide audiences and adaptable to ever-evolving streaming platforms. A critical part of ensuring this accessibility is the proper management of the video file sizes and formats.

**Size**

Size is an important consideration because video files can get very large, especially when the video is of considerable length or when high picture quality is required. Although modern video players can handle large files much better now than they could even a few years ago, there is still a reasonable limitation to the size of the video files due to concerns both from the aspect of storage space as well as streaming bandwidth.

To shrink file size without sacrificing quality or removing content, the data that makes up the file requires compression handled through a process known as *encoding.* Encoding is used to compress the data within a file to decrease its size or bitrate. An algorithm, or a series of algorithms, can utilize encoding methods to programmatically decrease the overall size of the file without significantly affecting quality.

The application that houses and applies the encoding algorithms is known as a *codec*. The word *codec* is a combination of the words *encoder* and *decoder.* A codec is programmed and supplied with a set of specific rules that it must follow as it processes the data and performs the encoding functions. These rules typically pertain to how aggressive the application needs to be in compressing content to satisfy all of the file size and quality requirements.

**Format**

While it is related to file size in some ways, the video format can be a problem of its own. Part of this issue is simply that there are so many different formats available (the Wikipedia *Video file format* page currently lists thirty-one different formats).

When discussing the differences between video formats, it is important to note that the format names are sometimes interchangeably referenced by their container. A container is a single file that includes all of the various elements that make up a completed ready-for-playback video: the video data, audio data, captions, metadata, etc. Common containers are typically referred to by their file extensions, such as MP4, AVI, or MOV (for .mp4, .avi, and .mov files respectively).

One of the more commonly used video containers is MP4, and one of the codecs most often used to encode video data is the Advanced Video Coding (AVC) codec, more commonly referred to as H.264.

The H.264 codec has a high rate of compression, high quality retention, and compatibility with most players and services (including YouTube, a requirement for any serious content creators). The MP4 container and the AVC codec are two parts of the same standard: the Motion Picture Editors Guild version 4, otherwise known as MPEG-4. More information about the MPEG-4 standard can be found here.

The successor to the H.264 codec is the H.265 codec, also known as the High Efficiency Video Coding (HEVC) codec. This codec was designed to process videos with higher resolution, along with several other important and useful upgrades.

The H.265 codec offers better data compression at the same video quality with an improved bitrate than the H.264 codec. H.265 is the codec part of the MPEG-H standard, which is the successor to the MPEG-4 standard. More information about the MPEG-H standard can be found here.

An Open Source alternative to the H.26x codecs is the AV1 codec, pioneered by the Alliance of Open Media. Intel® is part of the Alliance for Open Media. The AV1 codec is an Open Source, relaxed royalty video codec that produces high-quality, commercial-grade media with an excellent bitrate. While Intel® has supported HEVC and AVC for years, AV1 is well positioned to become very popular. This potential has us currently supporting AV1 and having our Intel® Arc™ graphic cards natively support AV1, an industry first.


**Figure 2.** The AV1 Alliance for Open Media logo.

The following section describes some of the ways that formats are utilized in common video development use cases.

## Encoding, Decoding and Transcoding

Very often it becomes necessary for video content to be downsized (for storage or faster streaming), upsized (to meet display requirements), or translated (changed to a different format). The processes that allow for these tasks are known as *encoding*, *decoding*, and *transcoding*.

*Encoding* works to make files smaller through compression. Encoding becomes necessary when a file is too large to be stored, transferred or streamed. A common use case for encoding is video storage on a server or a hard drive. Encoding is very compute intensive, so optimizations to software and hardware that make encoding more efficient are highly valued by content creators.

*Decoding* works to restore a file to its decompressed state. It is part of the playback of the file to de-compress a video file. Decoding also may be important when editing the video or with other use cases where file sizes no longer need to be as restricted by things like bandwidth or storage space.

*Transcoding* works to change the file parameters, which can include things like format, quality, bitrate and the specific codec. Not all playback and display applications are capable of utilizing the same file format, so the ability to transform video content to another format becomes important when trying to reach the widest audience. One of the most common use cases for transcoding is uploading a file to YouTube. According to YouTube, there are sixteen different video formats that it will accept for upload[1], but the H.264 video compression codec and the MP4 file container format to host encoded video are recommended.

But for the video creation community, is there an inexpensive, easy-to-use option for transcoding video content?

*There certainly is.*

## Introduction to HandBrake

As new technologies and opportunities continue to grow the video creation community, more and more resources are being consumed to capture this digital information. This causes new questions to arise:

- How can I ensure that my content will look good on the widest variety of target devices?
- How can I transcode batches of video content into different formats (and resolutions)?
- How can I utilize the full power of my computer to save time during the transcoding process?
- How can I measure the video performance of my current or upcoming systems?
- How do I manage large files and scale them down for use on the web?

For an increasing number of content creation specialists, video developers, and even technical journalists, the answers to all of these questions can be found in a single application: HandBrake.

