FrameView

Integrated Frame Benchmarking & Power Tool

USER GUIDE

INTRODUCTION

NVIDIA FrameView™ is a software tool designed to capture and measure performance, latency, and power utilization of PC-based graphics and CPU hardware. It's one of the first solutions to allow easy capture of PC Latency, also known as responsiveness or input lag, the time from when the mouse click is processed by the OS to when the completed frame is sent to the display. It's important to measure latency, since elevated levels of input lag will negatively affect a players performance in any game type, and will be felt by both casual and competitive players alike.

> Learn more about latency by reading the **GeForce Performance Toolkit** guide on the press site.

FrameView is a trusted and accurate performance measuring tool and works with a wide range of graphics cards, all major graphics APIs, and UWP (Universal Windows Platform) apps. It reports detailed metrics in real-time, many of them per frame, and compiles the results in log files for convenient analysis.

FrameView is the best tool for measuring the performance of a gaming system with minimal overhead, including Average FPS, 1% Low FPS for detecting stutters, PC Latency for checking for responsiveness, GPU/CPU clock speeds, utilization, temperature, NVIDIA total board and GPU power, perf-per-watt, and even laptop battery life among other metrics. FrameView has been optimized particularly for detailed frame time, present, and display scheduling metrics for measuring stutter. And it also includes an overlay that shows performance metrics in real-time.

Real-time power measurements for both total board power and GPU chip-only power are captured through application programming interfaces (APIs), which are publicly-available software that communicate with the hardware and return data.

Whether you want to examine each metric per frame for creating detailed reports or simply check the extensive summary file, you will find everything you need to accurately report your system's performance with Frameview.

API SUPPORT
DirectX APIs (versions 9-12), OpenGL, Vulkan

SINGLE-GPU CONFIGURATIONS
NVIDIA® GeForce®, AMD, Intel

MULTI-GPU CONFIGURATIONS
NVIDIA SLI, AMD Crossfire, MSHybrid- and Optimus-based platforms

DISPLAY SUPPORT
G-SYNC, Non-G-SYNC, Adaptive SYNC (including FreeSync)

SCREEN MODES
Full Screen, Windowed, UWP apps

OS SUPPORT
Windows 10 and above

OVERLAY
Average FPS, 1% Low FPS, PC Latency (PCL), GPU & CPU frequency/utilization/temps, display mode tags, DPI aware overlay, perf-per-watt for NVIDIA (NVAPI) and AMD (with PCAT)

NOTE: *DX9/DX10 games do not have overlay support although data capture is supported and properly logged.*

LOGGING
Average FPS, 1% Low FPS, PC Latency (PCL), 0.1% Low FPS, 1/5/10% FPS (Percentile), Benchmark Time, Min/Max PCL, GPU/CPU, Resolution, API, Application Tested, GPU & CPU frequency/power/utilization/temp, GPU memory frequency, perf-per-watt for NVIDIA (NVAPI) and AMD (with PCAT), total board power for NVIDIA (NVAPI) and AMD (with PCAT), GPU power for NVIDIA, AMD Pwr (API), laptop battery drain rate/charge metrics, HW/SW/driver info, and more.

What's New in FrameView 1.4

For the first time ever, gamers can now measure PC Latency in games that support "PC Latency Stats", including all DLSS 3 titles without needing any additional hardware. With FrameView, just press the benchmarking hotkey and go. FrameView makes latency accessible to everyone. Whether you're testing the built-in benchmark or real gameplay, with a press of a button you'll get both average frame rate and PC Latency, among many other metrics. Frameview works on different GPU vendors, making performance comparisons easy to capture.

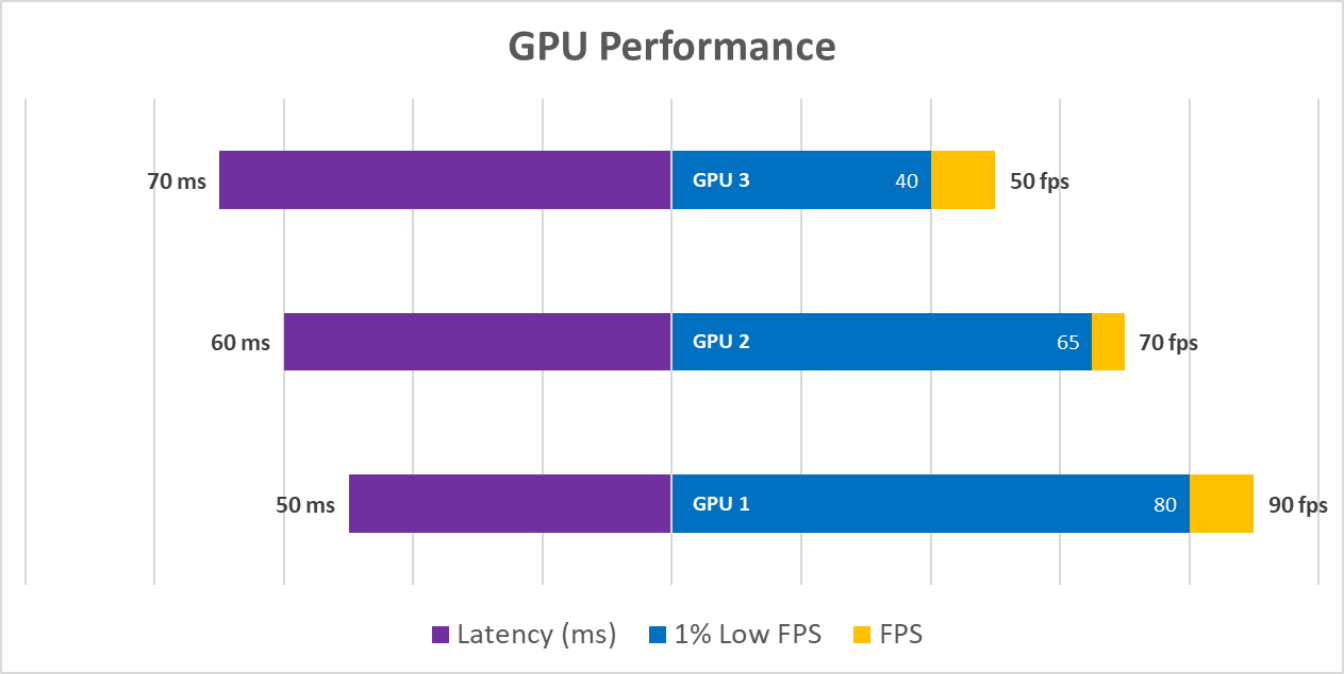
Learn how to capture FPS and latency in the latest games that support PCL Stats by checking out the **GeForce Performance Toolkit** guide on the press site. You will also learn how to enable NVIDIA Reflex Low-Latency Mode to reduce latency and how to benchmark specific games. If you're testing *Fortnite*, please remember to enable Latency Markers to get PC Latency.

The **GeForce Performance Toolkit** guide focuses on testing the three pillars of a GPU's performance: **Smoothness, Responsiveness, and Image Quality.**

FrameView is utilized to measure both **smoothness** and **responsiveness**, while *Image Quality Analysis Tool (ICAT)* is for comparing the differences in **image quality** between AI rendering with DLSS compared to other various spatial upscaling techniques. DLSS has 2nd generation AI, which is better at reconstructing image quality than even temporal upscalers, and as a result you will find **DLSS delivers better detail and more stable images in motion** (how games are actually played).

ICAT can be downloaded here.

- **Easy Latency Benchmarking with PC Latency**
  - No input required
  - No special hardware
  - No specific mouse or monitor
  - Focuses on the PC itself. Isolates PC latency away from mouse and display latency to center on a PC's raw performance
- **Measure latency at the same time as FPS. Just press the benchmarking hotkey to get immediate results. Testing latency has never been easier!**
- **All metrics compatible with DLSS 3 Frame Generation**
- **Cross-platform GPU reviews. Easily compare different GPUs and vendors.**
- **Easily see the latency reduction benefits of NVIDIA DLSS and NVIDIA Reflex**
- **Measure latency per frame**
- **1% Low FPS stutter metric now available**
- **Instant benchmarking results in the overlay - see your results as you go!**



▲ **With FrameView, you'll be able to easily measure latency along with FPS at the same time. In the chart above, latency performance is represented in purple on the left *(lower is better)* and FPS performance is represented on the right *(higher is better)*. Refer to the GeForce Performance Toolkit guide for more information. Click here to learn how to make this chart.**

PCL is N/A — How to resolve?

Make sure you are testing a game that supports PCL Stats. The FrameView overlay will show PCL (PC Latency) as "**N/A**" in most Reflex games at the main menu, but rest assured the FrameView PCL overlay metric will update accordingly when you begin actually playing the game or start the benchmark.

IMPORTANT: If you're testing latency in games that support PCL Stats and keep seeing "N/A" **during gameplay** (not the menus), make sure that you enabled **Latency Markers** by following these steps.

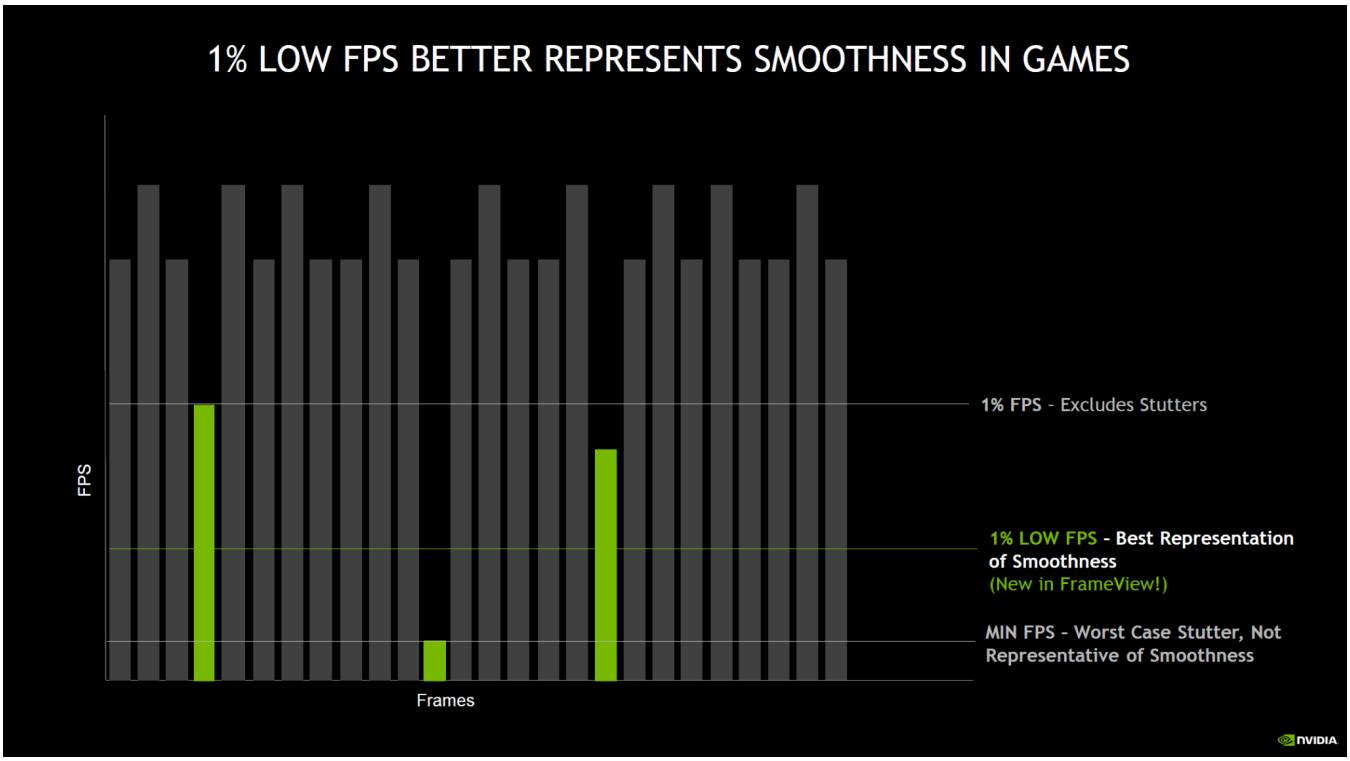
Latency Markers enable developers to show both game and latency metrics in their games.

Games that Support PCL Stats for Measuring Latency

FrameView can measure PC Latency and FPS at the same time in all games that support PCL Stats, which includes games that support DLSS 3.

