WHITEPAPER

REMOTING PROTOCOLS FOR GPU-ACCELERATED, VDI ENVIRONMENTS

Choose wisely to optimize the end-user experience

By
Simon Schaber
Nachiket Karmarkar

MARCH 2019

NVIDIA.

CONTENTS

EXECUTIVE SUMMARY

Because user experience is the key to VDI success, many enterprise organizations are choosing to add GPUs to traditional CPU-only VDI environments to improve the performance of graphics-rich applications and operating systems. Everything from general office applications on Windows 10 to specialized design applications for architects, engineers, and other visual artists can benefit greatly from GPU acceleration.

When using GPUs in a VDI environment, however, it's important to select the correct remoting protocol to optimize image quality and bandwidth consumption. Perceived issues with GPU-enabled desktops are often the result of remoting protocols that aren't optimized for a given use case.

This paper provides an overview of nine different VDI policy sets from Citrix and their impact on the user experience. You'll learn how to choose the right protocols for specific end-user needs.

POLICY OPTIONS

Using Citrix Virtual Desktop (formerly XenDesktop) 7.18 with GPU acceleration, IT administrators have many different video codec policies to choose from. Figure 1.0 outlines nine of these policies, showing the regions impacted, the resultant visual quality, and hardware encoding requirements for each one.

In this paper, we'll explore some of these policies in greater detail, explain how to set them up, and make recommendations about when to use them.

Protocol & Codecs

Citrix XenDesktop 7.18

Video Codec Policy	Region	Visual Quality	CODECS Used	HW ENC*
Do Not Use	Region optimized	Medium	Static: JPEG (90) + 2D/MDRLE Video: Adaptive JPEG (10-65)	No
For Entire Screen	Entire Screen	Medium	H.264 4:2:0	Yes
For act, charging regions	Region optimized	Medium	Static: JPEG (90) + 2D/MDRLE Video: H.264 4:2:0	Yes
H.264 + Text Optimization*	Entire Screen	Medium	H.264 4:2:0 + Loseless Text	No
For Entire Screen	Entire Screen	Build To Loseless	H.264 4:2:0 during activity, 2D/MDRLE when stationary	Yes
For Entire Screen	Entire Screen	Visual Loseless: Medium	H.264 4:4:4	Yes
For Entire Screen (H.265)	Entire Screen	Medium	H.264 4:2:0	Yes
For act, charging regions (H.265)	Region optimized	Medium	Static: JPEG (90) + 2D/MDRLE Video: H.265 4:2:0	Yes
For act, charging regions (H.265)	Entire Screen	Build To Loseless	H.265 4:2:0 during activity, 2D/MDRLE when stationary	Yes

* no policy available for TextOpt

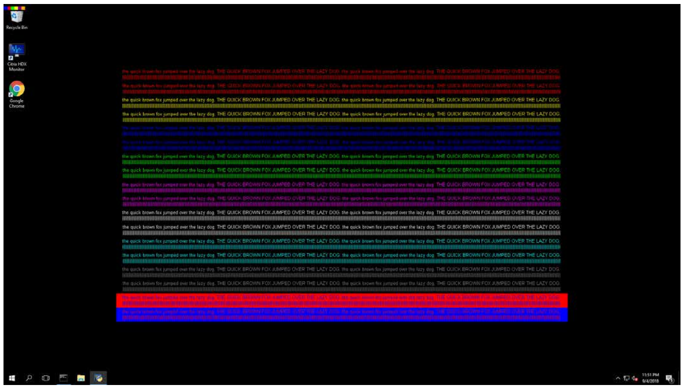
* videocodec (H.264 /H.265) part via NVENC

Figure 1: To optimize the end-user experience in Citrix Virtual Desktop 7.18,
IT administrators must decide between nine different video codec policy choices.

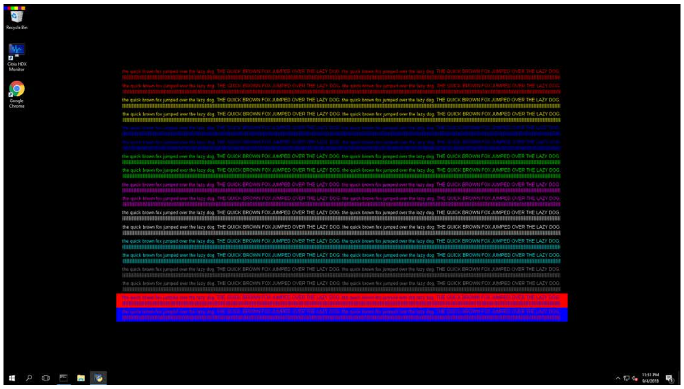
BITMAP REMOTING VS. VIDEO CODECS

Bitmap remoting, also known as "Thinwire," is the best possible protocol for static content as it retains visual quality close to the original image. It's based on JPG compression and RLE (Run Length Encoding), which originated from Citrix.

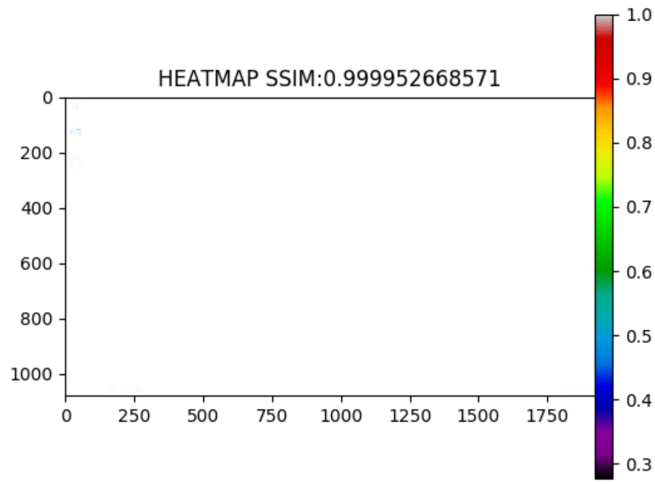
As you can see from the images below, the human eye can't really distinguish the difference between a reference image and one captured by bitmap remoting. The consistency between the two images is confirmed by a structural similarity index (SSIM) heatmap, a perceptual metric that quantifies image quality degradation caused by processing or losses in data transmission.



Reference Image



Captured Image (VDI session)



SSIM Heatmap Result

Bitmap remoting is a good choice for knowledge workers who use office applications or ERP software without many screen changes.

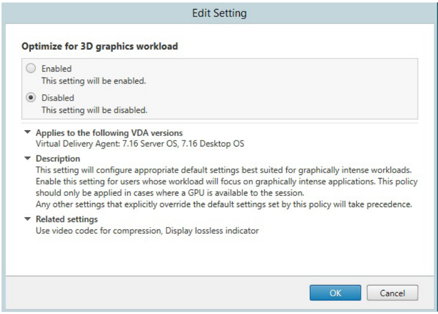
Because bitmap remoting doesn't require any specific hardware at the endpoint, and because bitmap compression requires very little CPU load, it works well on older machines or machines without hardware decoding support.