HandBrake is an Open Source application that is used to transcode video and audio to and from virtually any currently used format. The application is free to download and install and works with Windows, macOS, and Linux.

1 - YouTube transcodes all uploaded content to one of a small number of broadcast formats (including AV1) to accommodate different streaming devices, often changing things like size, bitrate, and format.

Initially released in 2003, HandBrake was created to make transcoding easier and faster for all users. It has seen many modifications and new releases since then, becoming a useful tool for professionals and enthusiasts alike.

**Figure 3.** The HandBrake logo.

HandBrake is a full-pipeline tool. This means it can be used to change containers, codecs, bitrates, frames-per-second, display resolution or any other video parameters as needed to suit the needs of any video specialist or content creator. If you need to make your video bigger, or smaller, or need to change its format, it's a safe bet that HandBrake has an option to do just that.

## Using Intel® Accelerators to Improve HandBrake Throughput

Due to its versatility, HandBrake is an incredibly useful application that can be used to make any number of necessary changes to video content. However, some of those changes require a significant amount of time and processing power. Using the Intel®-powered accelerators described below, systems using HandBrake can utilize all the available compute power to accelerate transcoding. Specifically, HandBrake uses Intel® Quick Sync Video and Intel® Deep Link through Intel® oneVPL to perform transcoding functions at a greatly accelerated pace.

### Intel® Quick Sync Video

Intel® Quick Sync Video (QSV) is Intel®'s solution to the growing demand for additional processing power needed when transcoding video. QSV channels the dedicated media processing capabilities of Intel® graphics technology (including many of our current processors, iGPUs, and dGPUs) and uses it to decode, encode or transcode video content at a much higher speed.

Figure 4 (below) shows video hardware acceleration options using the Preferences section in the file area. Use the following steps to select the recommended options for using QSV in HandBrake:

1. Click on the *Video* section within the *Preferences* menu.
2. Select "Allow use of the Intel QuickSync Encoders" and "Enable Low Power QuickSync Hardware (where supported)."
3. Check the "Prefer use of Intel QuickSync for decoding video when available" under *Decoding*.

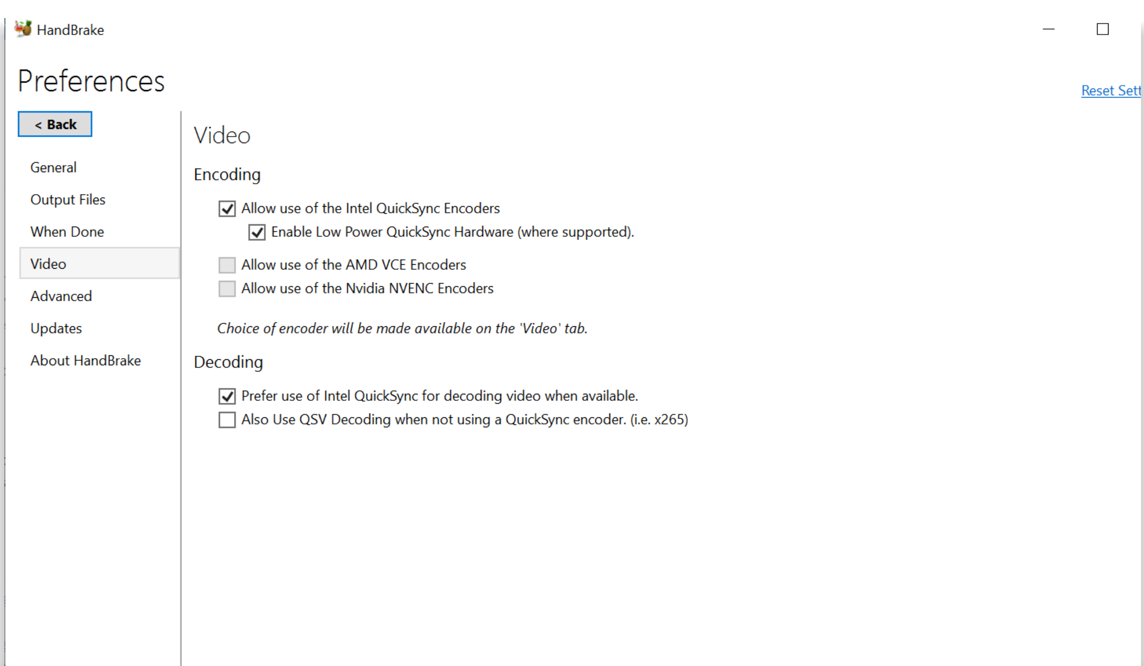


**Figure 4.** Selecting QSV in the Video Preferences menu.

The following dedicated hardware presets with most optimized encoding options were enabled to leverage the full power of Intel® QSV for the H.265 codec:

- H.265 QSV 2160p 4K
- H.265 QSV 1080p

Figure 5 (below) shows the selection of Intel® QSV powered hardware presets through the Preset menu.