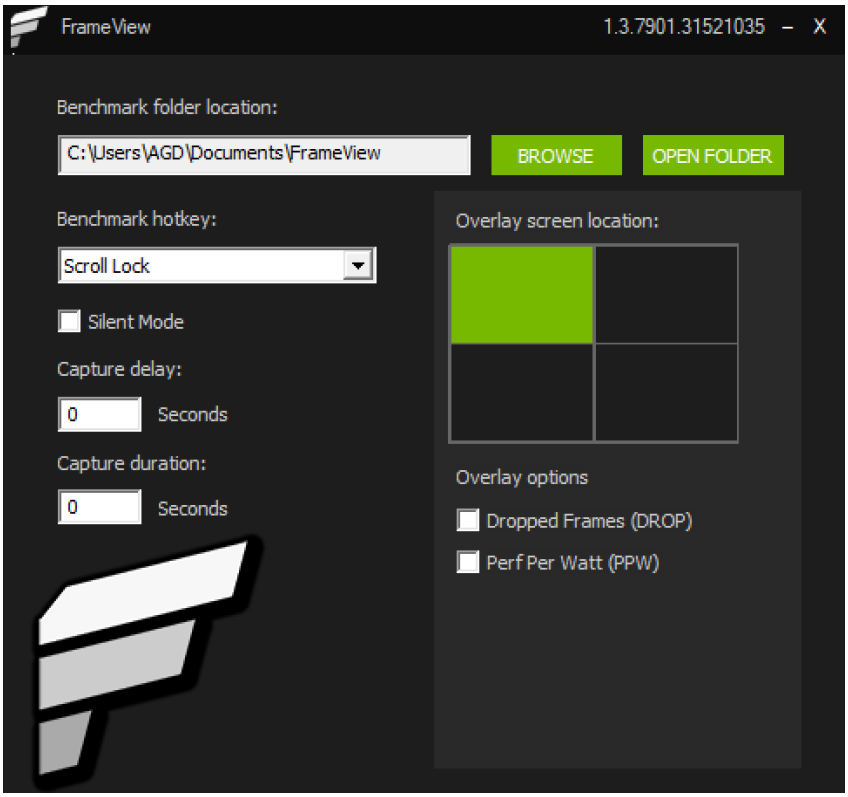
A comprehensive list of titles that support PCL Stats can be found here, under the PC Latency Stats/Reflex Stats column:
https://www.nvidia.com/en-us/geforce/technologies/reflex/supported-products/

Evaluating Smoothness with 1% Low FPS



▲ **1% Low FPS is new to FrameView, and gives reviewers the best metric for quickly evaluating stutter. In the chart above, we show how the old 1% FPS metric would miss capturing the worst stutter experienced in a game. 1% Low FPS takes the slowest 1% frames and averages them, reflecting all stutter experienced in a game.**

FRAMEVIEW INTERFACE & OVERLAY



Benchmark folder location

This is where the benchmark logs will be saved. Use the Browse button to choose a location and the Open Folder button to access saved results in Windows File Explorer.

Benchmark hotkey

This is the button assigned to start and stop the benchmarking process. FrameView supports **Scroll Lock** and **F10** as the benchmarking hotkeys.

Silent Mode

Checking this box will mute the chime when starting and ending the benchmark.

Capture delay

This will delay the capture of a game by the seconds specified in the window. The default is 0 seconds.

Capture duration

This will set a capture time limit for the benchmark. The default is 0 seconds, which means the benchmark capture logging must be manually started and stopped with the hotkey. When the

time limit is set to a number greater than 0, the benchmark logging must still be manually started, but it will be automatically stopped after the specified capture duration.

Overlay screen location

Average FPS, 1% Low FPS, and PC Latency, among other metrics will be displayed by default in the upper-left corner of your monitor when running a game. To change the overlay location, click a different quadrant in the FrameView interface.

> The overlay is automatically disabled during benchmarking to ensure more accurate results. The overlay will return once the benchmark hotkey is pressed a second time.

Overlay Options

- **Perf Per Watt (PPW)**

When enabled, the FrameView overlay will show performance-per-watt (PPW) data for NVIDIA GPUs. This value is reported in the FrameView Log under the ***Perf/W Total(F/J) (API)*** and ***Perf/W GPUOnly(F/J) (API)*** headers for NVIDIA GPUs. To get PPW on AMD GPUs, PCAT or other interposer setup must be utilized. AMD API power, shown as ***AMDPwr(W) (API)*** appears to report a value in-between chip-only and full board power. PCAT is necessary to measure the correct AMD power. Refer to the power measurement section for more information.

- **Dropped Frames (DROP)**

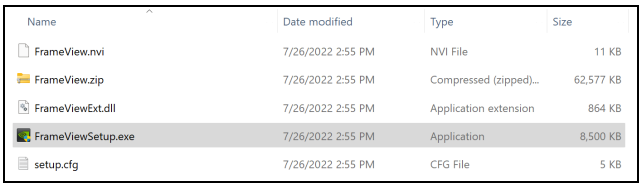
When enabled, the FrameView overlay will show whether the presented frame was dropped (1) or displayed (0). This value is reported in the FrameView Log under the ***Dropped*** header.

Installing FrameView

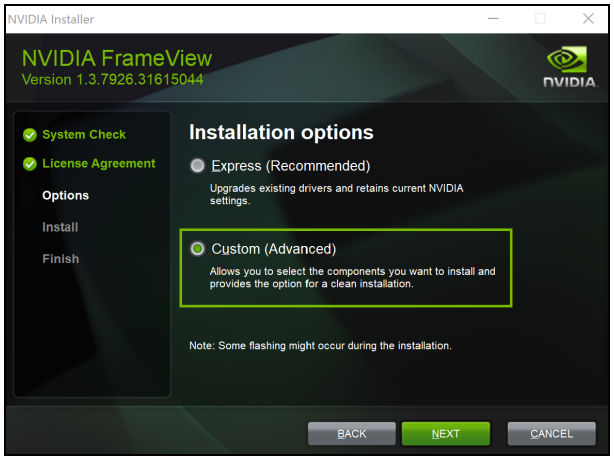
1. Download FrameView 1.4 from the NVIDIA website or NVIDIA press site.

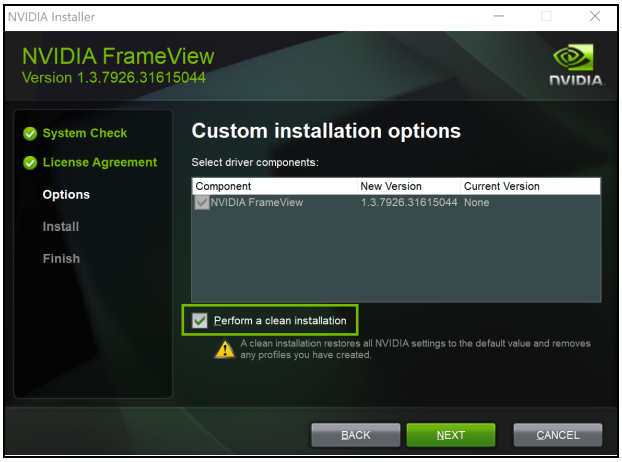
> **IMPORTANT:** Please select **Custom (Advanced)** and check the box to **Perform a clean installation** of FrameView. This will prevent certain issues from occurring.

2. Run **FrameViewSetup.exe** to initiate the installation process.

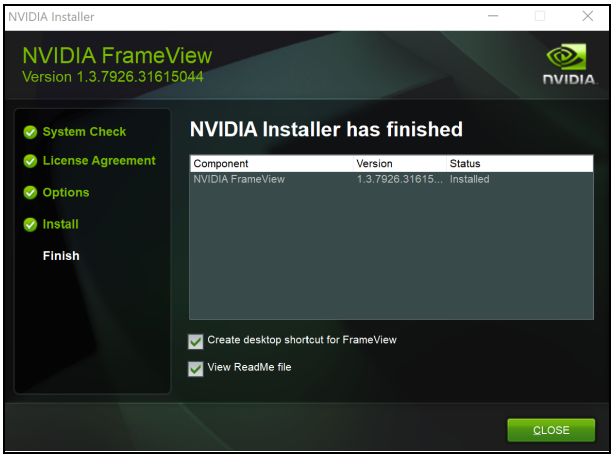


3. Perform a **clean installation** of FrameView.
    a. Click Custom (Advanced)
    b. Check the box to perform a clean installation.



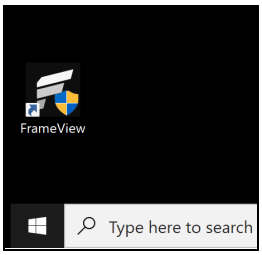


4. Installation complete.

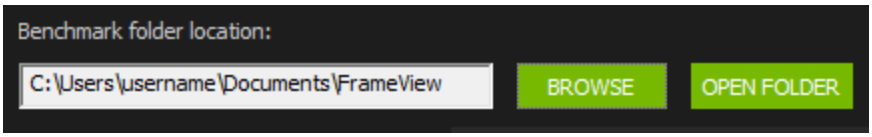


Running FrameView

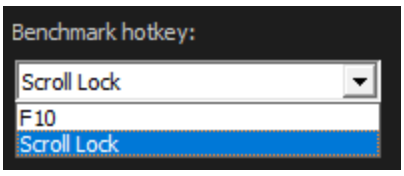
1. **Launch FrameView** using the desktop shortcut.



2. Click **Browse** to determine the **Benchmark folder location** where results will be stored.



3. At this time, FrameView supports **Scroll Lock** and **F10** as the benchmarking hotkeys.



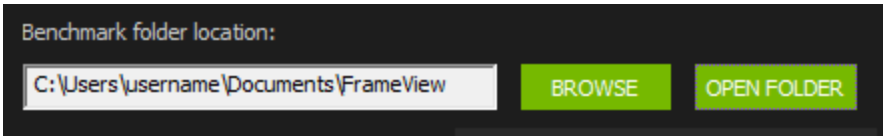
4. Launch a game that supports PCL Stats such as ***Fortnite*** or any title that supports **DLSS 3** to **measure latency alongside FPS.** Or launch any other game to measure performance.

5. The **FrameView overlay** should show up in the designated area chosen in the FrameView interface. Note that DX9/DX10 games do not have overlay support although data capture is supported and properly logged.

6. **Press** the benchmarking hotkey (default is **Scroll Lock**) to begin benchmarking.
   The overlay will disappear during data collection to reduce overhead in the captured data.

7. **Press** the benchmarking hotkey again to stop data collection.
   The overlay will reappear with a summary of your benchmark in the designated area.

8. **Exit the game** and return to FrameView.
   Click the **Open Folder** button to view benchmark results.



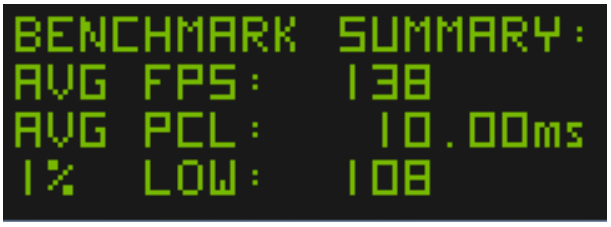
9. FrameView performance results will be saved as **.CSV files** with an application and timestamp name along with a summary file of all runs. Consider renaming the files or creating a directory to reflect the GPU, game, and settings tested.

Name	Date modified	Type	Size
FrameView_SOTTR.exe_2022_07_20T124206_Log.csv	7/20/2022 12:45 PM	Microsoft Excel Com...	4,278 KB
FrameView_SOTTR.exe_2022_07_20T124527_Log.csv	7/20/2022 12:48 PM	Microsoft Excel Com...	4,257 KB
FrameView_SOTTR.exe_2022_07_20T124849_Log.csv	7/20/2022 12:51 PM	Microsoft Excel Com...	4,253 KB
FrameView_SOTTR.exe_2022_07_20T125703_Log.csv	7/20/2022 12:59 PM	Microsoft Excel Com...	4,245 KB
FrameView_Summary.csv	7/20/2022 1:00 PM	Microsoft Excel Com...	3 KB

FrameView Overlay

The overlay will appear in games with proper API support when FrameView is running in the background. If the overlay does not appear, make sure to check if FrameView is running. You can adjust the overlay screen location in the FrameView settings to move the overlay to a different corner of the screen.

When benchmarking is enabled through the hotkey, the overlay will disappear. Removing the overlay reduces overhead to ensure a more accurate capture of the game data. A summary of the benchmark will appear when the hotkey is pressed again, and capture is stopped or capture duration expires.



After a short duration, the live updating overlay will reappear.

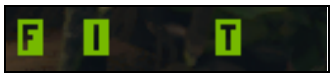
> Overlay information will not be shown in DX9/10 games. However, data capture is supported and the information will be properly logged.

Overlay Metrics

LABEL	METRICS	DESCRIPTION
FPS	**Avg FPS**	**One of 3 key metrics for evaluating a game.** Average frames per second, using rendered FPS scheduling metrics.
1%L	**1% Low FPS**	**One of 3 key metrics for evaluating a game.** Takes the slowest 1% frames and averages them. Reports stutter—the closer 1% Low is to Avg FPS, the more consistent the experience will be.
PCL	**AvgPCLatency (ms)**	**One of 3 key metrics for evaluating a game.** Average time between PC receiving input and frame being sent to the display, in milliseconds. Supported in titles with PCL Stats compatibility, including all DLSS 3 titles ("NA" in unsupported titles or when in in-game menus). When testing Reflex games, run the benchmark or play the game for the PCL metric to update in the overlay. PC Latency focuses on the performance of a desktop PC/GPU and does not include the mouse or monitor display latency. PC Latency is automatically reported by FrameView when measuring FPS.
GPUN GPUA GPUI	**GPUN/A/I**	The letter at the end of “GPU” indicates the vendor, signifying NVIDIA, AMD, or Intel GPUs. Both integrated and dedicated GPUs, when detected, will appear in the overlay.
	GPU Frequency	GPU Frequency (MHz)
	GPU % Utilization	GPU utilization (percentage)
	GPU Temperature	GPU Temperature (Celcius)
CPU	**CPU Frequency**	CPU Frequency (MHz)
	CPU Utilization	CPU utilization (percentage)
	CPU Temperature	CPU Temperature (Celcius)
DROP	**Dropped Frames**	Whether the present was dropped (1) or displayed (0). *Must be enabled in the FrameView interface settings by checking the box*.
PPW	**Perf Per Watt**	Performance-per-watt measured as frames/joule (F/J) and described in more detail below. Considering MsBetweenPresents for performance and NV-Total-USBCPwr(W) (API). *Must be enabled in the FrameView interface settings by checking the box.* **NOTE:** NVAPI power is used for NVIDIA. PPW is only shown for AMD boards if PCAT is used. PCAT Power Total(W) is used if connected and running in the background.

Overlay Mode Tags

The overlay also includes three mode tags. These letters are used to provide information about game settings that can impact overlay data reporting and data captures.




F (Full Screen)

Running the game at full screen will ensure that accurate performance results are measured at the resolution specified in the game.

I (Independent flip)

iFlip (also called Independent flip), is the mode where the app is simulating as if it was running in Full Screen Exclusive mode.