To "force" Bitmap remoting use the following policy settings:

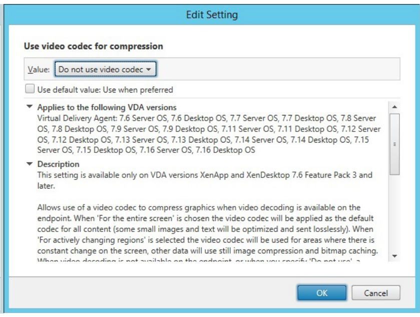
1.1 Optimize for 3D Graphics Workload -> Disabled

1.2 Use Video Codec for Compression -> Do Not Use Video Codec

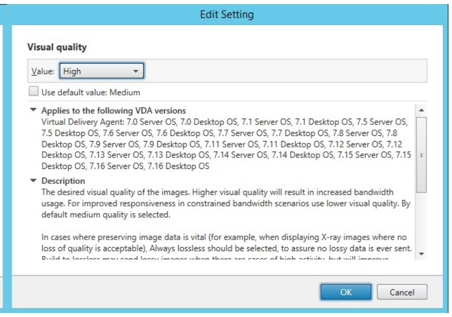
1.3 Visual Quality -> High
Certain use cases may require "build to lossless" or "always lossless" for pixel perfect quality.



1.1



1.2



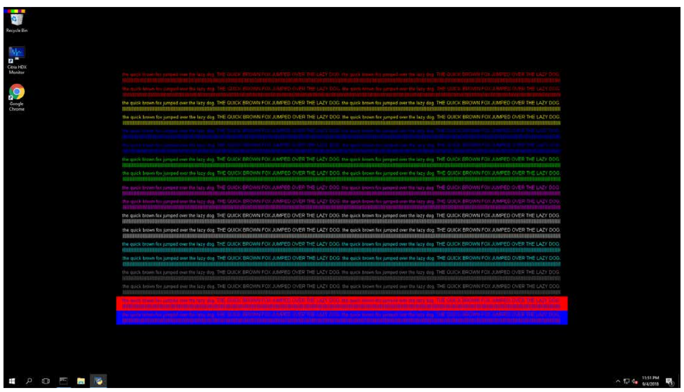
1.3

Video Codecs

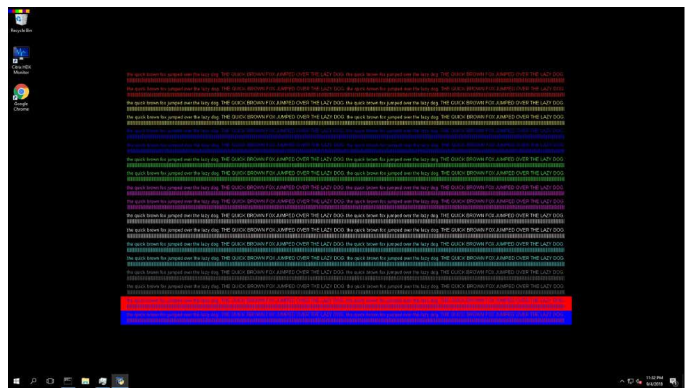
Because bitmap remoting performance degrades with higher frame rates, video codecs are a better choice when moving images are involved. Citrix now supports H.264 and H.265, along with the variations YUV420 and YUV444.

To conserve bandwidth and deliver smooth playback at high framerates, video codecs take advantage of the human visual system's lower acuity for color differences by encoding images with more resolution for luma information than chroma information.

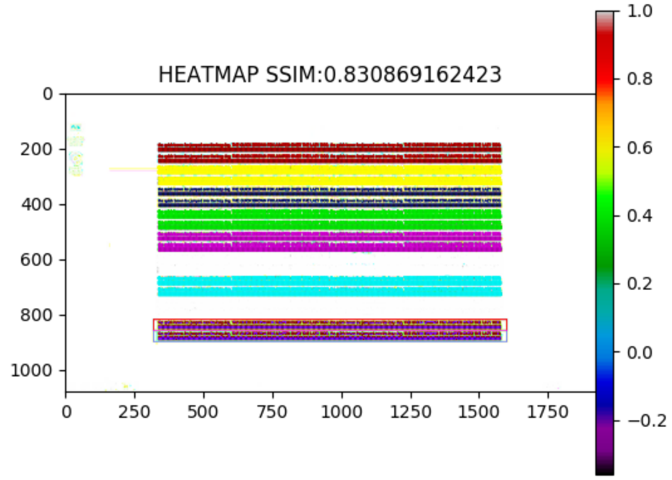
Keep in mind, however, that the chroma subsampling effect associated with video codecs using the YUV420 implementation (the Citrix default) can cause blurriness, especially for text. This is not an issue for video playback of objects, but it can be disturbing when working with office applications. You can see the difference between the reference image and the captured image, and the discrepancy is confirmed by the heatmap result that shows only 83% accuracy between the two images.



Reference Image



Captured image H.264 YUV420



SSIM heatmap result H.264 YUV420

When you use a video codec like H.264, captured images must be encoded on the server side. This requires up to 1 vCPU without hardware encoding. Because Citrix supports hardware encoding, however, you can use NVIDIA Encoding (NVENC) with our NVIDIA Tesla® GPUs to offload this task to specific ASICs on the GPU. Encoding in hardware is always faster than CPU-based encoding, and reduces latency in the range of 25 milliseconds.[1]

Similarly, endpoints need to have a hardware decoder or a modern CPU, starting with the 6th Generation Intel processors (based on Skylake architecture), with adequate power to use video codecs. Because hardware decoding reduces overall latency, thin clients are often unable to provide an adequate end-user experience.

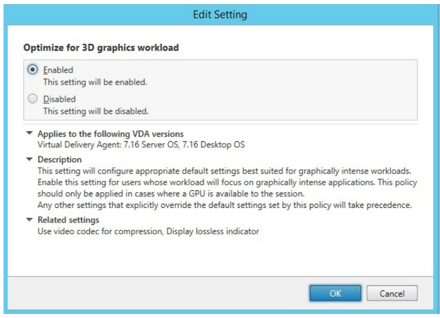
To use video codecs, configure the policies as shown below:

1.1 Optimize for 3D Graphics Workload -> Enabled
Enable this policy to use NVIDIA Direct FB Access (NVFBC) and NVENC.

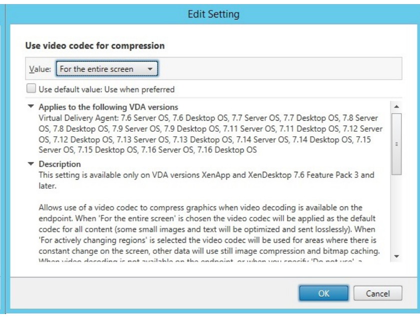
1.2 Use Video Codec for Compression -> For the Entire Screen
For NVENC to work on older Citrix Virtual Desktop versions (7.12 – 7.16), it's necessary to use the entire screen policy.