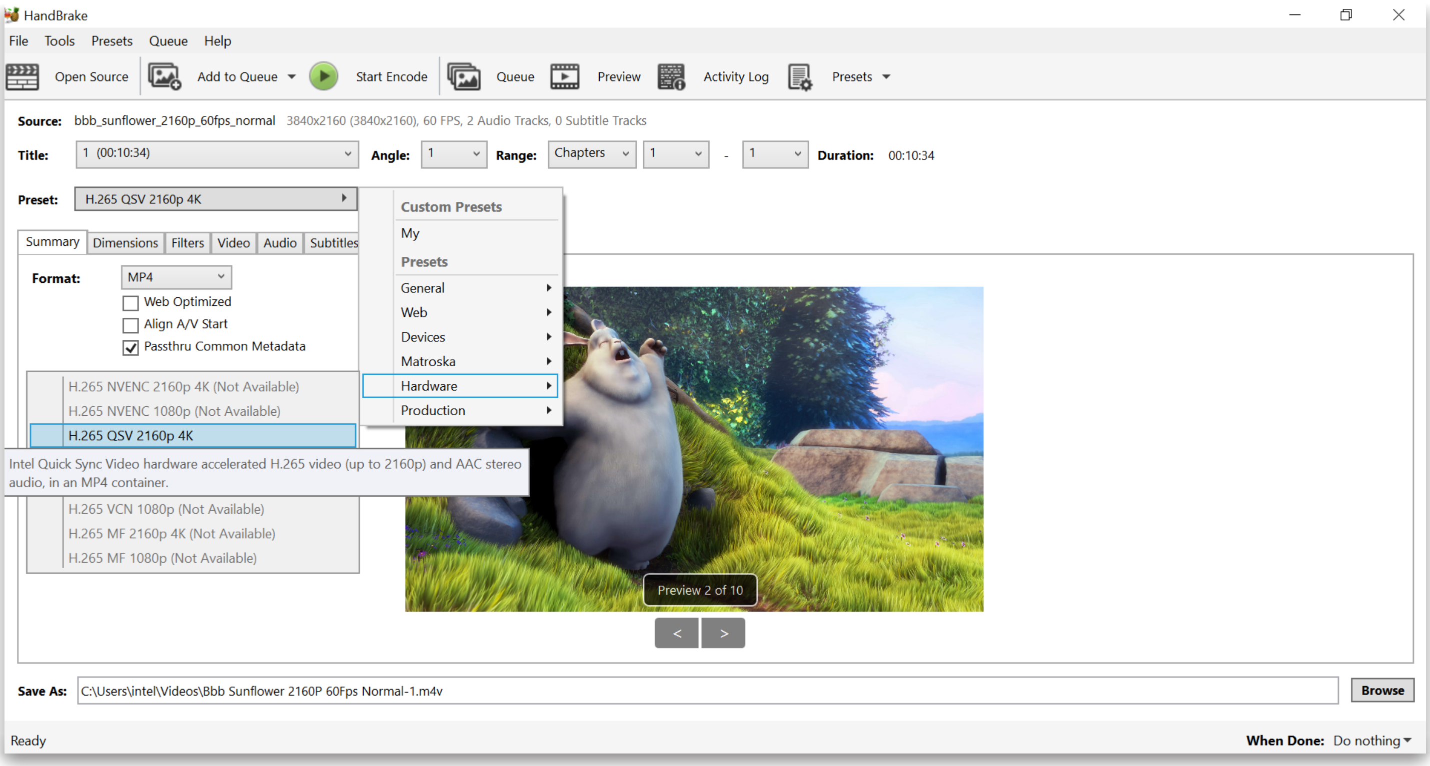


**Figure 5.** Selecting the "H.265 QSV 2160p" hardware preset.

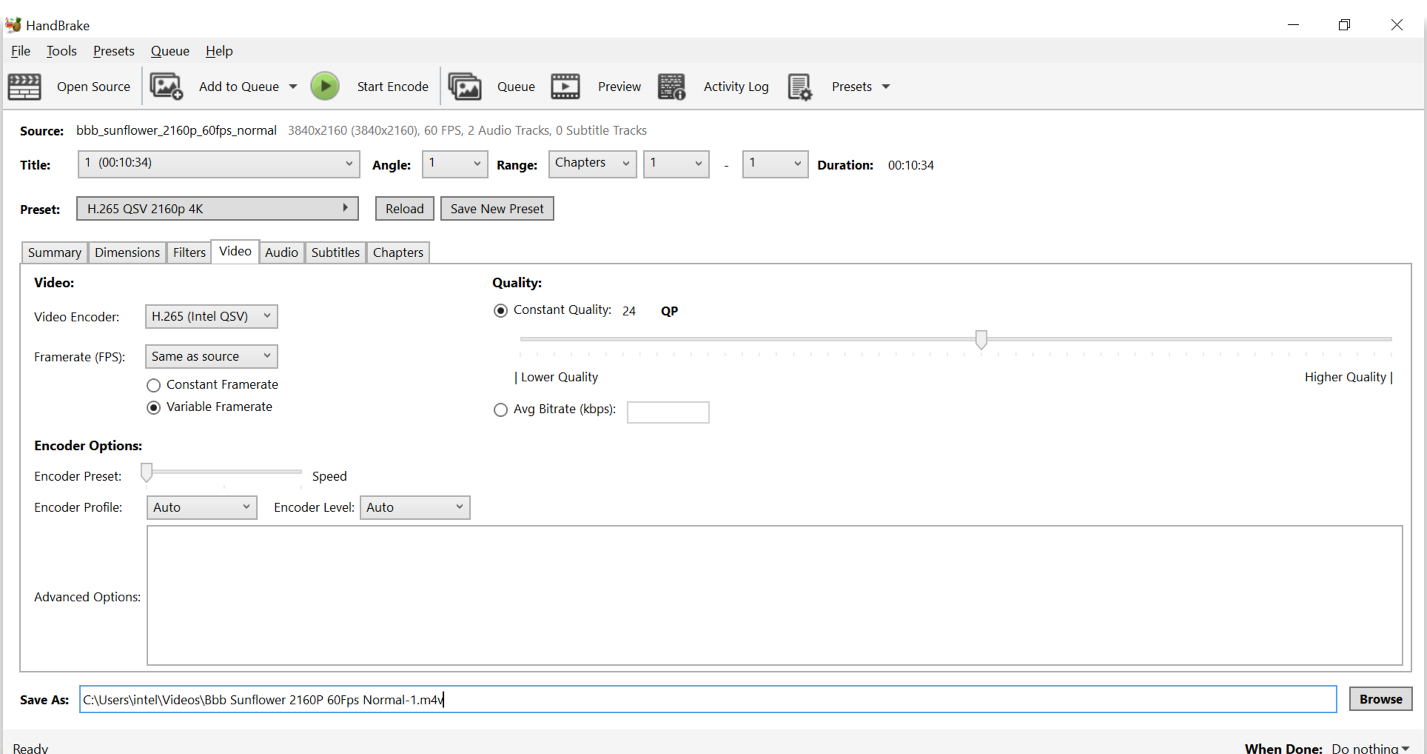


**Figure 6.** The QSV preset settings in the Video tab.

HandBrake users can easily enable QSV functionality when selecting which codec to use. Codecs that have Intel® QSV enabled have parentheses with "Intel QSV" beside them. Then Intel® QSV will be used in the transcoding process. You can see an example of this in Figure 7 below.

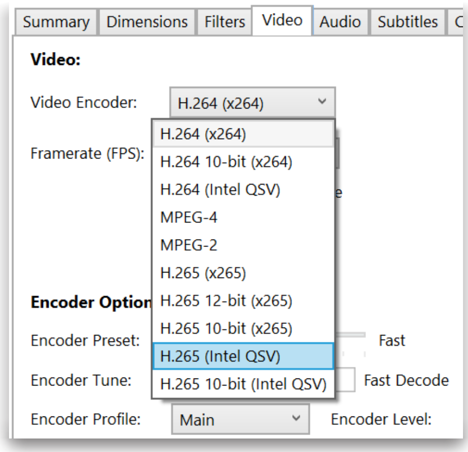


**Figure 7.** The available Video Encoder settings in the Video tab.

The following Intel® QSV encoders are enabled in HandBrake:

- H.264 (Intel QSV)
- H.265 (Intel QSV)
- H.265 10-bit (Intel QSV)

The following have been available since Intel® Arc™:

- AV1 (Intel QSV)
- AV1 10-bit (Intel QSV)

Intel® QSV allows HandBrake to better use the compute already available on your machine by accelerating performance and decreasing power consumption. This is done by offloading most of the transcoding pipeline from the CPU to the media processing part of the GPU. HandBrake is able to transcode video data at a significantly higher speed using Intel® QSV on a GPU, compared to doing the transcoding on the CPU.

**Intel® Deep Link Hyper Encode**

If HandBrake can perform with greater speed and efficiency with a single GPU, imagine what it can do with two. Using Intel® Deep Link, HandBrake can combine workloads on both the integrated GPU (available on certain Intel® Core processors) and the separate, discrete GPU (like Intel® Arc™). Because encoding is so compute intensive, this combination makes for much faster video transcoding and video processing.

By installing and using HandBrake software on a Deep Link-enabled machine, significant time and power savings can be realized thanks to the use of two GPUs. This is in part due to Deep Link's ability to understand and handle the full system compute requirements and manage the sharing of workloads between all available processors. By splitting the workload between the iGPU and the dGPU, transcoding is performed much faster than when being handled by the iGPU or dGPU alone, as shown in Figure 8 below.