T (Tearing)

When Vertical Sync is disabled, the full performance of the game can be measured (due to higher frame rates beyond the refresh rate of the monitor). However, a major artifact of disabling Vertical Sync is tearing. This is the optimal method for testing game performance.

W (Windowed mode)

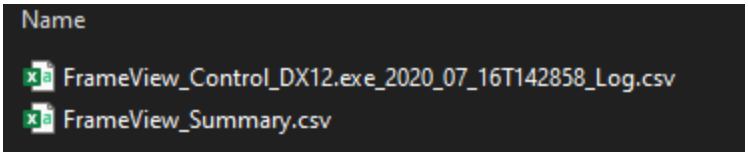
If the game runs in windowed mode, pressing the Alt+Enter keys on your keyboard while the game is running can sometimes force the game into fullscreen mode. Check the game settings if that does not work.

V (Vsync ON)

Vertical Sync is enabled, which forces the frame rate of the game to synchronize with the refresh rate of the monitor. However, this limits the frames that can be displayed, and will not show the full performance potential of the hardware being measured.

FRAMEVIEW FILES

Two files are created once the benchmarking is completed using the hotkeys. Both files are comma-separated value (CSV) files using UTF-8 encoding that can be imported into Microsoft Excel, OpenOffice, or Google Sheets. These files are saved in the benchmark folder location that is specified in the FrameView user interface. They are named using the application process name and include date and time stamps.



FrameView Summary File

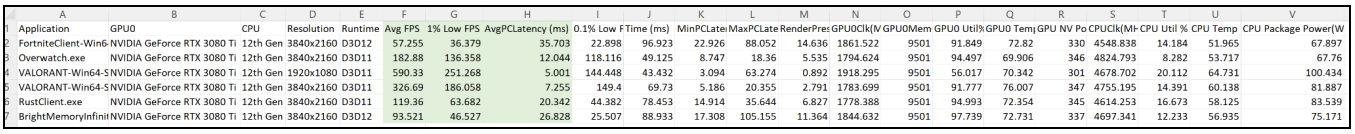
Frameview generates two output files:

- FrameView_Summary.csv - contains aggregate statistics
- FrameView_<Application exe name>_YYYY_MM_DDTHHMMSS_Log.csv - contains per frame data

FrameView_Summary.csv contains the high-level data from all captured runs using the FrameView hotkey. When opened, the file will look like this:

	A	B	C	D	E	F	G	H	I	J	K
1	TimeStamp	Application	Log Name	GPU0	GPU1	CPU	Resolution	Runtime	Avg FPS	1% Low FPS	AvgPCLatency (ms)
2	2022-05-12T153136	FortniteClient-Win	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	3840x2160	D3D12	57.255	36.379	35.703
3	2022-05-12T160147	Overwatch.exe	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	3840x2160	D3D11	182.878	136.358	12.044
4	2022-05-12T174610	VALORANT-Win64-	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	1920x1080	D3D11	590.327	251.268	5.001
5	2022-05-12T175159	VALORANT-Win64-	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	3840x2160	D3D11	326.686	186.058	7.255
6	2022-05-12T175938	RustClient.exe	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	3840x2160	D3D11	119.359	63.682	20.342
7	2022-05-12T181141	BrightMemoryInfir	FrameView	NVIDIA GeForce RTX 3080 Ti	NA	12th Gen Ir	3840x2160	D3D12	93.521	46.527	26.828

You can duplicate the summary file and make your own custom summary table:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Application	GPU0	CPU	Resolution	Runtime	Avg FPS	1% Low FPS	AvgPCLatency (ms)	0.1% Low F	Time (ms)	MinPCLate	MaxPCLate	RenderPres	GPU0Clk(M	GPU0Mem	GPU0 Util%	GPU0 Temp	GPU NV Po	CPUClk(MH	CPU Util %	CPU Temp	CPU Package Power(W
2	FortniteClient-Win6	NVIDIA GeForce RTX 3080 Ti	12th Gen	3840x2160	D3D12	57.255	36.379	35.703	22.898	96.923	22.926	88.052	14.636	1861.522	9501	91.849	72.82	330	4548.838	14.184	51.965	67.897
3	Overwatch.exe	NVIDIA GeForce RTX 3080 Ti	12th Gen	3840x2160	D3D11	182.88	136.358	12.044	118.116	49.125	8.747	18.36	5.535	1794.624	9501	94.497	69.906	346	4824.793	8.282	53.717	67.76
4	VALORANT-Win64-S	NVIDIA GeForce RTX 3080 Ti	12th Gen	1920x1080	D3D11	590.33	251.268	5.001	144.448	43.432	3.094	63.274	0.892	1918.295	9501	56.017	70.342	301	4678.702	20.112	64.731	100.434
5	VALORANT-Win64-S	NVIDIA GeForce RTX 3080 Ti	12th Gen	3840x2160	D3D11	326.69	186.058	7.255	149.4	69.73	5.186	20.355	2.791	1783.699	9501	91.777	76.007	347	4755.195	14.391	60.138	81.887
6	RustClient.exe	NVIDIA GeForce RTX 3080 Ti	12th Gen	3840x2160	D3D11	119.36	63.682	20.342	44.382	78.453	14.914	35.644	6.827	1778.388	9501	94.993	72.354	345	4614.253	16.673	58.125	83.539
7	BrightMemoryInfinit	NVIDIA GeForce RTX 3080 Ti	12th Gen	3840x2160	D3D12	93.521	46.527	26.828	25.507	88.933	17.308	105.155	11.364	1844.632	9501	97.739	72.731	337	4697.341	12.233	56.935	75.171

This table explains each header and the data contained in it:

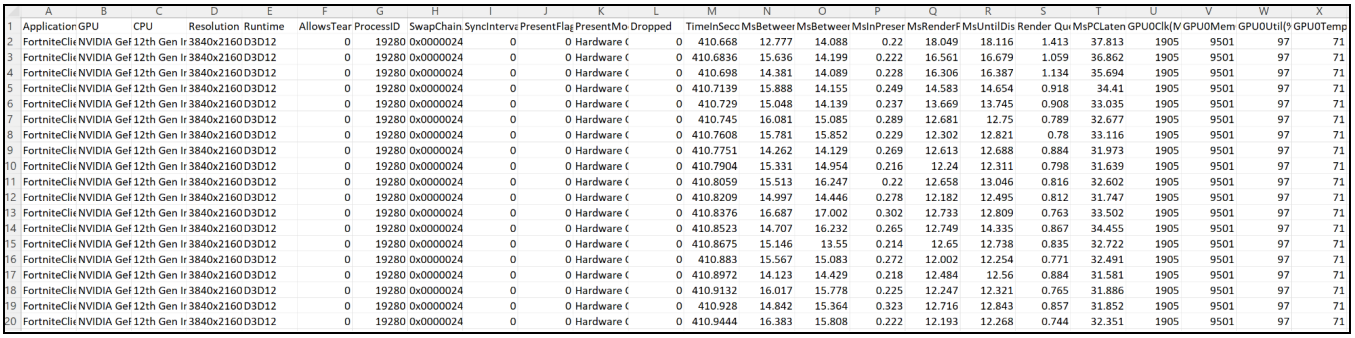
SUMMARY HEADERS	DESCRIPTION
TimeStamp	End of Benchmark, represented as YYYY-MM-DD and HHMMSS PC local time
Application	Executable name that was captured
Log Name	Name of the corresponding log contributing to summary file
GPU0	If more than one GPU is available (including iGPU), this is the first GPU
GPU1	If more than one GPU is available (including iGPU), this is the second GPU
CPU	Retail CPU make/model name
Resolution	Output resolution of the game/application captured
Runtime	The Graphics API version/type used by the game
Avg FPS	**One of 3 key metrics for evaluating a game.** Average frames per second, using rendered FPS scheduling metrics.
1% Low FPS	**One of 3 key metrics for evaluating a game.** Takes the slowest 1% frames and averages them. Reports stutter—the closer 1% Low is to Avg FPS, the more consistent the experience will be.
AvgPCLatency (ms)	**One of 3 key metrics for evaluating a game.** Average time between PC receiving input and frame being sent to the display, in milliseconds. Supported in titles with PCL Stats compatibility, including all DLSS 3 titles ("NA" in unsupported titles). When testing Reflex games, run the benchmark or play the game for the PCL metric to update in the overlay. PC Latency focuses on the performance of a desktop PC/GPU and does not include the mouse or monitor display latency. PC Latency is automatically reported by FrameView when measuring FPS.
Min FPS	Minimum framerate
Max FPS	Maximum framerate
0.1% Low FPS	Takes the slowest 0.1% frames and averages them. Reports stutter—the closer 0.1% Low is to Avg FPS, the more consistent the experience will be.
0.1% FPS	The FPS separating the slowest 0.1% frame rates from the fastest 99.9% frame rates (percentile measurement)
1% FPS	The FPS separating the slowest 1% frame rates from the fastest 99% frame rates (percentile measurement)
5% FPS	The FPS separating the slowest 5% frame rates from the fastest 95% frame rates (percentile measurement)

10% FPS	The FPS separating the slowest 10% frame rates from the fastest 90% frame rates (percentile measurement)
Time (ms)	The amount of time that comprises the capture.
MinPCLatency (ms)	Minimum time between PC receiving an input and frame being sent to the display, in milliseconds. Supported in titles with PCL Stats compatibility, including all DLSS 3 titles ("NA" in unsupported titles).
MaxPCLatency (ms)	Maximum time between PC receiving an input and frame being sent to the display, in milliseconds. Supported in titles with PCL Stats compatibility, including all DLSS 3 titles ("NA" in unsupported titles or when in in-game menus).
Render Present Latency	The time when the present call entered the queue to the time the present call was executed on the GPU.
GPU# Clk (MHz)	GPU frequency (MHz) [# denotes GPU number]
GPU# MemClk (MHz)	GPU frame buffer frequency (MHz) [# denotes GPU number]
GPU# Util %	GPU utilization (percentage) [# denotes GPU number]
GPU# Temp (C)	GPU temperature (Celcius) [# denotes GPU number]
Perf/Watt (F/J) (PCAT)	Performance-per-watt considering MsBetweenPresents for performance and board power as measured by PCAT.
PCAT Power (Watts)	If a PCAT device is available, the total power consumption of the GPU as reported by PCAT.
GPU NV Power (Watts) (API)	NVIDIA GPU power as reported by NVAPI (same as PCAT power).
CPUClk (MHz)	CPU frequency (MHz)
CPU Util %	CPU utilization (percentage)
CPU Temp (C)	CPU temperature (Celcius)
CPU Package Power (Watts)	Total CPU power.
Current Battery Capacity (Wh)	Only for Notebooks - current battery capacity (watt hours)
Total Battery Capacity (Wh)	Only for Notebooks - total battery capacity (watt hours)
Battery Percentage	Only for Notebooks - battery percentage calculated from current battery capacity and total battery capacity
Battery Drain Rate (W)	Only for Notebooks - rate of battery consumption (watts). Will be negative when discharging, positive when charging, and zero at full charge
Battery Charge Rate (W)	Only for Notebooks - rate of battery charging (watts). Will be positive when charging and “NA” when discharging or at full charge
OS	Operating System Info - taken from WMI (Windows Management Instrumentation)
GPU Base Driver	Base Driver Version - taken from WMI (Windows Management Instrumentation)
GPU Driver Package	Driver Package - taken from WMI (Windows Management Instrumentation)

System RAM	Size and type of System RAM - taken from WMI (Windows Management Instrumentation)
Motherboard	Motherboard make/model - taken from WMI (Windows Management Instrumentation)

FrameView Log File

FrameView_<Application exe name>_YYYY_MM_DDTHHMMSS_Log.csv contains the per frame metrics, including some metrics not found in the FrameView_Summary.csv. When opened, the file will look like this:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Application	GPU	CPU	Resolution	Runtime	AllowsTear	ProcessID	SwapChain	SyncInterva	PresentFlag	PresentMo	Dropped	TimeInSecc	MsBetweer	MsBetweer	MsInPreser	MsRenderF	MsUntilDis	Render Qu	MsPCLaten	GPU0Clk(M	GPU0Mem	GPU0Util(9	GPU0Temp
2	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.668	12.777	14.088	0.22	18.049	18.116	1.413	37.813	1905	9501	97	71
3	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.6836	15.636	14.199	0.222	16.561	16.679	1.059	36.862	1905	9501	97	71
4	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.698	14.381	14.089	0.228	16.306	16.387	1.134	35.694	1905	9501	97	71
5	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.7139	15.888	14.155	0.249	14.583	14.654	0.918	34.41	1905	9501	97	71
6	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.729	15.048	14.139	0.237	13.669	13.745	0.908	33.035	1905	9501	97	71
7	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.745	16.081	15.085	0.289	12.681	12.75	0.789	32.677	1905	9501	97	71
8	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.7608	15.781	15.852	0.229	12.302	12.821	0.78	33.116	1905	9501	97	71
9	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.7751	14.262	14.129	0.269	12.613	12.688	0.884	31.973	1905	9501	97	71
10	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.7904	15.331	14.954	0.216	12.24	12.311	0.798	31.639	1905	9501	97	71
11	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8059	15.513	16.247	0.22	12.658	13.046	0.816	32.602	1905	9501	97	71
12	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8209	14.997	14.446	0.278	12.182	12.495	0.812	31.747	1905	9501	97	71
13	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8376	16.687	17.002	0.302	12.733	12.809	0.763	33.502	1905	9501	97	71
14	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8523	14.707	16.232	0.265	12.749	14.335	0.867	34.455	1905	9501	97	71
15	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8675	15.146	13.55	0.214	12.65	12.738	0.835	32.722	1905	9501	97	71
16	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.883	15.567	15.083	0.272	12.002	12.254	0.771	32.491	1905	9501	97	71
17	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.8972	14.123	14.429	0.218	12.484	12.56	0.884	31.581	1905	9501	97	71
18	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.9132	16.017	15.778	0.225	12.247	12.321	0.765	31.886	1905	9501	97	71
19	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.928	14.842	15.364	0.323	12.716	12.843	0.857	31.852	1905	9501	97	71
20	FortniteClie	NVIDIA GeF	12th Gen Ir	3840x2160	D3D12	0	19280	0x0000024	0	0	Hardware (	0	410.9444	16.383	15.808	0.222	12.193	12.268	0.744	32.351	1905	9501	97	71

The log file is useful for graphing metrics over time such as frame times (MsBetweenDisplayChange) or GPU clocks; simply highlight the columns and insert a line chart.