1.3 Visual Quality -> High
Chroma subsampling may cause blurriness, but choosing "High" delivers adequate quality for most use cases.

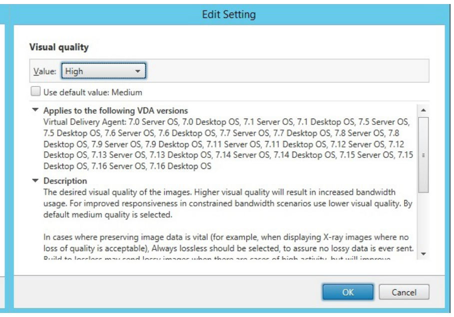
1.4 Use Hardware Encoding -> Enabled
This is the Citrix default setting.



1.1



1.2



1.3

[1] Testing by NVIDIA on 1080p desktop.

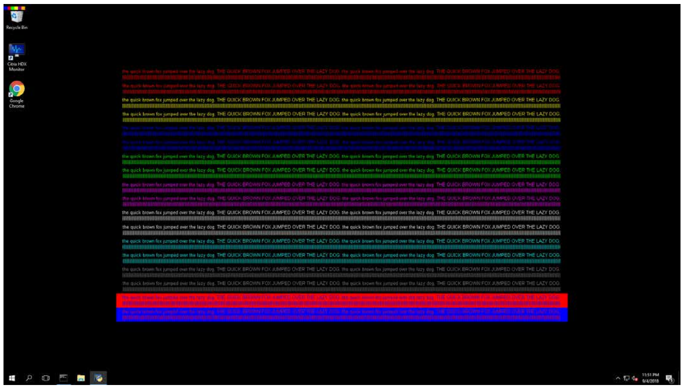
H.264—YUV420 AND YUV444

Many end users in GPU-accelerated VDI environments can't tolerate the blurriness caused by the chroma subsampling effect. This can occur when you use the H.264 video codec for office VDI users.

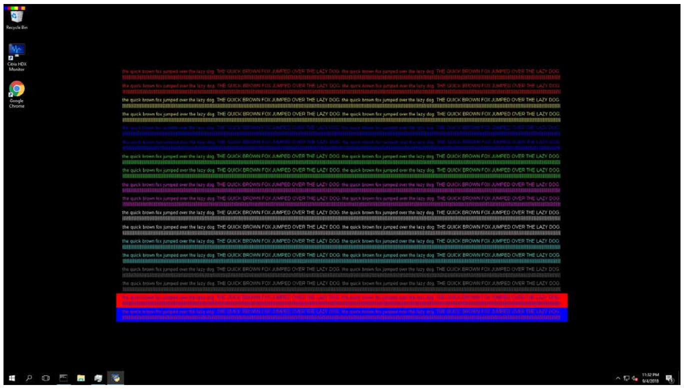
If you set your Citrix policy to "Optimize for 3D workloads," the default protocol is H.264 with entire screen hardware encoding, using the YUV420 sampling system for image compression. With YUV420, half of the vertical and horizontal chroma information is removed to reduce bandwidth requirements.

The other option is YUV444, which doesn't remove any chroma information from your images.

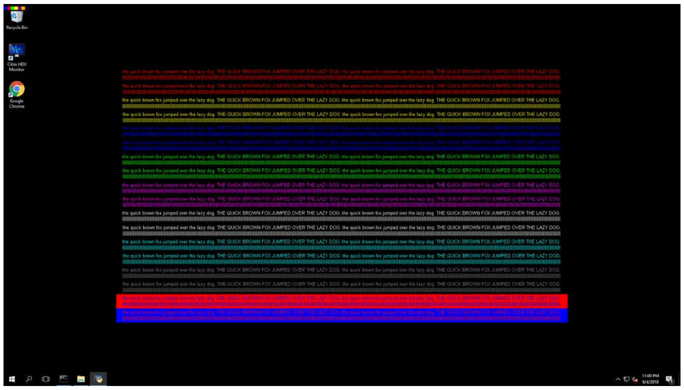
As you can see from the images below, YUV444 provides much better image quality than YUV420, in terms of color accuracy and sharpness.



Reference Image

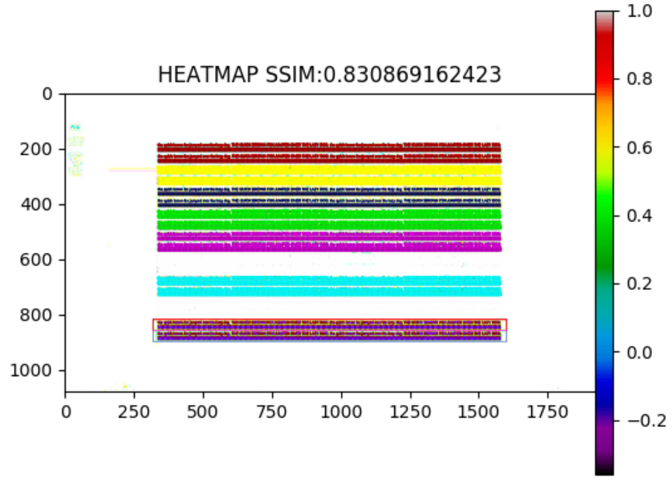


Captured image H.264 YUV420

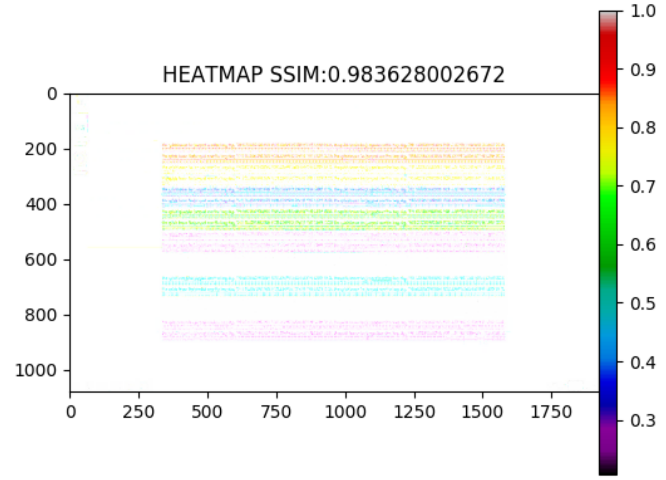


Captured H.264 YUV444 image

On the YUV420-captured image, notice the chroma subsampling effect in the areas where blue type sits on a red background and where red type sits on a blue background. This can appear for users of common office applications, such as Microsoft Excel, when background colors are used.