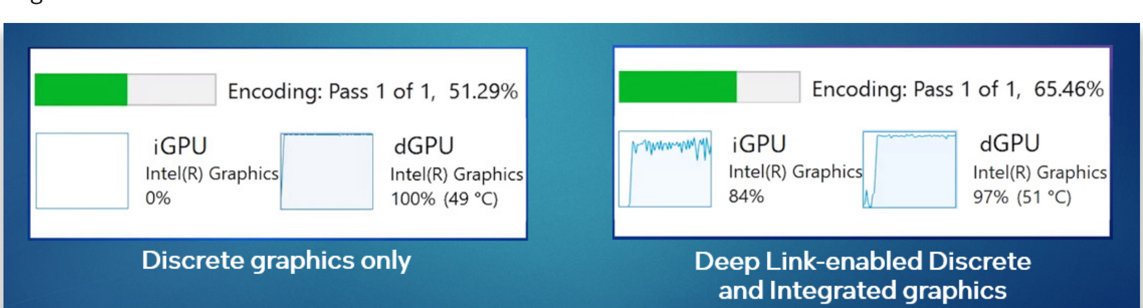


**Figure 8.** Single GPU transcode (left) and Intel® Deep Link Hyper Encode (right) in action.

Intel® Deep Link offers two key methods to boost encoding speed using multiple GPUs: single-stream acceleration and multi-stream acceleration.

### Single Stream Acceleration Mode

In single stream acceleration mode, a Deep Link-enabled application works to split the transcoding of a single file across multiple GPUs. This allows for faster processing of one of the most common tasks undertaken by HandBrake for its users: transcoding video to a specific file format for use in a particular application.

As shown in Figure 9 (below), hyper encoded tasks are split between the dGPU and iGPU.

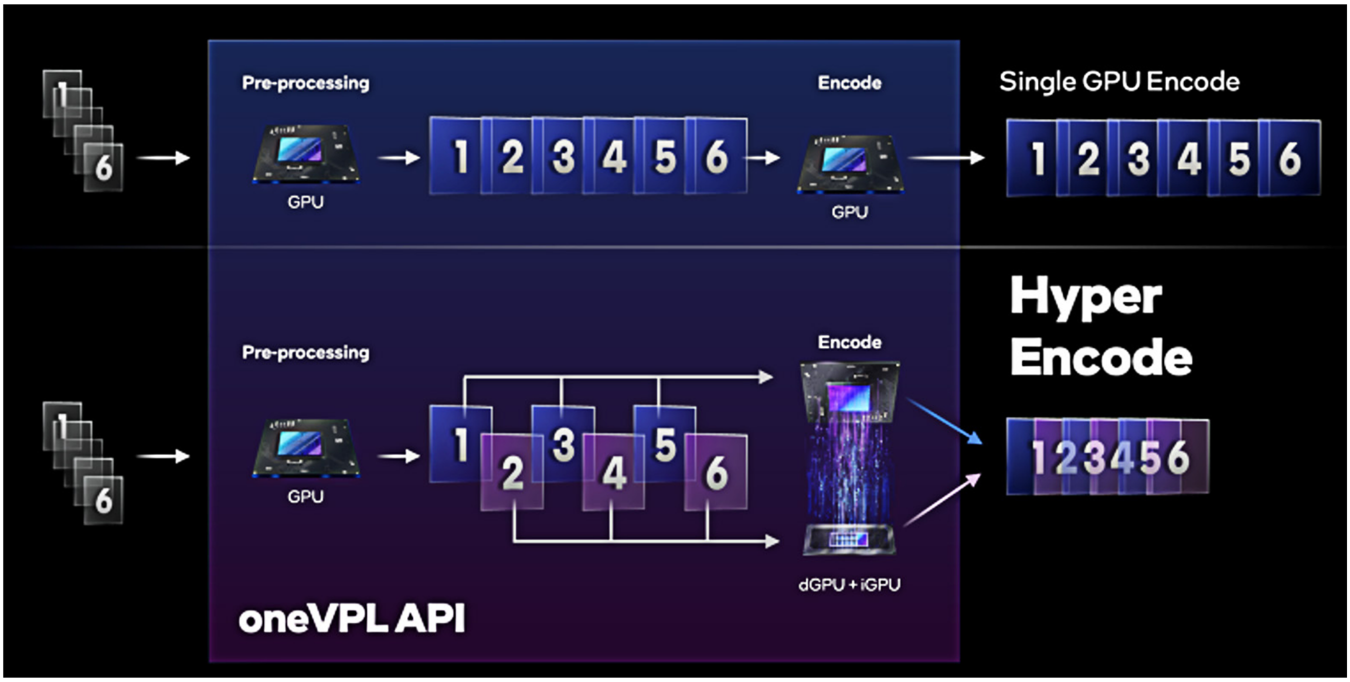


**Figure 9.** Intel® Deep Link Hyper Encode splits the workload between two units.

Encoding on a Deep Link-enabled system shows faster encoding times and more frames-per-second (FPS) than systems without Deep Link enabled. A test[2] (shown in Figure 10 below) showed that a Deep Link-enabled system achieved 65% faster transcoding the same workload than with a dGPU alone. When the workload is encoder bound, more gains can be realized.