This table explains each log file header, if the data is collected per frame or sampled on a regular interval, and how to interpret the column as a whole:

LOG HEADER	COLLECTION	DESCRIPTION
Application	Per Frame	The name of the process that called Present (if known)
GPU	Per Frame	Retail GPU make/model name
CPU	Per Frame	Retail CPU make/model name
Resolution	Per Frame	Output resolution of the game/application captured
Runtime	Per Frame	The runtime used to present (e.g., D3D9, DXGI, VK, OGL)
AllowsTearing	Per Frame	Whether tearing possible (1) or not (0)
ProcessID	Per Frame	The process ID of the process that called Present
SwapChainAddress	Per Frame	The address of the swap chain that was presented into
SyncInterval	Per Frame	Sync interval used in the Present call
PresentFlags	Per Frame	Flags used in the Present call
PresentMode	Per Frame	Flip model used for this Present

Dropped	Per Frame	Whether the frame was dropped (1) or displayed (0); if dropped, MsUntilDisplayed will be 0
TimeInSeconds	Per Frame	The time of the Present call, measured from when FrameView recording started in seconds
MsBetweenPresents	Per Frame	The time between this Present call and the previous one, in milliseconds
MsBetweenDisplayChange	Per Frame	The time between when the previous frame was displayed and this frame was, in milliseconds
MsInPresentAPI	Per Frame	The time spent inside the Present call, in milliseconds
MsRenderPresentLatency	Per Frame	The time when the present call entered the queue to the time the present call was executed on the GPU.
MsUntilDisplayed	Per Frame	The time between the Present call (TimeInSeconds) and when the frame was displayed, in milliseconds
Render Queue Depth	Per Frame	Maximum pre-rendered frames
MsPCLatency	Per Frame	Time between PC receiving input and frame being sent to the display, in milliseconds. Supported in titles with PCL Stats compatibility, including all DLSS 3 titles ("NA" in unsupported titles or when in in-game menus).
GPU#Clk (MHz)	Sampled	GPU0 frequency (MHz) [# denotes GPU number]
GPU#MemClock (MHz)	Sampled	GPU0 memory clock (MHz) [# denotes GPU number]
GPU#Util (%)	Sampled	GPU0 utilization (percentage) [# denotes GPU number]
GPU#Temp (C)	Sampled	GPU0 temperature (Celcius) [# denotes GPU number]
PCAT Power Total (W)	Sampled	GPU board power as measured by PCAT
Perf/W Total(F/J) (PCAT)	Sampled	Performance-per-watt considering MsBetweenPresents for performance and board power as measured by PCAT
Perf/W Total(F/J) (API)	Sampled	Performance-per-watt considering MsBetweenPresents for performance and board power
Perf/W GPUOnly(F/J) (API)	Sampled	Performance-per-watt considering MsBetweenPresents for performance and GPU/Chip
Perf/W Total-USBC(F/J) (API)	Sampled	Performance-per-watt considering MsBetweenPresents for performance and board power excluding USB-C
GPUOnlyPwr(W) (API)	Sampled	GPU/Chip/ASIC power, post-regulator
NV-Total-USBCPwr(W) (API)	Sampled	Board power excluding USB-C
NV Pwr(W) (API)	Sampled	Board power
AMDPwr(W) (API)	Sampled	Board power **NOTE**: The AMD API used by FrameView appears to report a value in-between chip power and board power for AMD graphics cards. Therefore it's currently not possible to use FrameView to directly compare AMD board power to NVIDIA board power. PCAT is necessary.
CPUClk (MHz)	Sampled	Average CPU frequency (MHz)

CPUUtil (%)	Sampled	CPU utilization (percentage)
CPU Package Temp (C)	Sampled	Overall CPU temperature (Celsius)
CPU Package Power (W)	Sampled	Total CPU power
CPU TDP (W)	Sampled	CPU thermal design power
CPUCoreUtil%[##]	Sampled	CPU utilization (percentage) [## denotes CPU core number]
Current Battery Capacity (Wh)	Sampled	Only for Notebooks - current battery capacity (watt hours)
Total Battery Capacity (Wh)	Sampled	Only for Notebooks - total battery capacity (watt hours)
Battery Percentage	Sampled	Only for Notebooks - battery percentage calculated from current battery capacity and total battery capacity
Battery Drain Rate (W)	Sampled	Only for Notebooks - rate of battery consumption (watts). Will be negative when discharging, positive when charging, and zero at full charge

LAPTOP BATTERY LIFE

It's no surprise that battery life is the most important factor determining the feasibility of gaming and running other power hungry applications on a laptop. Regardless of whether you plan to unplug or not, you will eventually find yourself in a situation without a power outlet, and it's important to know how your device will perform in these situations. FrameView gives you the power to test and compare battery life on various performing laptops to see which is best suited for unplugged gamers.

FrameView reports four battery metrics in the log file and five metrics in summary file. The extra metric (Battery Charge Rate) is present because we split the battery drain rate in log file into two based on positive and negative values when calculating averages for the summary file.

Current Battery Capacity (Wh)	Remaining battery life in watt-hours.
Total Battery Capacity (Wh)	Maximum battery capacity in watt-hours.
Battery Percentage	Remaining battery life as a percentage.
Battery Drain Rate (W)	Current battery drain rate in watts.
Battery Charge Rate (W)	Current battery charge rate in watts.

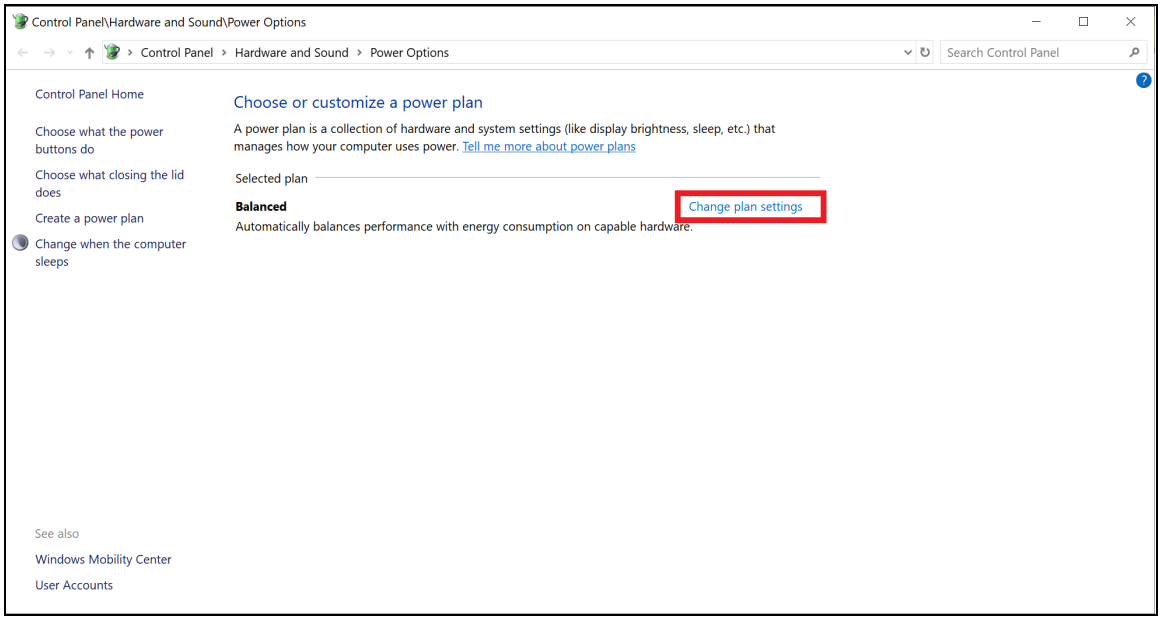
Setting Up

Close all non-essential applications in the background and note the remaining applications in use. It's recommended to choose the default/balanced Windows power plan that ships with the laptop.

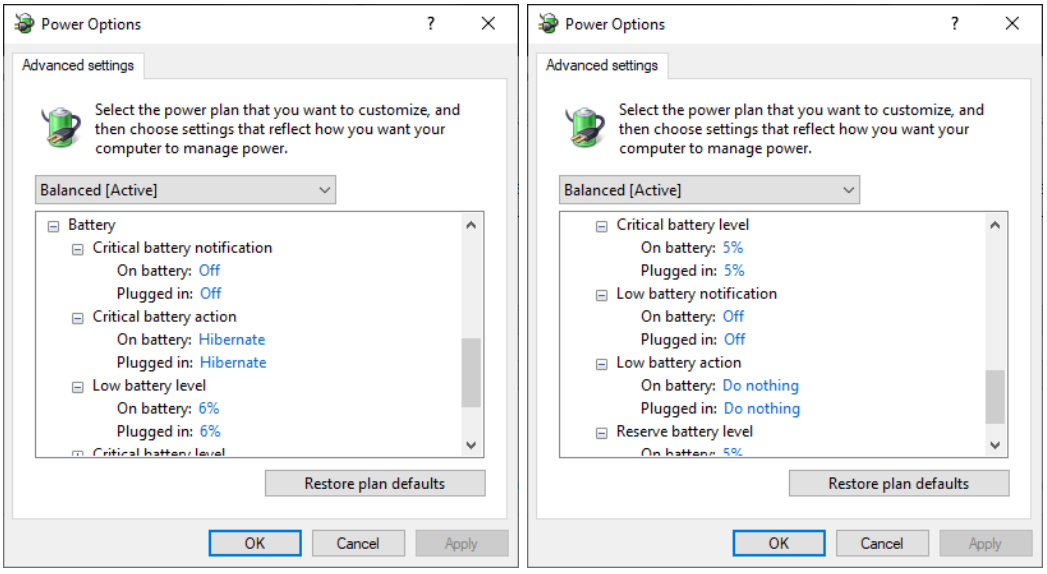
You can verify the current Windows power plan by navigating to Control Panel → Hardware and Sound → Power Options. Click **Change Plan Settings** to modify other settings. See our recommended list of changes below.

Recommended Power Settings for Gaming on Battery

1. **Windows Power Plan** (Control Panel → Hardware and Sound → Power Options)
    a. Windows Power Plan: **Balanced**

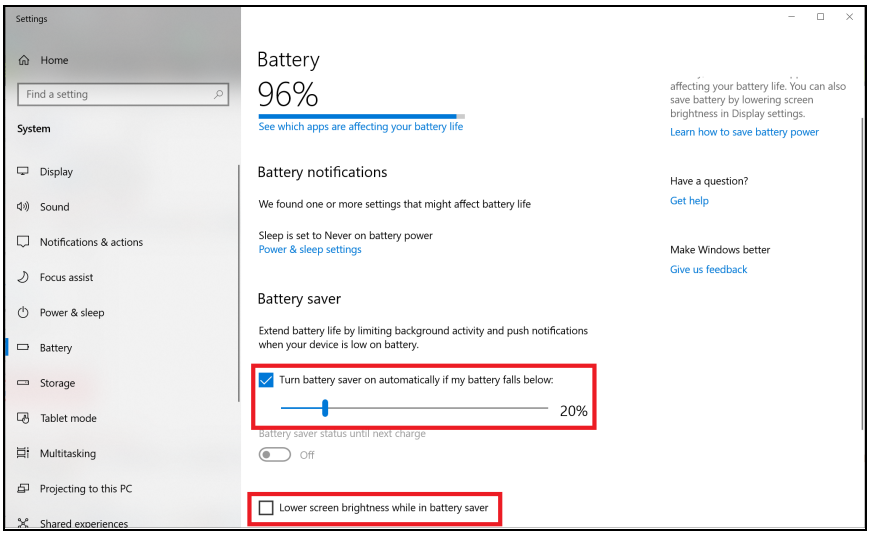


2. **Advanced Power Settings** (Control Panel → Hardware and Sound → Power Options → Change Plan Settings)
    a. Critical Battery Action: **Hibernate**
    b. Low Battery Level: **6%**
    c. Critical Battery Level: **5%**
    d. Low Battery Notification: **Off**
    e. Low Battery Action: **Do Nothing**
    f. Reserve Battery Level: **5%**
    g. Adaptive Brightness: **Off**
    h. Wi-Fi: **On**

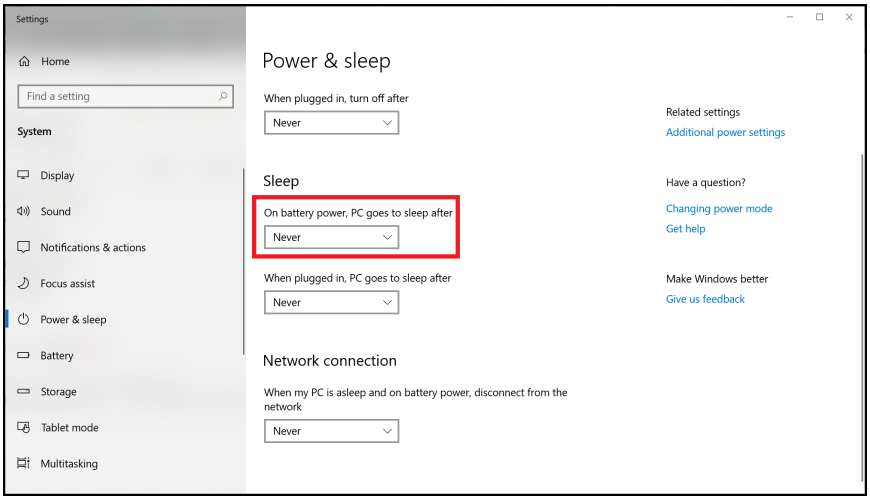


3. **Battery Settings** (Settings → System → Battery)