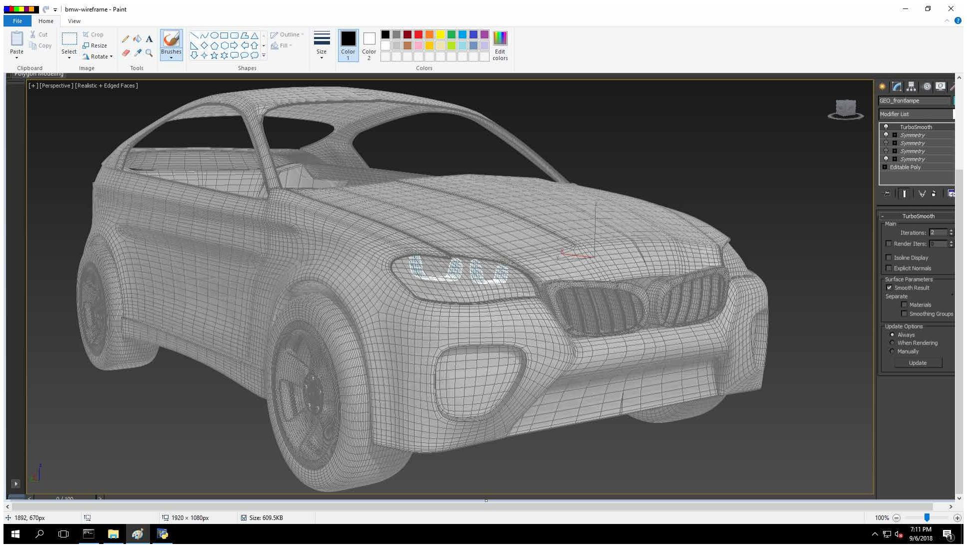
SSIM heatmap result H.264 YUV420



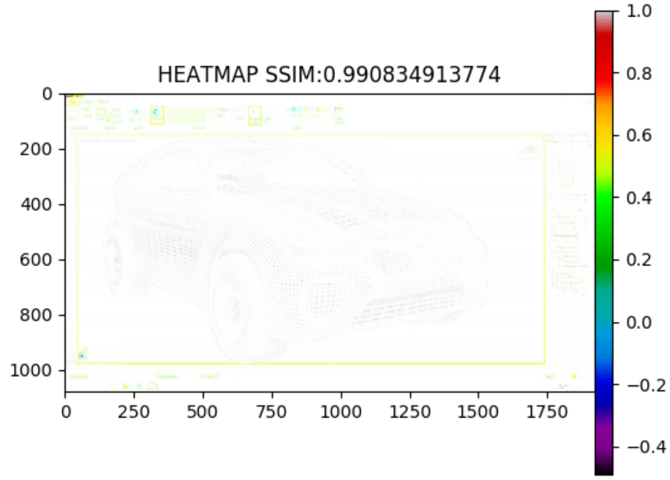
SSIM heatmap result H.264 YUV444

Comparing the SSIM results side by side, we see a huge improvement in color accuracy when using YUV444.

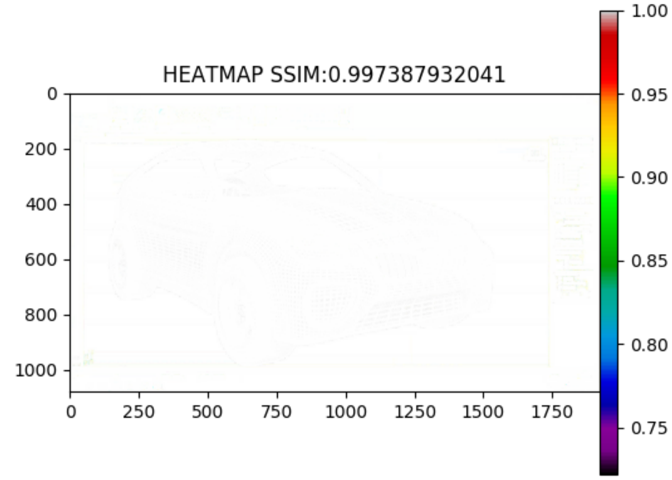
Before you get the impression that YUV420 is unusable, however, consider that the reference image above is essentially a worst-case scenario for that sampling system. If your users work mostly with gray scale images, the distinctions aren't as dramatic as the luminance part of the image is unchanged. In these types of use cases, YUV420 provides 99% color accuracy and YUV444 is only slightly better at 99.7%.



Reference image



SSIM heatmap result H.264 YUV420

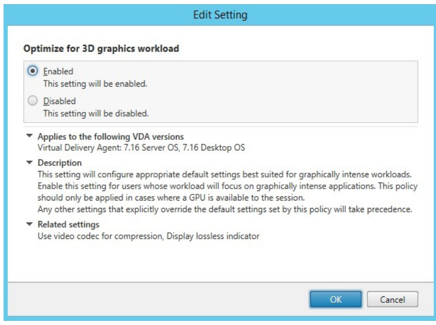


SSIM heatmap result H.264 YUV444

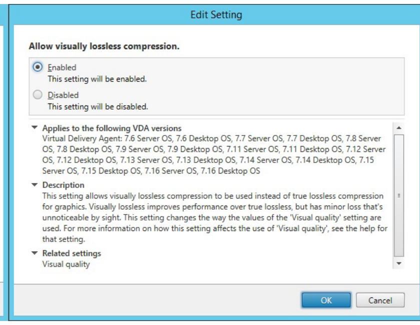
The downside of YUV444, of course, is that it requires much more bandwidth—about two to three times more in multimedia sessions. This can be an important consideration if your end users have low-bandwidth, remote connections. Additionally, Linux-based thin clients do **not** support YUV444.

Assuming you have the bandwidth and aren't stuck with a Linux receiver like Citrix Virtual Desktop 7.15 LTSR, YUV444 is recommended over YUV420 for 3D VDI use cases, especially if users are sensitive to color accuracy or work with single-pixel line drawings.

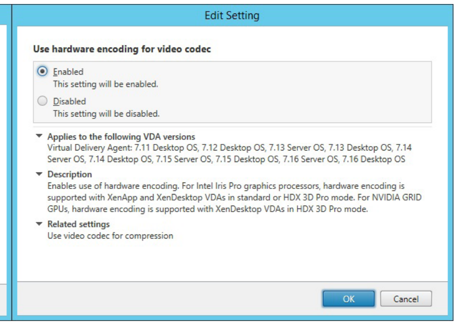
Hardware encoding (NVENC) is supported on both YUV420 as well as YUV444, so both will see good end-user latency. YUV444 will have slightly better end-user latency due to increased encoding capability.