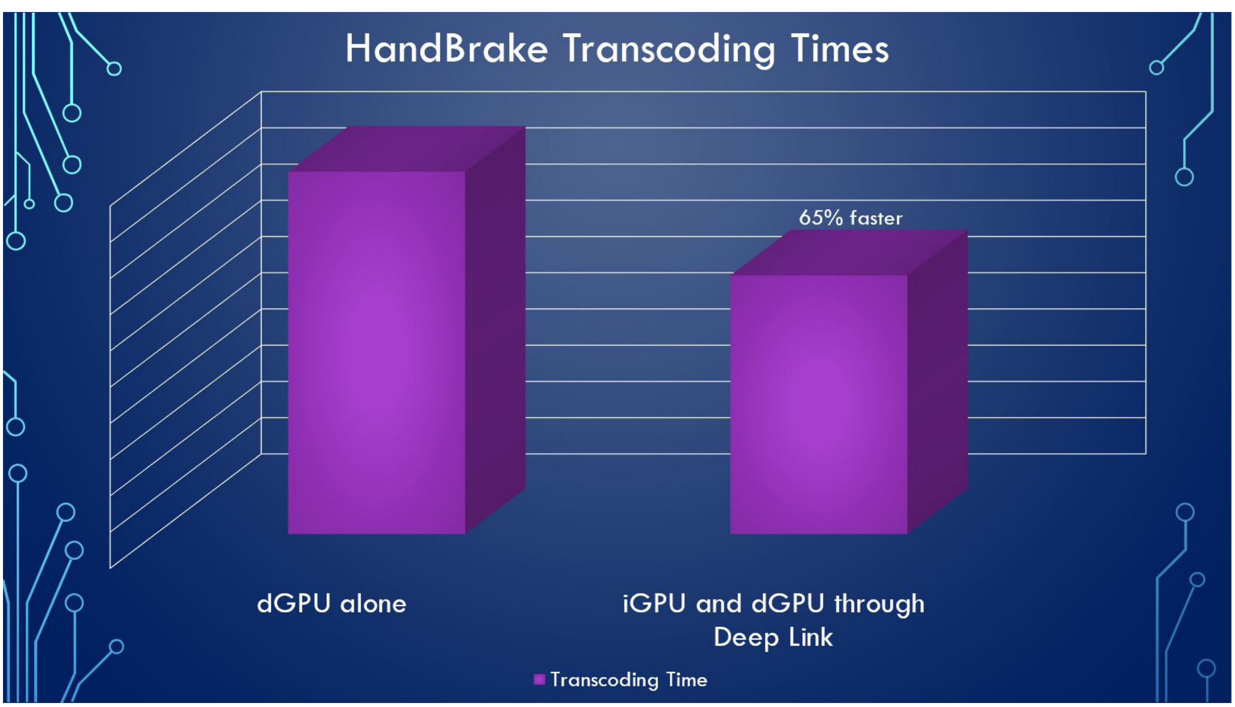


**Figure 10.** Transcoding with Deep Link lessens processing times up to 65%.

2 - This internal test was done on HandBrake using an Intel® Arc™ A380 dGPU and Intel® UHD Graphics 770 iGPU using an Intel® Core™ i5-12600K processor. For more information, please go to www.index.com/PerformanceIndex, "Intel® Arc™ Graphics", and look for the information under "Period" at May 23-27.

Figure 11 (below) shows video hardware acceleration options using the Preferences section in the file area. Use the following steps to enable Hyper Encode on a Deep Link-enabled system in HandBrake:

1. Click on the Video section within the Preferences menu.
2. Select "Allow use of the Intel QuickSync Encoders" and "Enable Low Power QuickSync Hardware (where supported)."
3. Select the "Enable QuickSync Deep Link Hyper Encode".
4. Select the "Prefer use of Intel QuickSync for decoding video when available" under Decoding.