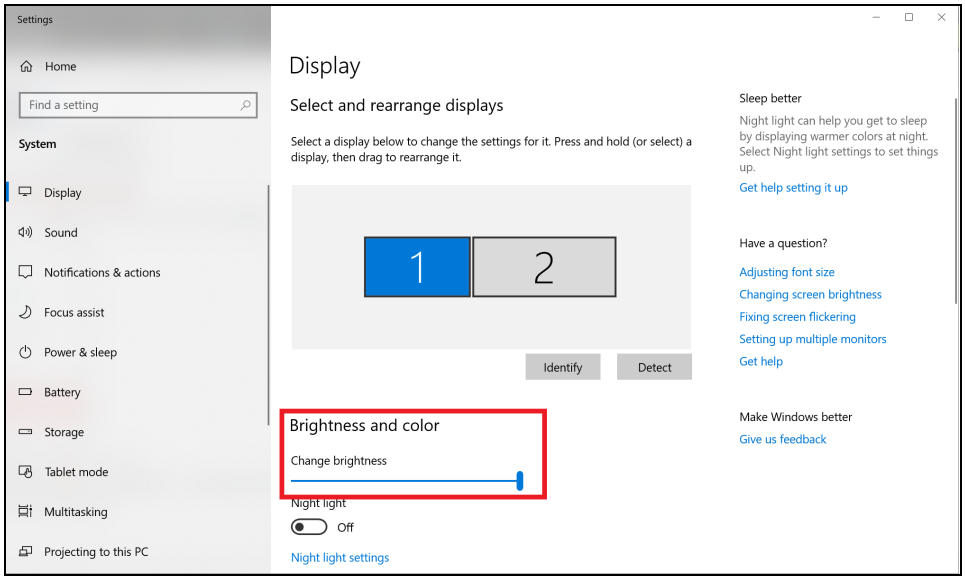
a. Turn battery saver on automatically if my battery falls below: **20%**
b. Lower screen brightness while in battery saver: **Unchecked**



4. **<u>Power & Sleep Settings</u>** (Settings ⟶ System ⟶ Power & Sleep)
    a. On battery power, PC goes to sleep after: **Never**



5. **<u>Display Settings</u>** (Settings ⟶ System ⟶ Display)
    a. Open a Notepad window and make sure it is centered in the middle of the screen. Measuring with a light meter in the center of the screen over the blank notepad page, adjust brightness using the slider getting as close to 150 nits as possible.



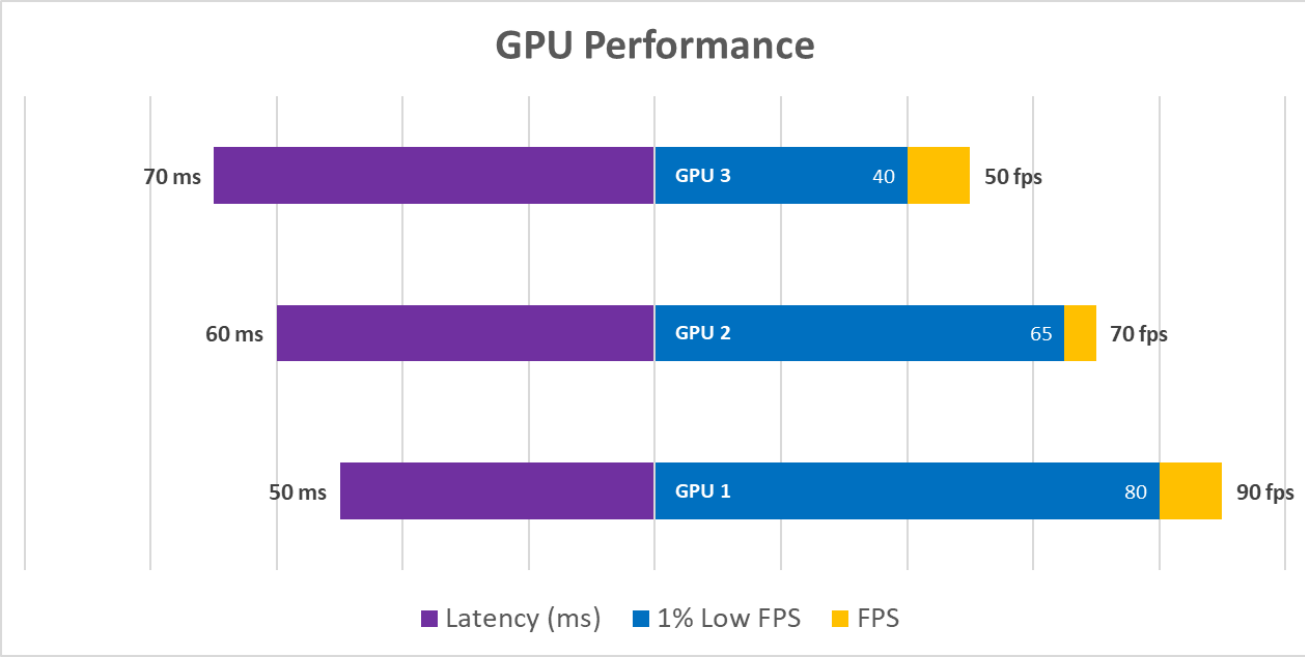
6. Also keep in mind other system components like Bluetooth or other devices installed/attached to the laptop that could be draining battery life.
7. It's recommended to Install GeForce Experience™ for enabling NVIDIA Battery Boost™. Battery Boost works hand in hand to extend battery life whether you are working or playing games, extending battery life by up to 2x.

Measuring Battery Life

- Open FrameView using the desktop shortcut.
- Set the correct benchmarking hotkey depending on your setup. **SCROLL LOCK** or **F10**.
- Decide which game to test and keep note of the app's graphics and video settings.
- Launch the game and find a static scene you wish to test. Unplug the power to the laptop and then press the **benchmarking hotkey** to start collecting system and battery life performance while running a game. Do not interact with the system until the battery runs out. Stop the capture manually by pressing the **benchmarking hotkey** or wait until the system shuts off (log will be saved). If you completely drain the battery, reconnect the power and boot the system.
- Learn how to plot battery life data.

CHARTING THE DATA

Plotting PC Latency alongside Average FPS and 1% Low FPS

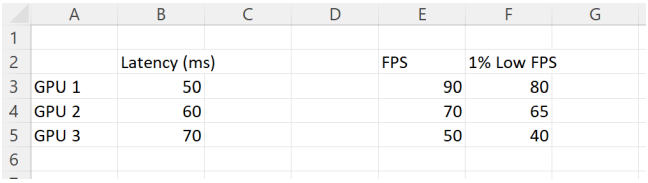


If you would like to give this chart a try, below you will find step-by-step instructions on how to create this horizontal split chart for visualizing **FPS**, **1% Low FPS**, and **Latency** all on a single chart. What makes this chart great is that readers can quickly identify which GPU delivers the best performance in regards to both FPS and Latency, all without visual distraction. Latency performance is represented in purple on the left *(lower is better)* and FPS performance is represented on the right *(higher is better)*. In your testing, you may encounter scenarios where Vendor A has higher FPS performance than Vendor B, but has higher latency than Vendor B. This higher latency, lack of responsiveness, or lag *will* hamper a player's ability to aim and react in games.

Horizontal Split Chart Creation (Step-by-Step)

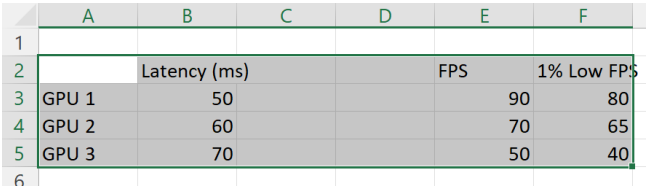
1. Open Microsoft Excel.	

2. Input your results in this format while **leaving 2 blank columns in between PCLatency (ms) and FPS.**



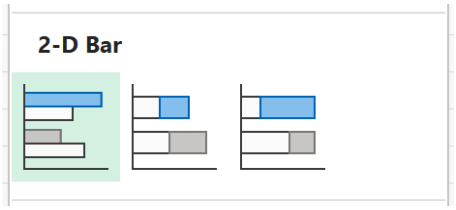
	A	B	C	D	E	F	G
1							
2		Latency (ms)			FPS	1% Low FPS	
3	GPU 1	50			90	80	
4	GPU 2	60			70	65	
5	GPU 3	70			50	40	
6							

3. Highlight the data.

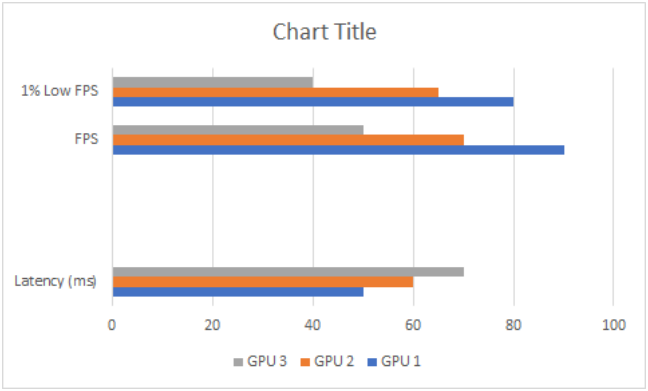


	A	B	C	D	E	F
1						
2		Latency (ms)			FPS	1% Low FPS
3	GPU 1	50			90	80
4	GPU 2	60			70	65
5	GPU 3	70			50	40
6						

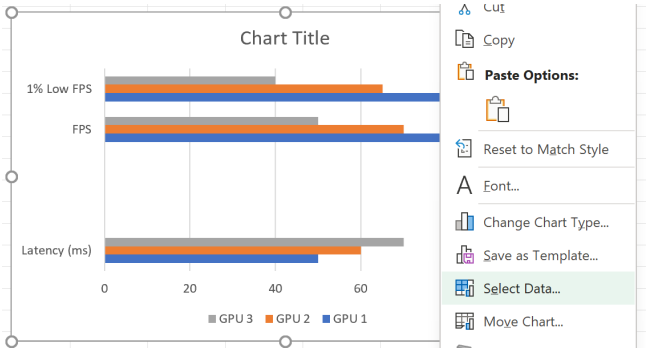
4. Click Insert and choose a **2-D Bar (Clustered Bar)**



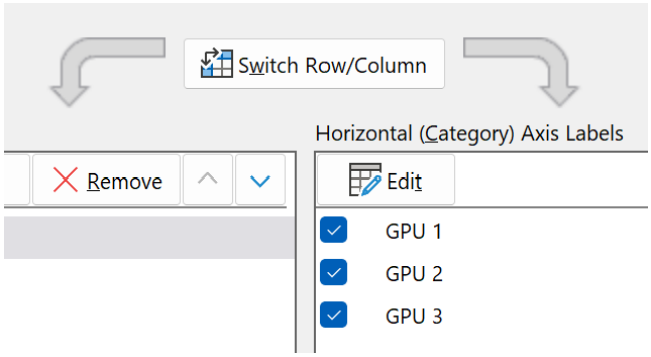
5. Temporary chart design.



6. Right-click the chart and choose **Select Data**.



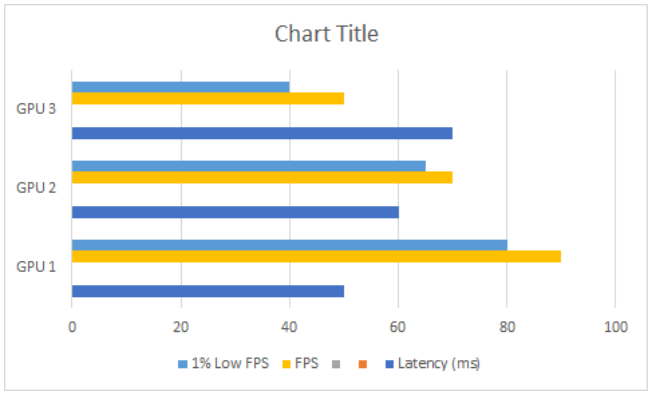
7. Click Switch Row/Column.