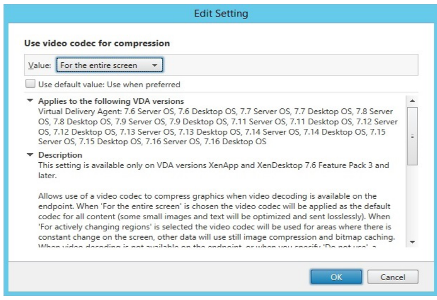
1.1



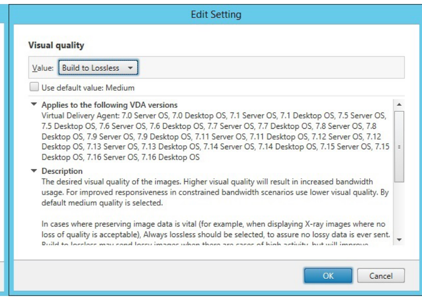
1.2



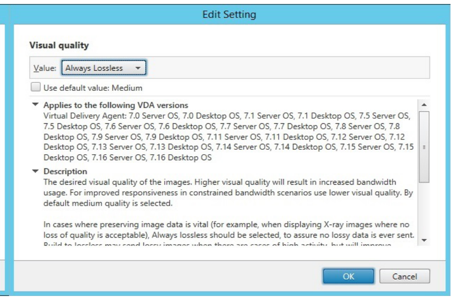
1.3



1.4



1.5



1.6

To configure YUV444 properly, follow the steps below:

1.1 Optimize for 3D Graphics Workload -> Enabled
Enable this policy to use NVFBC and NVENC.

1.2 Allow Visually Lossless -> Enabled

1.3 Use Hardware Encoding -> Enabled

1.4 Use Video Codec for Compression -> For the Entire Screen
For NVENC to work on older Citrix Virtual Desktop versions (7.12 – 7.16), it's necessary to use the entire screen policy.

1.5 & 1.6 Visual Quality -> Build to Lossless or Always Lossless
A lot of people miss this step and wonder why things aren't working as expected.

H.264 VS. H.265

If you can fulfill the prerequisites for H.265 encoding and decoding, you can expect better compression at the same visual quality, or the same quality at lower bandwidth, than you can with H.264.

PREREQUISITES FOR H.265

Server side

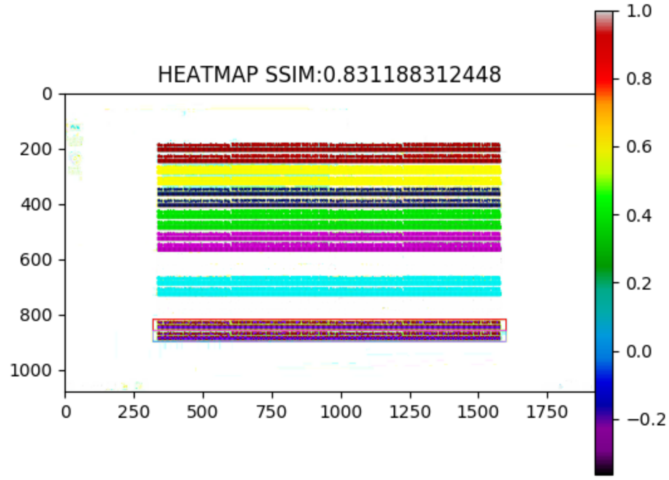
- GPUs based on Maxwell™, Pascal™, Volta™ or Turing™ architecture, which support H.265 encoding. See the complete list of Tesla GPUs for Virtualization here and the encode/decode GPU matrix here
- Citrix licensing (Platinum)
- Correct Citrix policies (see below)

Client side

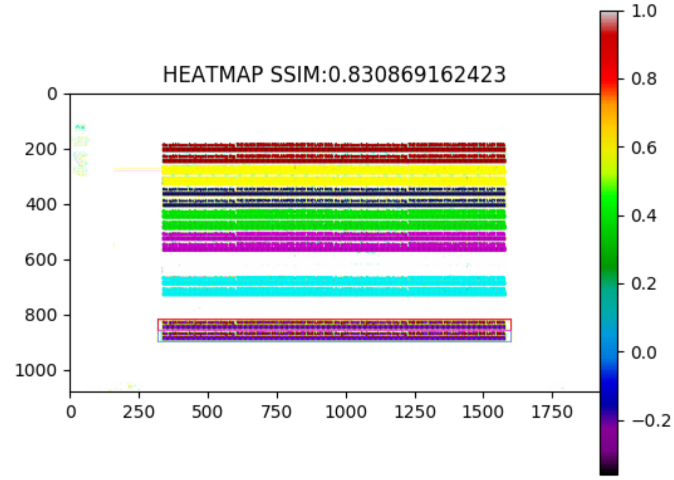
- Endpoints that support H.265 decoding
- Correct policy with the Citrix receiver admx template on your endpoint (Enable H.265 decoding)

To the human eye, image quality looks much better with H.265. But surprisingly, the SSIM heatmap results show very little discrepancy between the reference image and the captured image. This indicates that while blurriness from the chroma subsampling effect is less recognizable with H.265, color accuracy between H.264 and H.265 are about the same.

H.264 YUV420 (left half) / H.265 YUV420 (right half)

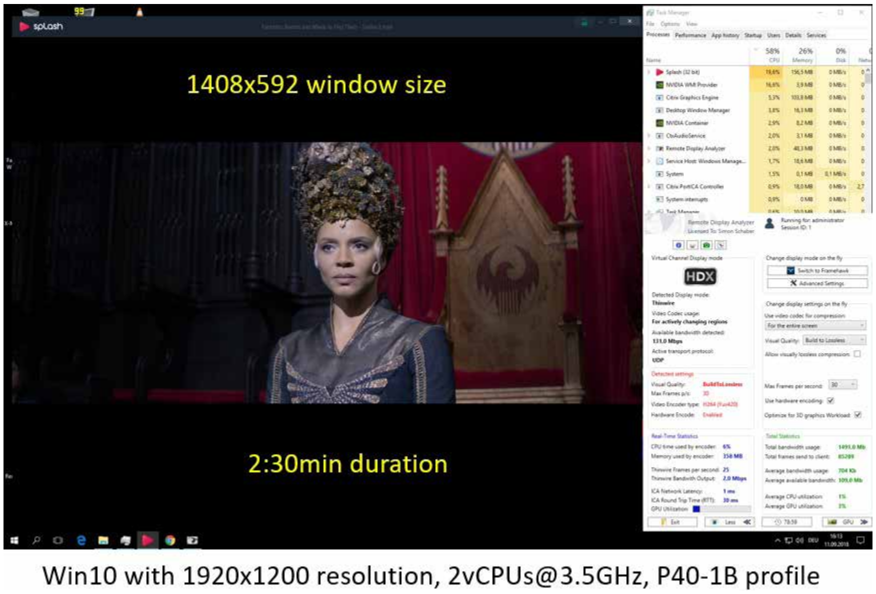


SSIM heatmap result H.265 YUV420



SSIM heatmap result H.264 YUV420

In test cases, H.265 uses about 20% less bandwidth than H.264, which can be helpful in low-bandwidth situations, such as WAN connections.[2]



Video playback in window mode

Bandwidth Comparison

Video playback @ 30fps

CODEC	Visual Quality	Encoder CPU	Total FPS	MB transferred
H.264 YUV420	Medium	2%	3736	220MB
H.265 YUV420	Medium	2%	3766	180MB
H.264 YUV420	High	2%	3719	210MB
H.265 YUV420	High	3%	3780	185MB

[2] Testing by NVIDIA, based on video playback in window mode.