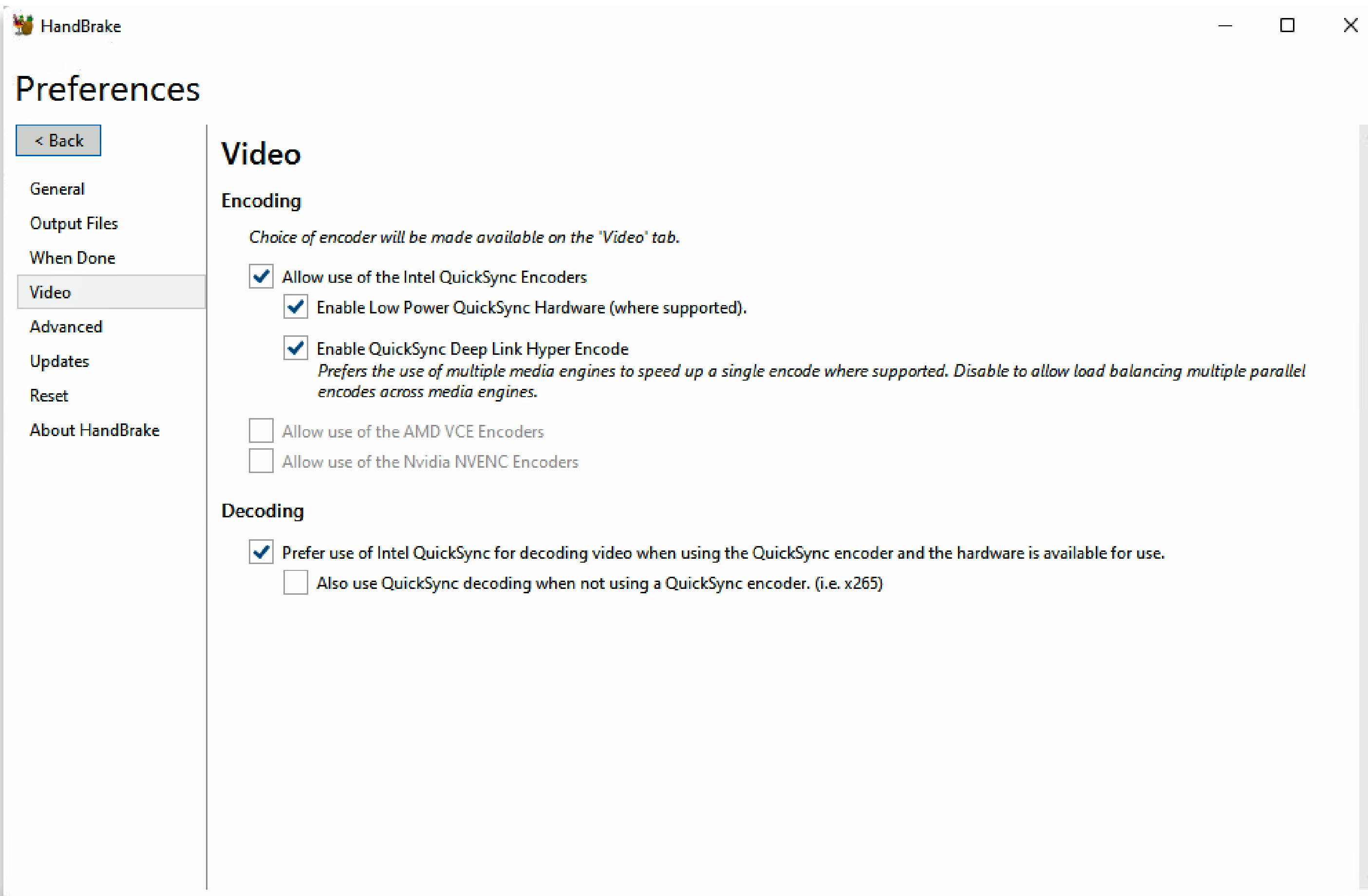


**Figure 11.** Enabling Hyper encoding with Deep Link in HandBrake preferences.

### Multi-Stream Acceleration Mode

Similar to single stream acceleration mode in some respects, the multi-stream acceleration mode splits a multi-file workload between the two GPUs by assigning whole files to each GPU. Video specialists often need to transcode multiple (a batch of) files instead of just one. This is done in HandBrake through batch scan. This allows a batch transcoding of multiple files all at once, which is faster than even the single stream acceleration mode mentioned above. With this time savings, video specialists can transcode more files in a shorter amount of time, allowing them to accomplish their tasks faster and more efficiently.

### Intel® oneVPL

HandBrake implements Intel® QSV and Intel® Deep Link Hyper Encode technologies through an API called the Intel® oneAPI Video Processing Library (oneVPL). oneVPL is part of Intel®'s oneAPI Spec, and is designed to fine-grain tuning and control of the hardware through this API by accessing any and all compute available. oneVPL is Open Source and open spec. This allows developers to use oneVPL API either via their own implementation or to integrate Intel® implementation into their individual source code. oneVPL is used to implement all the Intel® QSV functionality mentioned in this paper into HandBrake.

Finally, Intel®'s commitment to scalability is well represented in oneVPL. oneVPL can be deployed and scaled across multiple different units of compute to all Intel® XPUs[3] . This scalable design means that a developer could start coding today for current Intel® processors and utilize that same code in the future on new Intel® hardware as it is introduced.

3 - XPU is Intel® shorthand for any component that offers processing capabilities. The XPU moniker is meant to include CPUs, GPUs and VPUs, but also includes ASICs and FPGAs as well. oneVPL works with available processors and compute engines to provide the best performance possible.

Intel® wants to help developers implement the same Deep Link benefits that HandBrake has implemented, into their own applications. We are making this happen by making our development libraries Open Source and available for developers.

- To get started using oneVPL, check out the homepage for oneVPL.
- Source code for oneVPL is available on the GitHub oneVPL page.
- Source code for HandBrake is available on the GitHub HandBrake page.