8. This is what you'll see after clicking **Switch Row/Column**.

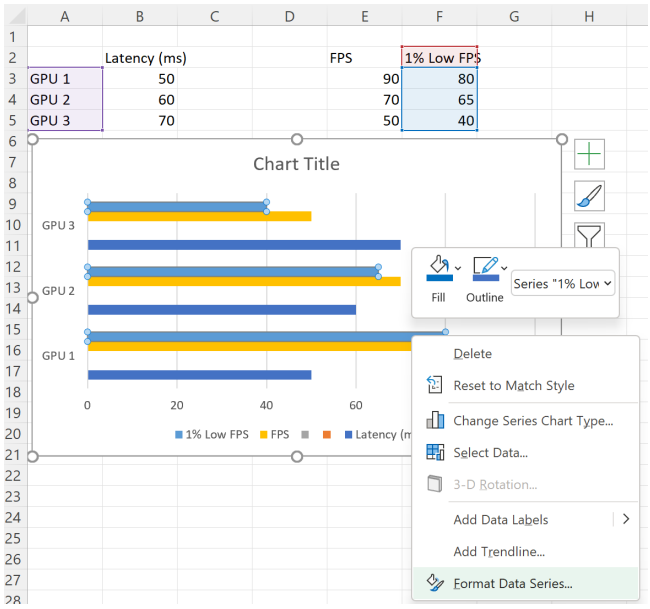
Go ahead and scale the chart and text to a larger size if desired.

a. CTRL + SHIFT + <
b. CTRL + SHIFT + >

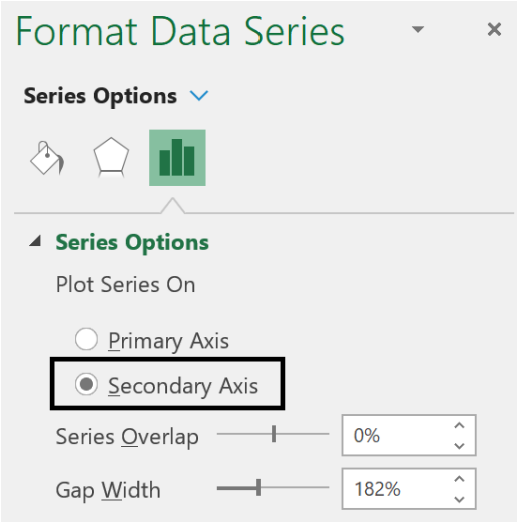


9. Right click the top bar so that it selects **all 1% Low FPS data points** for each GPU and choose **Format Data Series**.

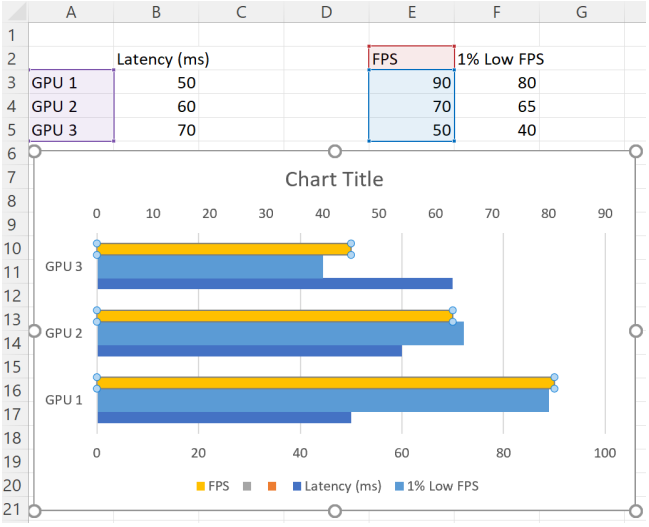
For quick access, do not close the Format Data Series box moving forward.



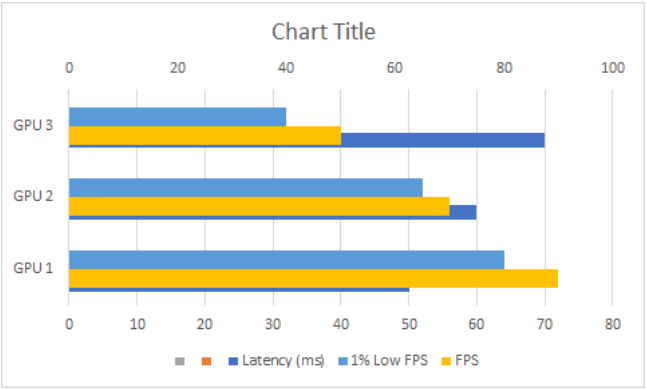
10. Click the radial button for **Secondary Axis**.



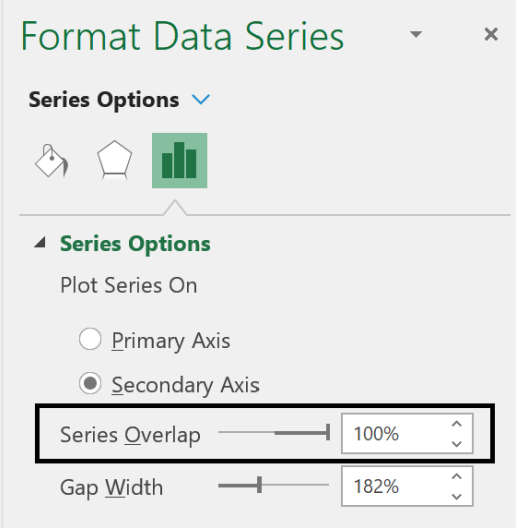
11. Repeat the same step for FPS.



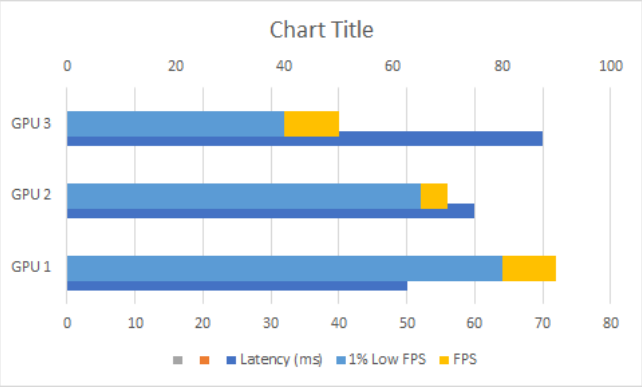
12. The result of choosing **Secondary Axis** for both 1% Low FPS and FPS.



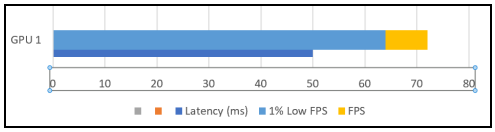
13. Again, select all 1% Low FPS data points for each GPU. Now adjust **Series Overlap** to 100%.



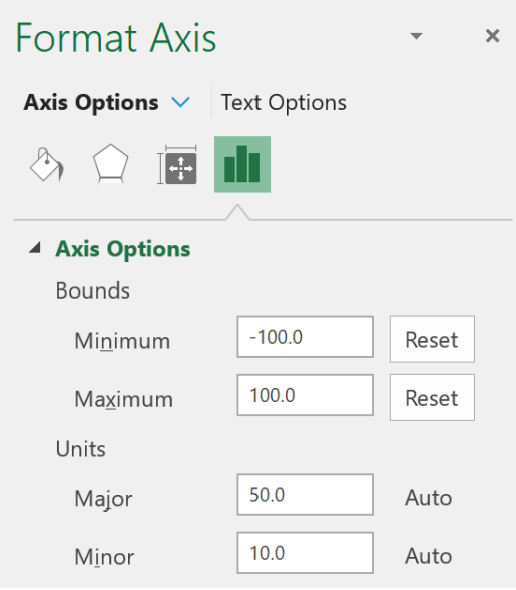
14. The result of adjusting **Series Overlap** to 100%.



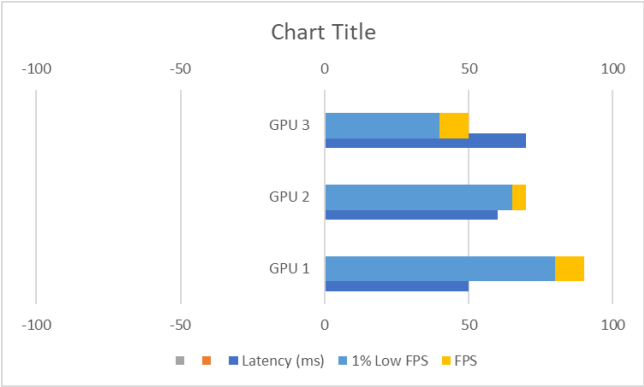
15. Click the bottom axis and set these bounds:
Minimum to -100
Maximum to 100



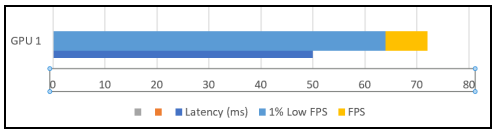
16. Complete the same step for the top axis.

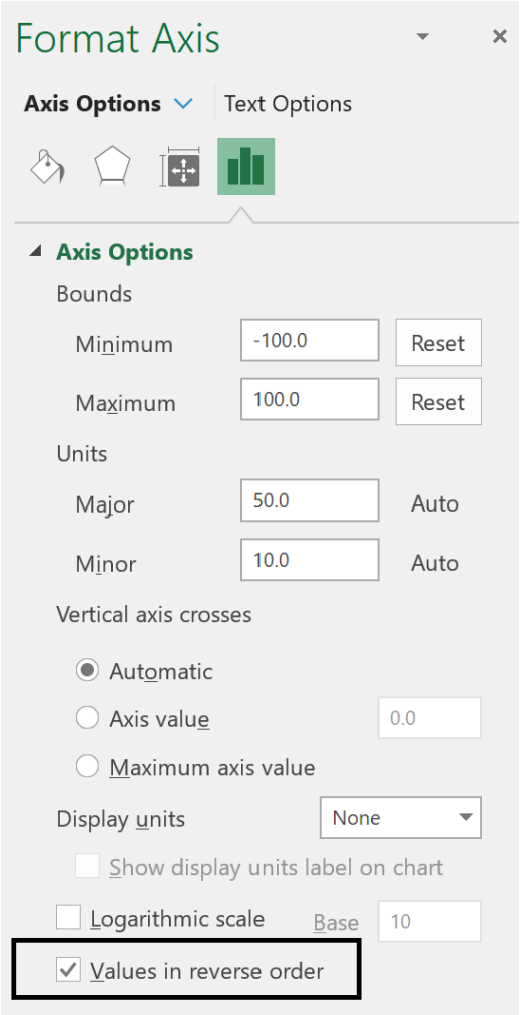


17. The result of modifying axis bounds to fit your data.

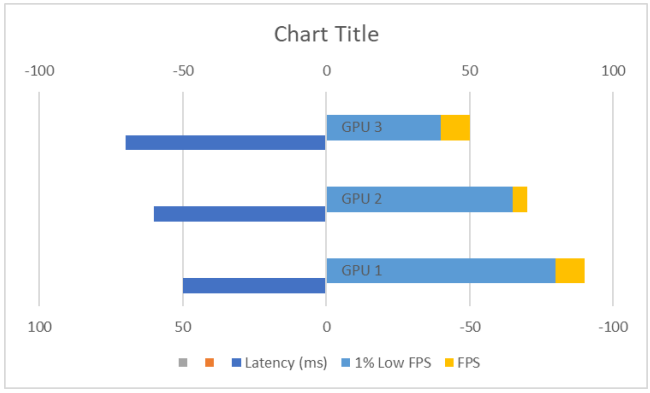


18. Click the bottom axis again but this time check the **Values in reverse order** box.

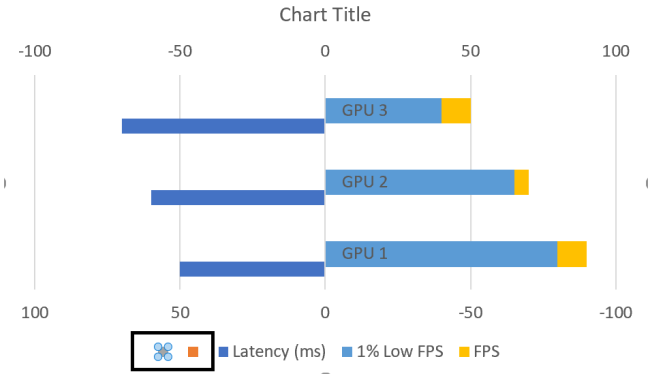




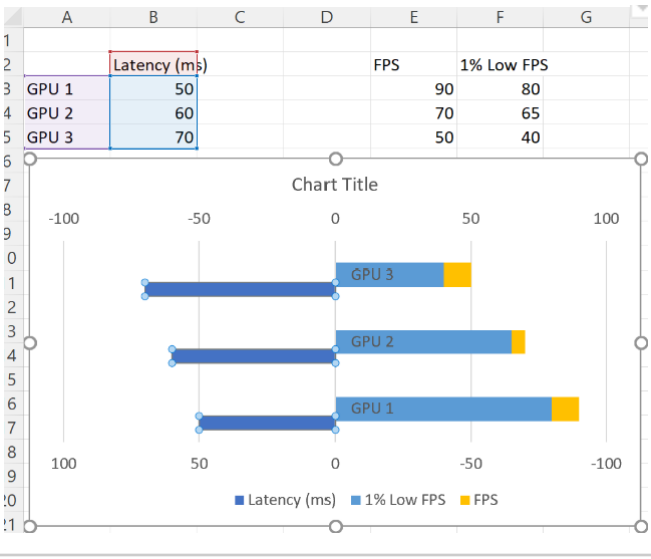
19. The result of checking the box.



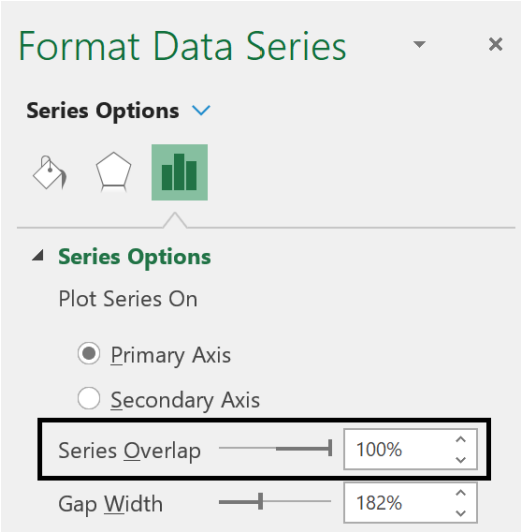
20. In the legend, remove both blank series with the DEL key.