Since both protocols require hardware encoding, there is almost no difference in end-user latency between H.264 and H.265 In tests.

The limiting factor for H.265 is the requirement for endpoint decoding. Very few thin clients support H.265 hardware decoding, and even fat clients must have the right GPU (e.g., using Pascal™ or Turing™ architecture) or CPU (e.g., Skylake).

If you can meet the prerequisites, H.265 is a better choice than H.264 in situations where you have limited bandwidth for 3D VDI workflows.

The policy set for H.265 is the same as the one for H.264. To ensure that H.265 is used, you must import the .admx Receiver template from Citrix and enable the policy. Follow these steps to use H.265:

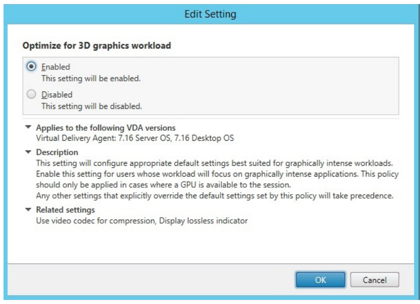
1.1 Local Computer Policy -> Computer Configuration -> Administrative Templates -> Citrix Receiver -> User Experience

1.2 Optimize for 3D Graphics Workload -> Enabled
Enable this policy to use NVFBC and NVENC.

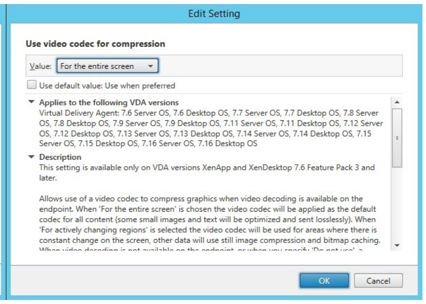
1.3 Use Video Codec for Compression -> For the Entire Screen
For NVENC to work on older Citrix Virtual Desktop versions (7.12 – 7.16), it's necessary to use the entire screen policy.

1.4 Visual Quality -> High
This won't eliminate the chroma subsampling effect, but quality will be adequate for most use cases.

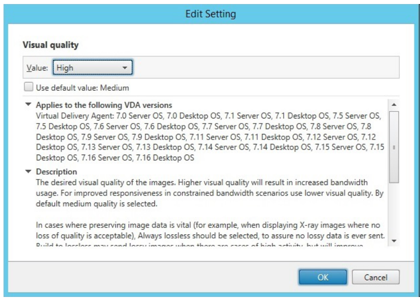
1.5 Use Hardware Encoding -> Enabled



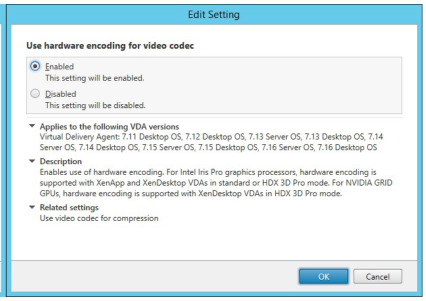
1.2



1.3



1.4



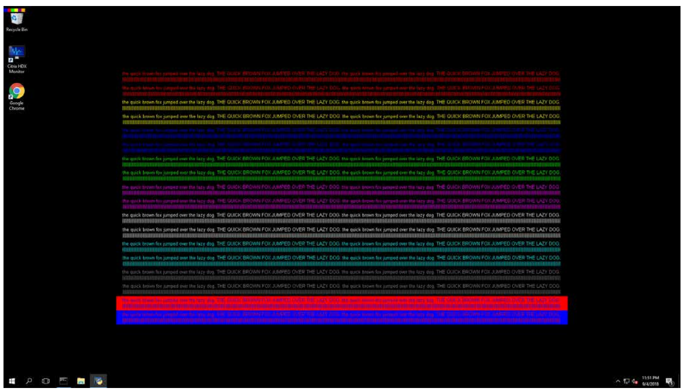
1.5

[2] Testing by NVIDIA, based on video playback in window mode.

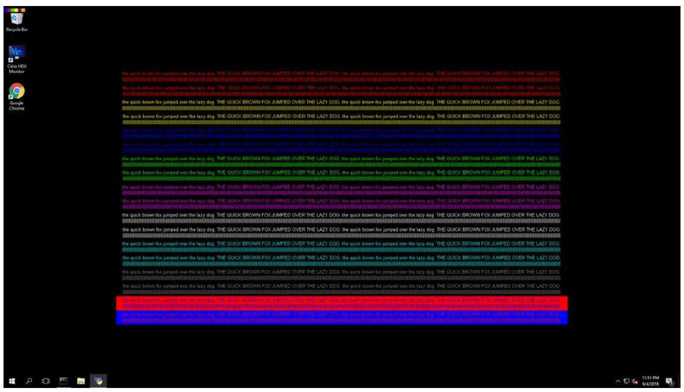
BUILD-TO-LOSSLESS ADVANTAGES

Starting with Citrix Virtual Desktop 7.18, when the user sets a policy for visual quality to “Build to Lossless” (BTL), H.264 or H.265 is used instead of JPEG right after user interaction. Content is initially displayed at low quality to improve overall interactivity, and once user input has stopped, the screen is updated with smaller, transient images for a noticeable sharpening of the desktop image.

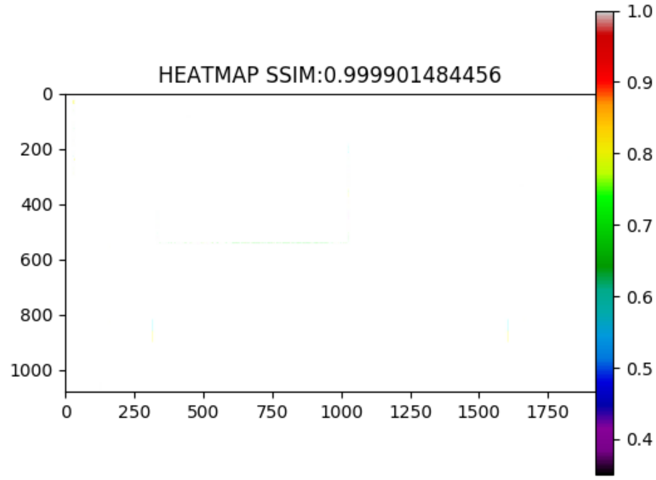
To the human eye, the captured image looks just like the reference image, and the SSIM score reveals that the client-side image is 99.99% identical to the one rendered on VDI desktop.