**Intel® QSV Codec Support**
The following chart shows the codecs which currently support oneVPL and provide hardware acceleration:

- *D* stands for Hardware Decoding
- *E* stands for Hardware Encoding

| Codec | 10th Generation Intel Core Processors with Intel Iris Plus graphics or Intel UHD Graphics | 11th and 12th Generation Intel Core Processors with Intel Iris Xe Graphics or Intel UHD Graphics | Intel Iris Xe Discrete Graphics | Intel Iris Xe Discrete Graphics with Deep Link Hyper Encode | Intel Arc A-Series Graphics | Intel Arc A-Series Graphics with Deep Link Hyper Encode |
|---|---|---|---|---|---|---|
| AVC 8-bit 420 | D/E | D/E | D/E | E | D/E | E |
| MPEG-2 | D/E | D/E | D/E | | D | |
| VC-1 | D | D | D | | D | |
| JPEG | D/E | D/E | D/E | | D/E | |
| HEVC 8-bit | D/E | D/E | D/E | E | D/E | E |
| HEVC 8-bit 422 | D/E | D/E | D/E | E | D/E | E |
| HEVC 8-bit 444 | D/E | D/E | D/E | E | D/E | E |
| HEVC 10-bit | D/E | D/E | D/E | E | D/E | E |
| HEVC 10-bit 422 | D/E | D/E | D/E | E | D/E | E |
| HEVC 10-bit 444 | D/E | D/E | D/E | E | D/E | E |
| HEVC 12-bit | | D | D | | D | |
| HEVC 12-bit 422 | | D | D | | D | |
| HEVC 12-bit 444 | | D | D | | D | |
| VP9 8-bit | D/E | D/E | D/E | | D/E | |
| VP9 8-bit 444 | D/E | D/E | D/E | | D/E | |
| VP9 10-bit | D/E | D/E | D/E | | D/E | |
| VP9 10-bit 444 | D/E | D/E | D/E | | D/E | |
| VP9 12-bit | | D | D | | D | |
| VP9 12-bit 444 | | D | D | | D | |
| AV1 8-bit | | D | D | | D/E | |
| AV1 10-bit | | D | D | | D/E | |

**Table 1.** Table listing codecs that are supported by oneVPL.

## Best Known Methods

HandBrake is used by a wide variety of different users for a wide variety of use cases. However, some methods seem to work better than others. Below are some Best Known Methods (BKMs) for using HandBrake on Intel® Architecture.

1. Utilize a processor that is Intel® 6th Generation (Skylake) or newer. To use Deep Link features mentioned in this paper, you need a Deep Link-enabled system, available on Intel® 11th and 12th Core Processors paired with Intel® dGPUs.
2. Have your system running either Windows (version 10 or later) or Linux.
3. Have the latest Intel® Graphics drivers installed, available here. Many related issues can be resolved simply by updating the GPU drivers.

4. Keep HandBrake as up-to-date as possible. We recommend using HandBrake 1.5.1 or later. HandBrake is released in two different configurations: a regular release and a development build. For the best user experience, use the most recent regular release. For the newest features and optimizations including Intel® Deep Link Hyper Encode, consider using the development build.
5. Enable and use Intel® QSV when encoding and decoding:
    a. Select and enable QSV on the Video Preferences menu (see Figure 4 for details).
    b. Select Intel® QSV powered hardware presets through the Preset menu (see Figure 5 for details).
    c. Select QSV when selecting Video Encoder (see Figure 7 for details).

## Conclusion

The ability to compress video files and transcode file types is in heavy demand, making HandBrake an increasingly essential tool for any video specialist. By using HandBrake, users of all skill levels can meet the size, format, and performance requirements that exist for their video files. Intel® is working hard to make that process faster and more efficient for all users, and to provide the developer community with the tools that they need to incorporate these advances into their real-world applications.

With Intel® Deep Link, Intel® QSV, and oneVPL, Intel® is helping to build an innovative hardware and software acceleration solution that can take HandBrake - *and your application* -to the next level by unlocking the new streams of computing capability available on Intel®-powered systems.

## Links for Further Reading

HandBrake is an amazing tool we recommend developers and video specialists continue to use, become more familiar with, and have in their arsenal. To read more about HandBrake, check out the HandBrake website.

To see Deep Link in action, check out our video on YouTube.

To read more about Intel® Quick Sync Video and how to enable the most compute for your video encoding, transcoding, and decoding needs, check it out the Intel® website here.

You can see what codecs are supported by oneVPL on our website.

Currently using MediaSDK? Read about how to upgrade from MediaSDK to oneVPL at this link.

This paper is the fourth in the Deep Link series. The Deep Link main site can be found here, while more information on Deep Link tools and processes covered by the previous three whitepapers can be viewed using the following links:

- Client Artificial Intelligence (AI) Using Intel GPUs (use case: Topaz Gigapixel AI)
- Increasing Workload Throughput Using Intel GPUs (use case: Leela Chess Zero)
- Accelerating the Inferencing Process Using Intel GPUs (use case: CyberLink Power Director)

NOTE: All screenshots and instructions are taken from a HandBrake development build from April 2022 and may change.

intel®