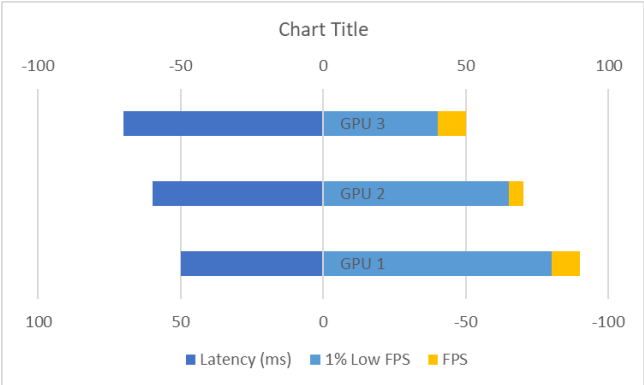
21. Click the latency bars on the left so that all are selected at once.



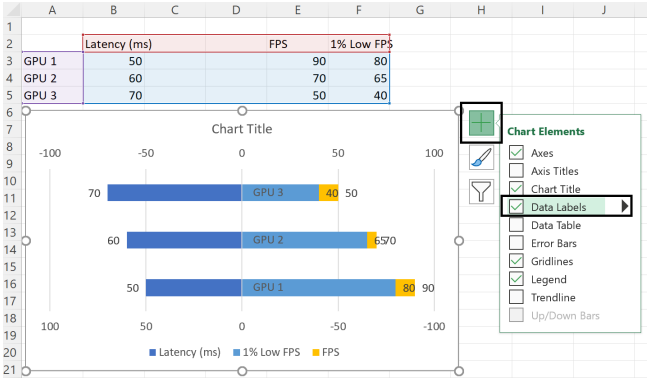
22. Adjust **Series Overlap** to 100%.



23. The result of changing Series Overlap to 100%.

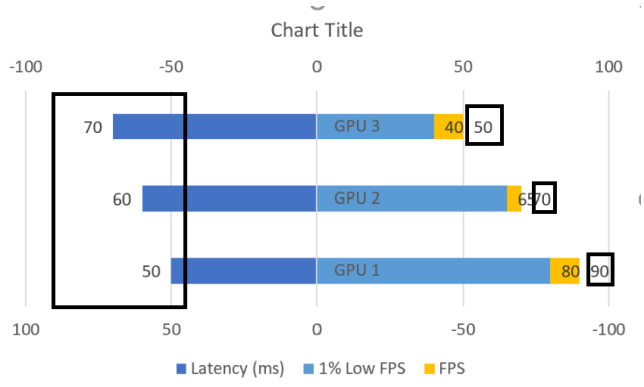


24. With the chart selected, click the **+** at the top right of the chart to add a Chart Element. **Check the Data Labels box.**



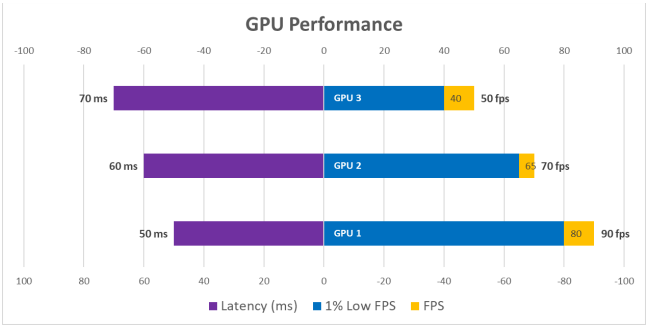
25. Once Data Labels have been created, left-click the **data label/number** on the right three times until "FPS" is highlighted. Now use the arrow keys to move the cursor to the right and add "fps" after each value.

Repeat the same step for the left side, but this time use "ms" to denote latency.

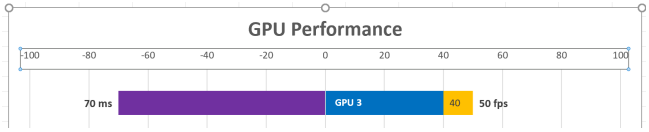


26. Begin beautification of the chart.

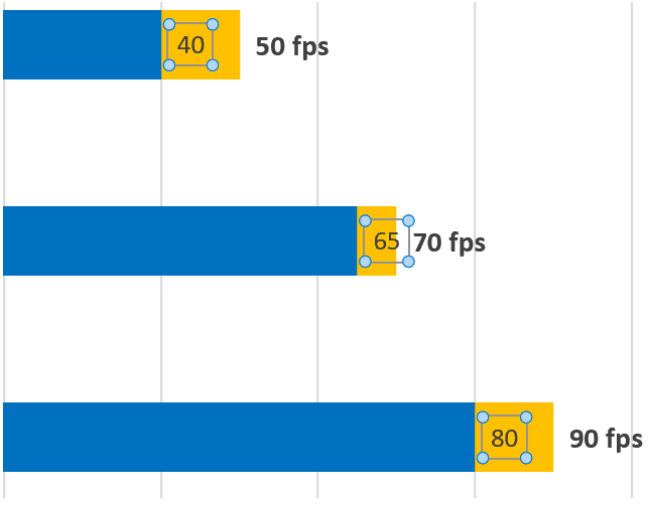
a. Scale the chart to a larger size.
b. Change the text color of the GPU name.
c. Bold text and numbers as needed.
d. Change the size of the chart elements as needed.
e. Change the color of the bars.
f. Type a new Chart Title.
g. Adjust the size of the Chart Title and Legend.
h. Adjust Axis Major and Minor units if necessary.



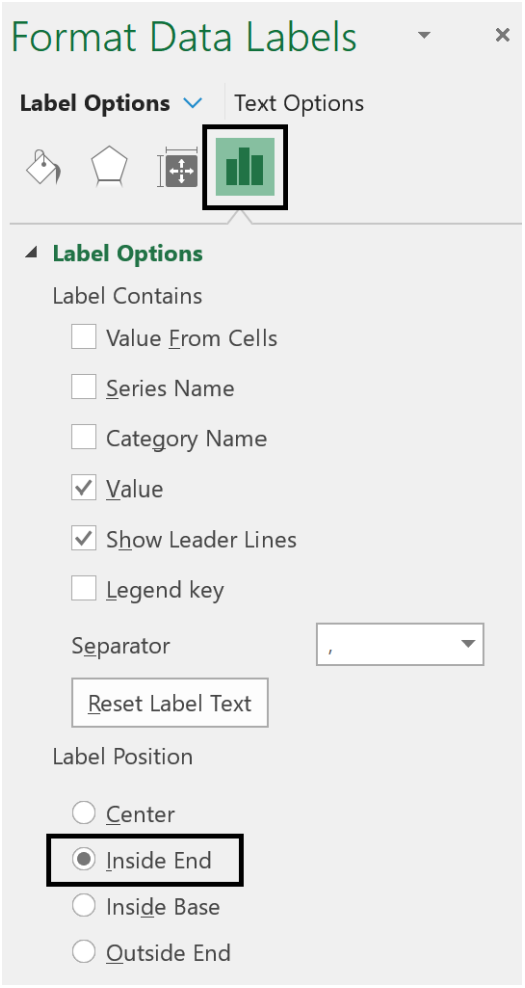
27. Select the top axis and remove it with DEL. Repeat this step for the bottom axis.



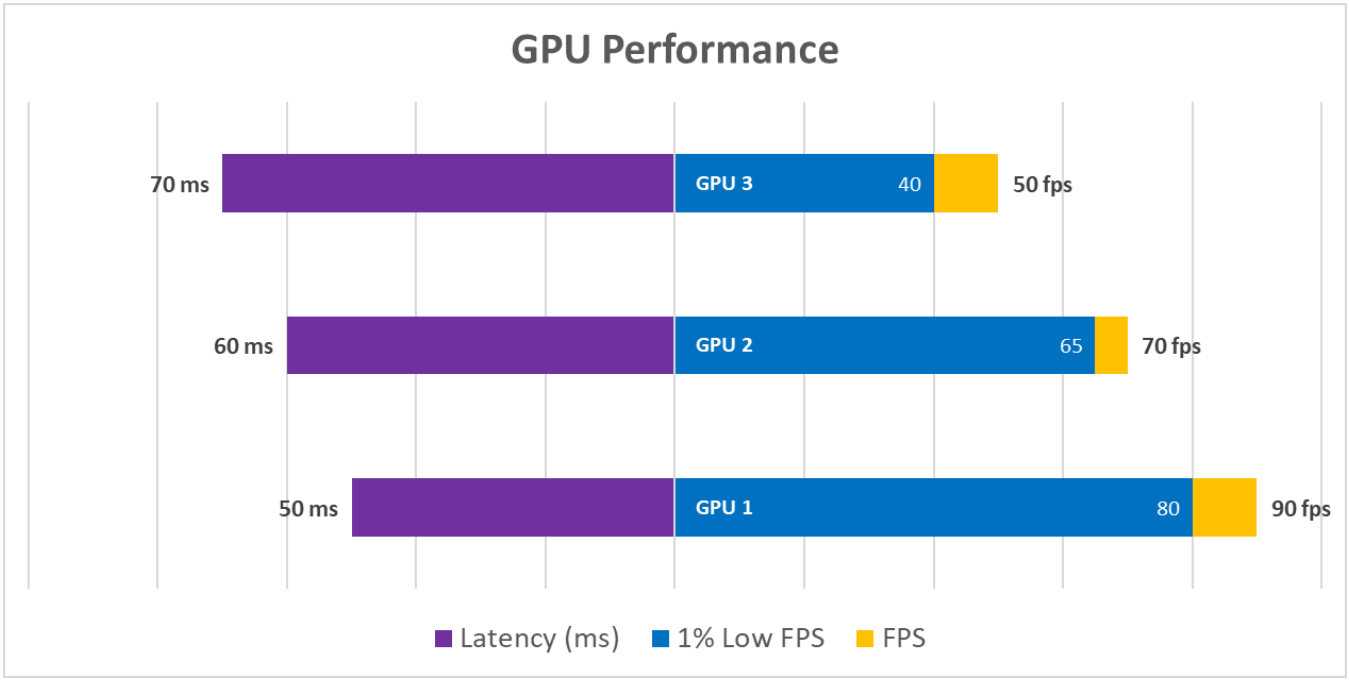
28. Highlight all **1% Low FPS data labels**.



29. With all **1% Low FPS data labels selected**, navigate to **Label Options** and change the **Label Position** to **Inside End**.

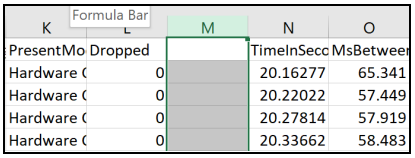


Congrats! You have successfully created the horizontal split chart for visualizing FPS, 1% Low FPS, and Latency all on a single chart.

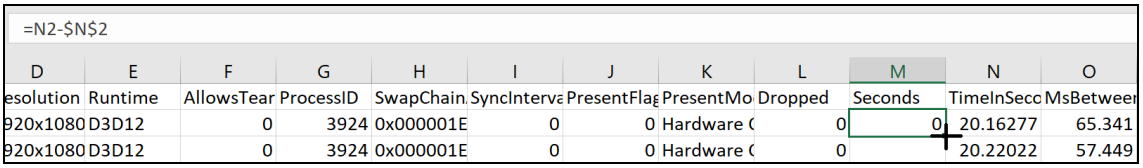


Plotting Battery Life

1. Open the FrameView log file (not the summary file) and "Save as…" a new **.xlsx** extension.
    a. Default location of benchmark files: C:\Users\NVIDIA\Documents\FrameView
2. Open the new **.xlsx** file and **create a new column** called Time next to TimeInSeconds.



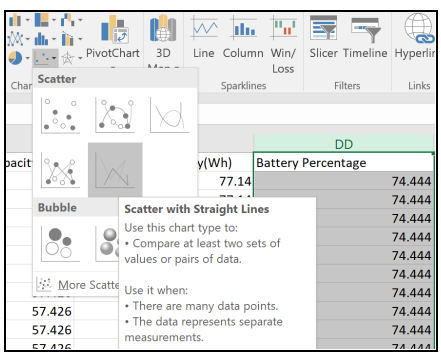
3. In the second row of the new column (M/Seconds), type the formula **=N2-N2** and press ENTER. **Double click the bottom right corner of the 0** to propagate values for all rows. This formula continuously subtracts other TimeInSeconds values from the first TimeInSeconds value in order to get true time for plotting your chart.



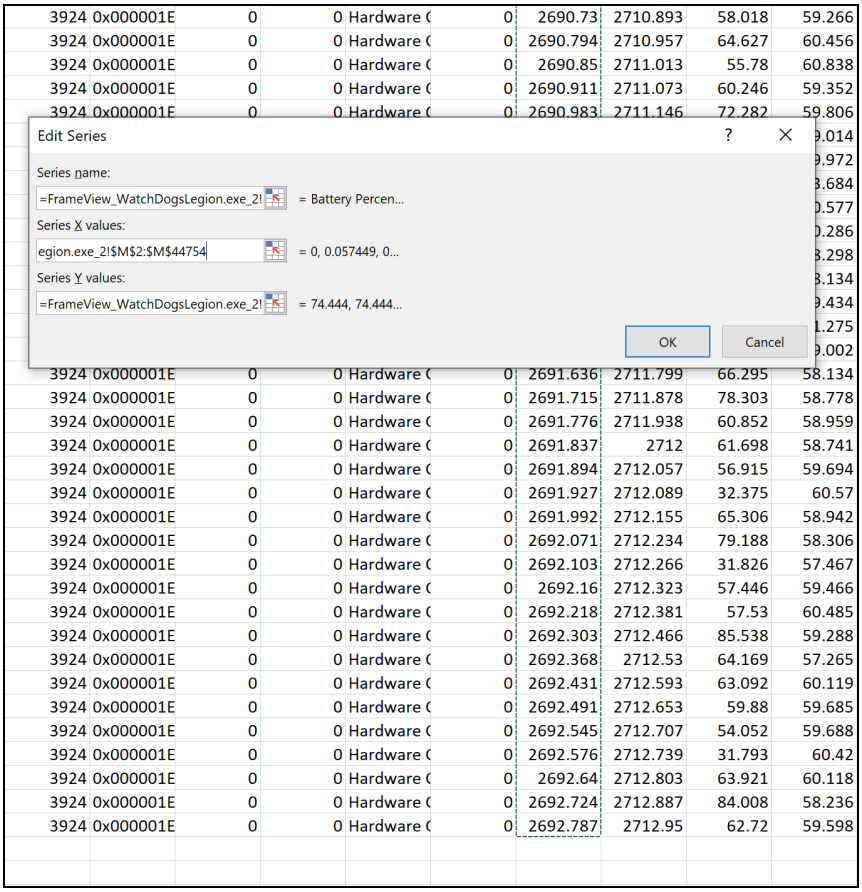
4. Now horizontally scroll to the end of the log to find battery life metrics.