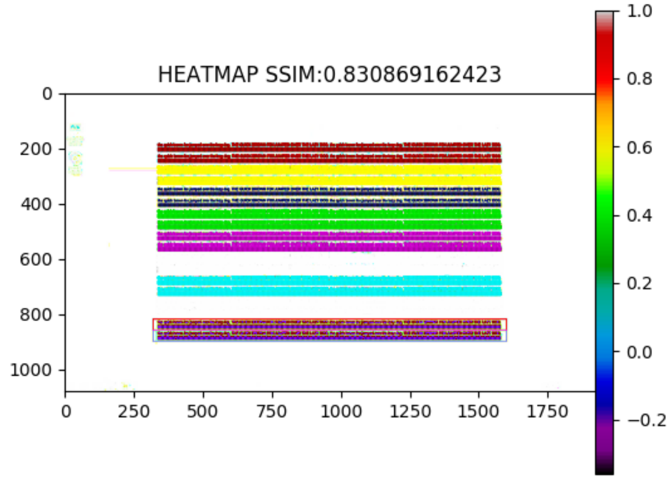
Reference Image



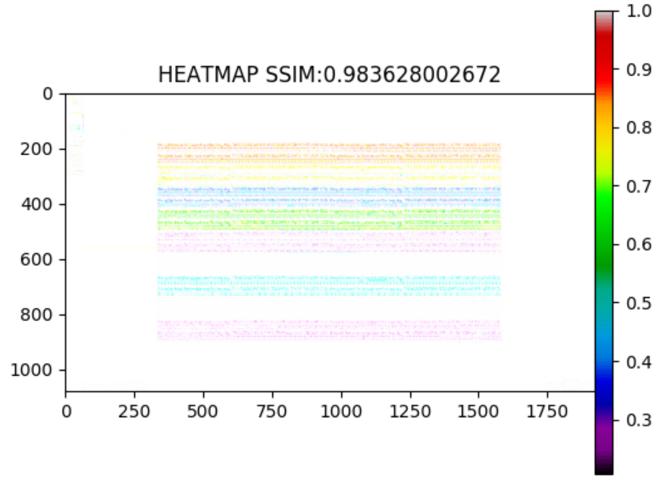
Captured H.264 YUV 420 image with BTL



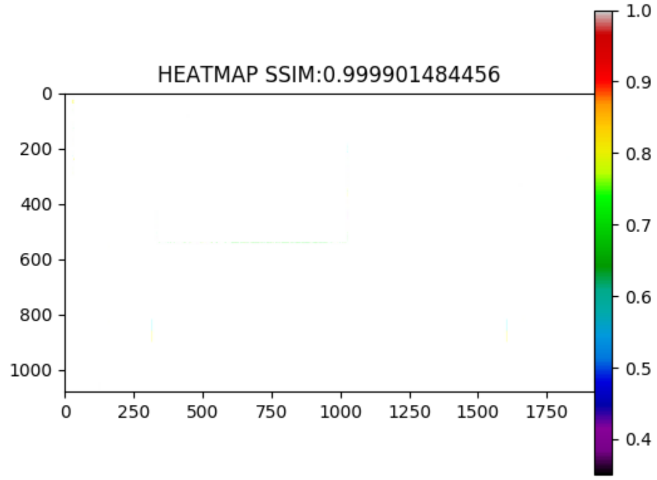
SSIM Heatmap for H.264 YUV 420 with BTL



SSIM heatmap result H.264 YUV420



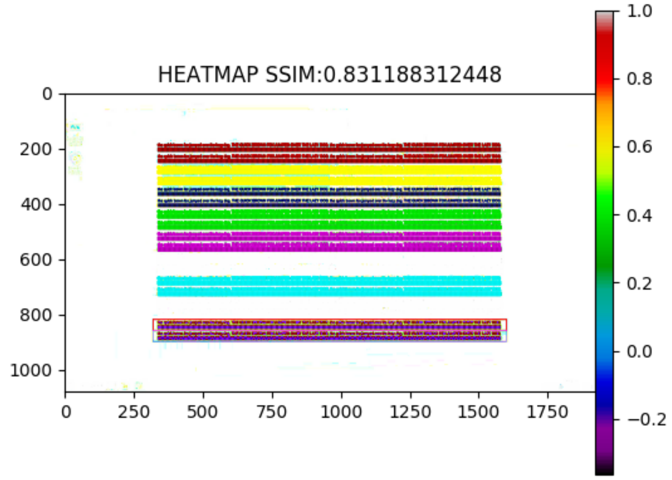
SSIM heatmap result H.264 YUV444



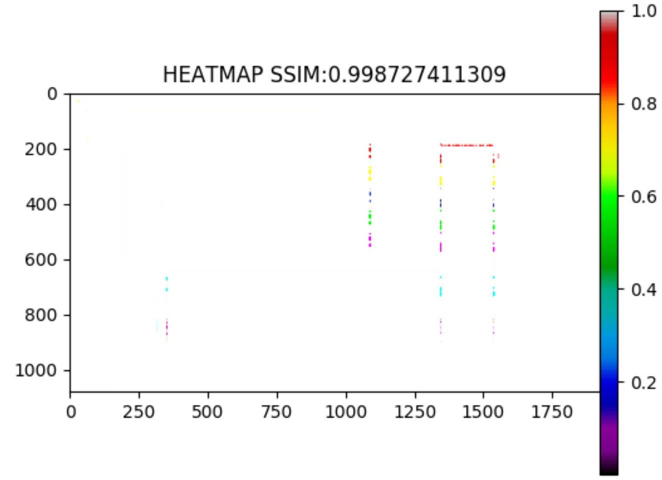
SSIM Heatmap for H.264 YUV 420 with BTL

If we compare SSIM heatmaps across H.264 YUV 420, H.264 YUV 444, and H.264 YUV 420 with build to lossless, build to lossless delivers the best possible image quality.

We can also set visual quality to 'Build-to-lossless' in combination with the H.265 codec and get the same great results:



SSIM heatmap result H.265 YUV 420



SSIM Heatmap for H.265 YUV 420 with BTL

Establishing bandwidth utilization with the bitmap codec as our baseline, we tested a 720p video playback in windowed mode to see the ramifications of BTL on bandwidth consumption. Using the H.264 video codec with BTL resulted in a nearly 45% bandwidth savings.

In comparison, H.264 YUV 420 with visual quality set to medium only reduced bandwidth utilization by about 11%. The biggest difference was observed when H.264 YUV 444 with BTL was used. In this case, BTL demonstrates a whopping 70% savings in bandwidth utilization.

CODEC	Visual Quality	Encoder CPU	Total FPS	MB transferred
Bitmap JPG/RLE	Medium	7%	3693	355MB
H.264 YUV420	Medium	2%	3736	220MB
H.264 YUV444	Medium	3%	3728	655MB
H.264	Build To lossless	5%	3642	195MB
Bitmap JPG/RLE	High	8%	3633	610MB
H.264 YUV420	High	2%	3719	210MB
H.264 YUV444	High	4%	3716	690MB

Table 1: Bandwidth Comparison Between Bitmap Codec, H.264, and H.264 with BTL

Moving over to the H.265 video codec, we see that adding the BTL feature yields about 50% lower bandwidth utilization when compared with the bitmap codec with visual quality set to medium. When the visual quality is set to high, the bandwidth savings jumps to about 71%.

CODEC	Visual Quality	Encoder CPU	Total FPS	MB transferred
Bitmap JPG/RLE	Medium	7%	3693	355MB
H.265 YUV420	Medium	2%	3766	180MB
H.265	Build To Lossless	5%	3796	175MB
Bitmap JPG/RLE	High	8%	3633	610MB
H.265 YUV420	High	3%	3780	185MB

Table 2: Bandwidth Comparison Between Bitmap Codec, H.265, and H.265 with BTL

While ‘build to lossless provides better image quality with lower bandwidth requirements, users may need to adjust to the sharpening effect that occurs when a user is idle. This is the best possible compromise between visual quality, performance, and bandwidth consumption, which ultimately leads to the best overall user experience.

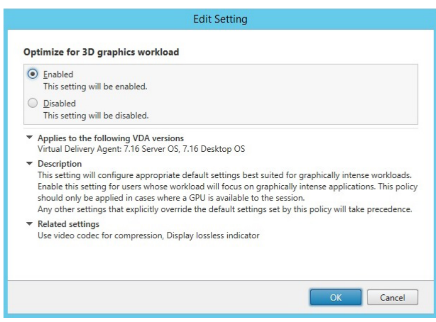
To enable the BTL feature, set the following policies:

1.1 Optimize for 3D Graphics Workload -> Enabled
Enable this policy to use NVFBC and NVENC.

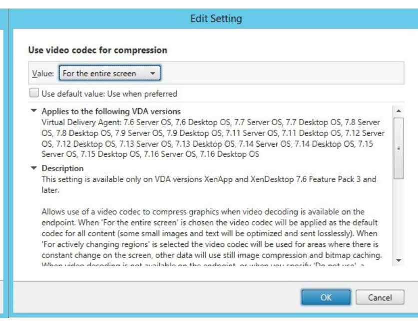
1.2 Use Video Codec for Compression -> For the Entire Screen

1.3 Visual Quality -> Build to Lossless

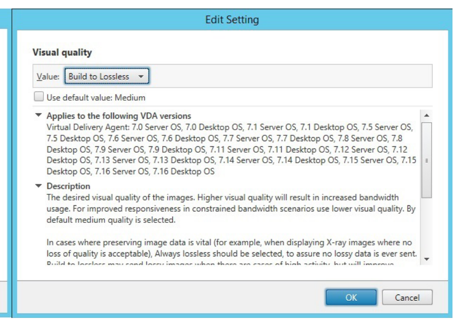
1.4 Use Hardware Encoding -> Enabled



1.1



1.2



1.3

MIXED PROTOCOL ADVANTAGES

Combining bitmap and video codecs lets you use the best available codec for specific screen regions. Also known as active changing regions (ACR) or selective H.264/H.265, this scenario uses the bitmap codec as the default and enables the H.264 or H.265 video codecs to kick in whenever, and wherever, moving images are detected.

Because video codecs apply only to specific screen regions with ACR, bandwidth consumption will be slightly less than it would be if you were using the H.264 or H.265 video codecs across the entire screen.

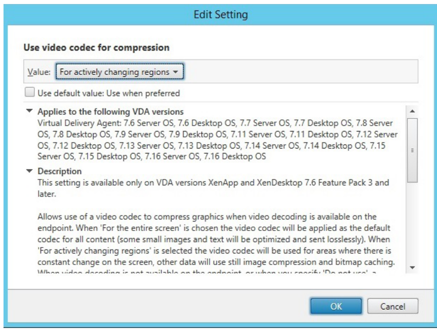
One of the downsides of ACR is latency, because hardware encoding can't be leveraged by the bitmap codec. While you can leverage hardware encoding with the H.264 video codec, it only works with the entire screen and you need to be using Citrix Virtual Desktop 7.17 or later. Prior to Citrix Virtual Desktop 7.17, ACR is possible but you'll see a heavily increased CPU load when the video codecs are initiated.

Because ACR improves video playback, reduces bandwidth, and provides near-pixel perfect static images, it's a pretty good use case for knowledge worker workloads running multimedia content. To enable this policy, follow the steps below

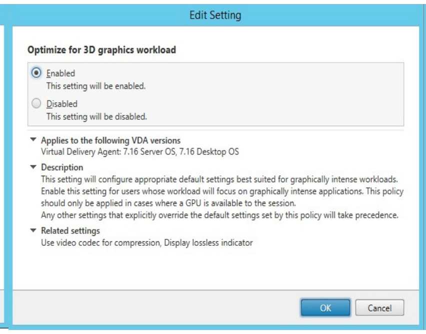
1.1 Use Video Codec for Compression -> For Active Changing Regions

1.2 Use Hardware Encoding -> Enabled

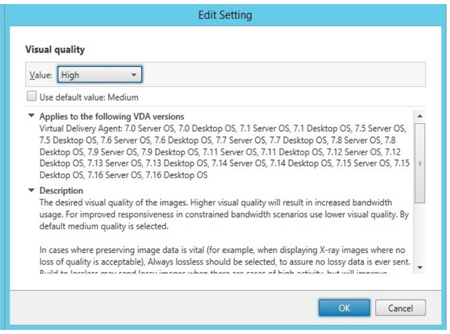
1.3 Visual Quality -> High



1.1



1.2



1.3

CONCLUSION: OPTIMIZED USE CASES

Achieving the best end-user experience in a GPU-accelerated VDI environment depends on user demands for image quality, latency, and bandwidth consumption. These demands are largely determined by the particular task at hand, as well as the machine being used by the end user.

Use the chart below as a guideline to provide your VDI end users with the best experience possible.

Use Case	Protocol	Example
Knowledge Worker VDI Usage	Bitmap (JPG, RLE) *Very bandwidth efficient for static content.*	Office workers using office applications or ERP software without many screen changes. Also applies to use cases where pixel-perfect quality is required at all times.
3D VDI Usage	H.264 YUV420 *Bandwidth efficient for video/moving images*	Engineers/designers who don't need high color accuracy.
3D VDI Usage with high color accuracy requirements	H.264 YUV444 *Very good visual quality, but with increased bandwidth*	An engineer designing a part for a new car model. Without this codec he would experience some "bleeding" of luminance and color.
3D VDI Usage in low bandwidth scenarios	H.265 YUV420 *Better compression at the same visual quality, or the same quality at lower bandwidth (compared to H.264)*	Users accessing VDI over WAN, with a significant amount of video. Video is bandwidth intensive and the biggest benefit of H.265 is compression.
Knowledge Worker VDI usage with multimedia content	Adaptive Display/Selective H.264/H.265 *Very good image quality for static content and low bandwidth for moving images/video*	Knowledge workers using video conferencing applications like Skype for Business, streaming townhall meetings or YouTube videos, or engaged in multimedia-based training.
3D VDI usage with high color accuracy requirements and low bandwidth	H.264 BTL or H.265 BTL *Pixel-perfect quality and lower bandwidth consumption*	Could apply to any use case, as long as the user can accept the sharpening effect.