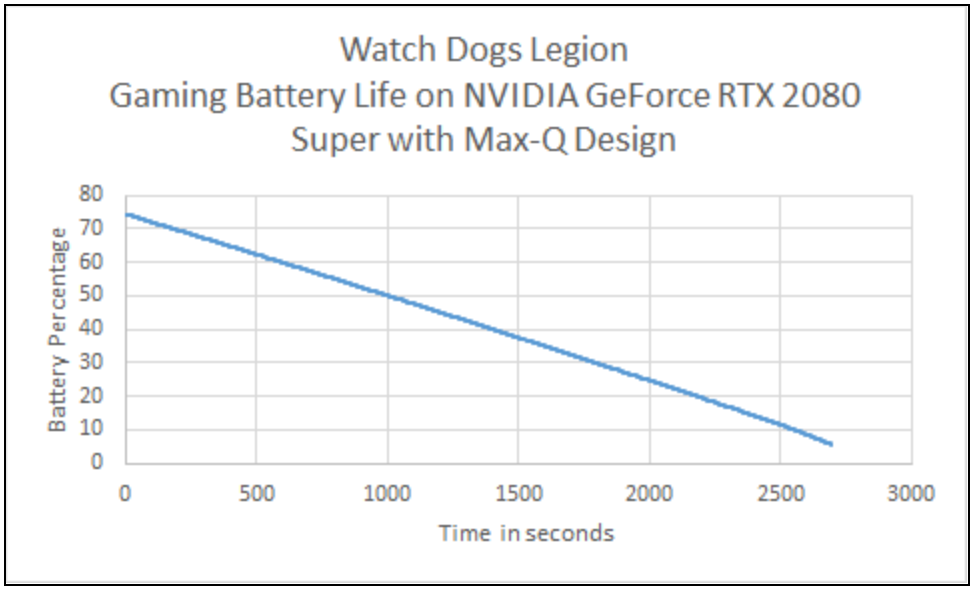
Current Battery Capacity(Wh)	Total Battery Capacity(Wh)	Battery Percentage	Battery Drain Rate(W)
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038
57.426	77.14	74.444	-69.038

5. Highlight the entire Battery Percentage column and add a **Scatter with Straight Lines**.

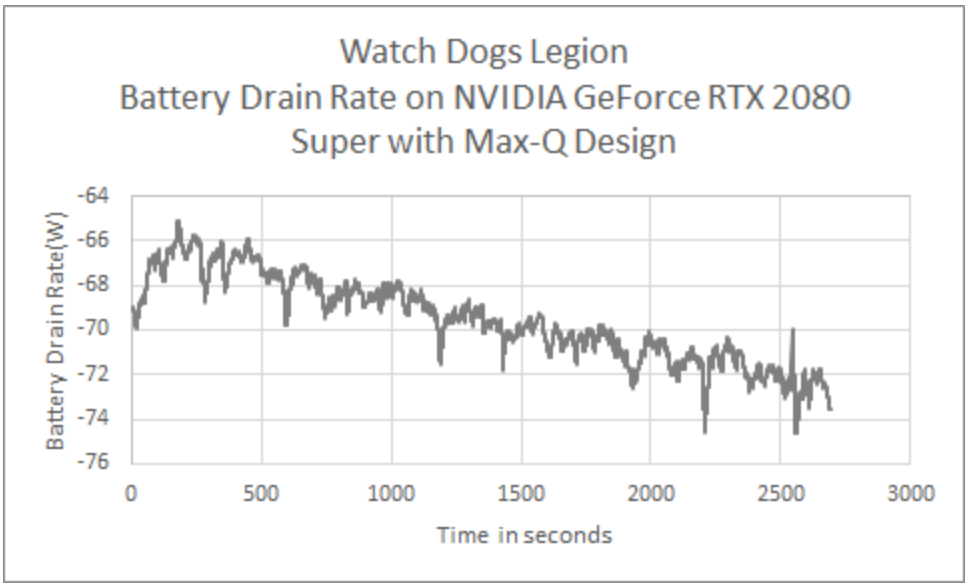


6. Right click the graph, choose Select Data, click Edit, click within the **Series X Values** box and navigate to the newly created Time column using the scrolling bar. Click the first 0 in the newly created Time column then **hold SHIFT + CTRL** at the same time and press **DOWN ARROW** once to select all values for that column. Press ENTER and click OK.





7. You can also plot the battery drain rate.

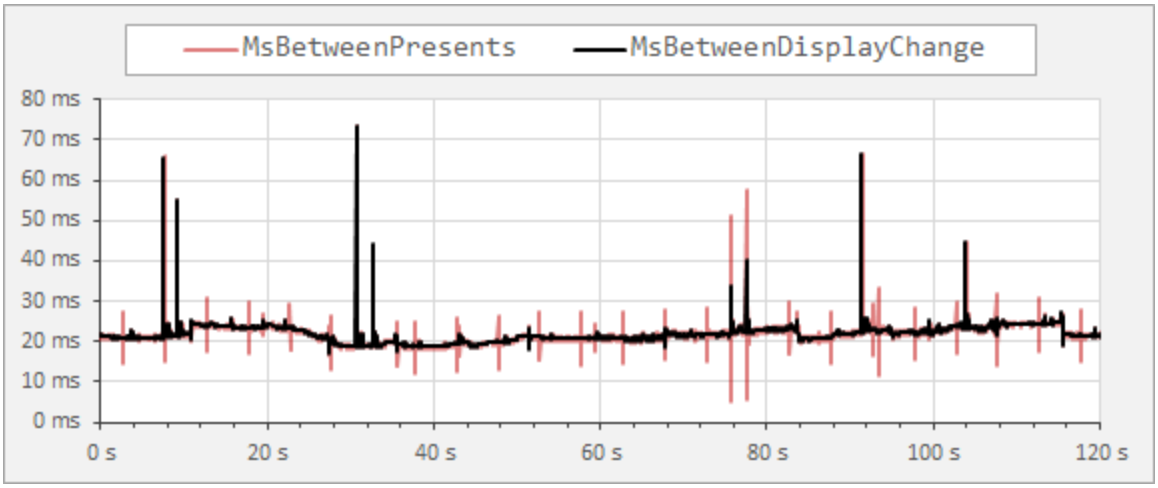


Plotting Frame Time Performance Data

MsBetweenDisplayChange (Displayed FPS) — Should be used to chart displayed FPS. This data is captured from the end of the graphics pipeline and is an indicator of what the user actually sees displayed on screen.

MsBetweenPresents (Rendered FPS) — Can alternatively be used to chart rendered FPS. This data is captured from the beginning of the graphics pipeline and indicates the smoothness of the animation delivered to the GPU. This is the data that is typically provided by other benchmarking capture tools, but can't capture driver-side improvements like frame metering.

The plot below illustrates the difference between these metrics for the same capture:



Plotting Power Data

Since FrameView captures both performance and power data, it allows users to create accurate perf-per-watt statistics to determine GPU efficiency by viewing the performance of the GPU alongside the power it uses. This metric is called performance-per-watt (PPW). The lower the power utilization and the higher the game performance, the better the perf-per-watt.

NVIDIA Power Data

- **GPUOnlyPwr(W) (API)**
  Should be used for charting the average GPU (chip) power consumption.
- **Perf/W GPUOnly(F/J) (API)**
  Should be used for charting performance-per-watt data for NVIDIA GPUs (chip) where F is frames and J is joules (one joule is the equivalent of one watt of power radiated or dissipated for one second). So F/J would be frames per second (F/S) divided by watts (J/S).

  $$F/J \;=\; (F/S) \,/\, (J/S)$$

  For more details on measuring power of GPUs, please refer to the **NVIDIA GeForce GPU Power Primer**.

- **NV-Total-USBCPwr(W) (API)**
  Should be used for charting Total Graphics Power (TGP). TGP is the maximum power in watts that a power supply should provide to the graphics board. TGP is also defined as the average power consumed by the entire graphics board subsystem while executing a very stressful "real world" application. TBP or Total Board Power is essentially the same as TGP. Using this data will be more accurate since it does not include the power used by devices that may be connected to the USB-C connector on NVIDIA GeForce RTX graphics cards.

> FrameView is not as accurate as interposer/riser card techniques for measuring ***idle*** chip or ***idle*** board power. It is accurate for load testing, so it is best to use FrameView when running real-world applications that stress the GPU.

- **Perf/W Total-USBC(F/J) (API)**
  Should be used for charting performance-per-watt data for Total Graphics Power (TGP) where F is frames and J is joules (one joule is the equivalent of one watt of power radiated or dissipated for one second). So F/J would be frames per second (F/S) divided by watts (J/S).

  $$F/J \;=\; (F/S) \,/\, (J/S)$$

For more details on measuring power of GPUs, please refer to the **NVIDIA GeForce GPU Power Primer.**

AMD Power Data

- **AMDPwr(W) (API)**
  Power reported by the AMD API. Should be used for charting the average GPU (board) power consumption for AMD GPUs; however, it is not accurate.

While FrameView accurately reports both chip and board power for NVIDIA graphics cards, the AMD API used by FrameView currently only reports a value that appears to be in-between chip power and board power for AMD graphics cards. Therefore it's currently not possible to use FrameView to directly compare AMD GPU power to NVIDIA GPU power. It may be useful to ask AMD if they can report chip-only and full board power with the APIs, similar to NVIDIA. FrameView will be updated to capture total board power and chip power for AMD if they make such information publicly available in their API.

- **AMD Board Perf-Per-Watt (PPW)**

Since AMD board power cannot be accurately measured with FrameView, performance-per-watt data can not accurately be calculated. Therefore, it's currently not possible to use FrameView to directly compare AMD GPU perf-per-watt data to NVIDIA. It may be useful to ask AMD if they can report chip-only and full board power with the APIs, similar to NVIDIA. FrameView will be updated to capture total board power and chip power for AMD if they make such information publicly available in their API. Note that AMD total board power can be measured by third-party combination hardware/software interposer testing methodologies including NVIDIA PCAT and others. PCAT works in conjunction with FrameView as described above.

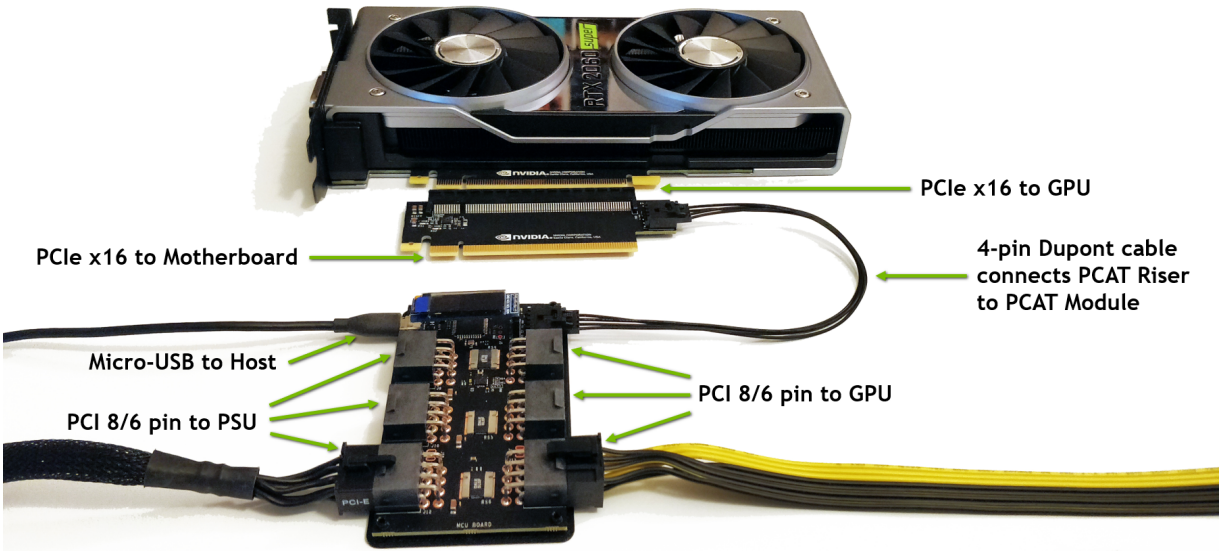
AMD perf-per-watt and total board power can only be measured properly with PCAT or another interposer. The correct values are represented by the **PCAT Power Total(W)** and **Perf/W Total(F/J) (PCAT)** headers in the FrameView log and summary files when using Power Capture Analysis Tool (PCAT).

POWER CAPTURE ANALYSIS TOOL (PCAT)

The PCAT Module contains three 8-pin power connectors on each side with a shunt between each pair of connectors that accurately reads and captures the power data from the PCIe power leads from the PSU to the GPU.

Power data is captured from the PCAT PCIe Riser over the 4-pin cable connecting it to the PCAT Module, and is then combined with the power data captured from the 6- or 8-pin PCIe power cables from the PSU. All of this data is sent to the PC across the micro-USB cable from the PCAT Module.

> Reach out to an NVIDIA PR representative to get your hands on a **PCAT Module** and **User Guide**.

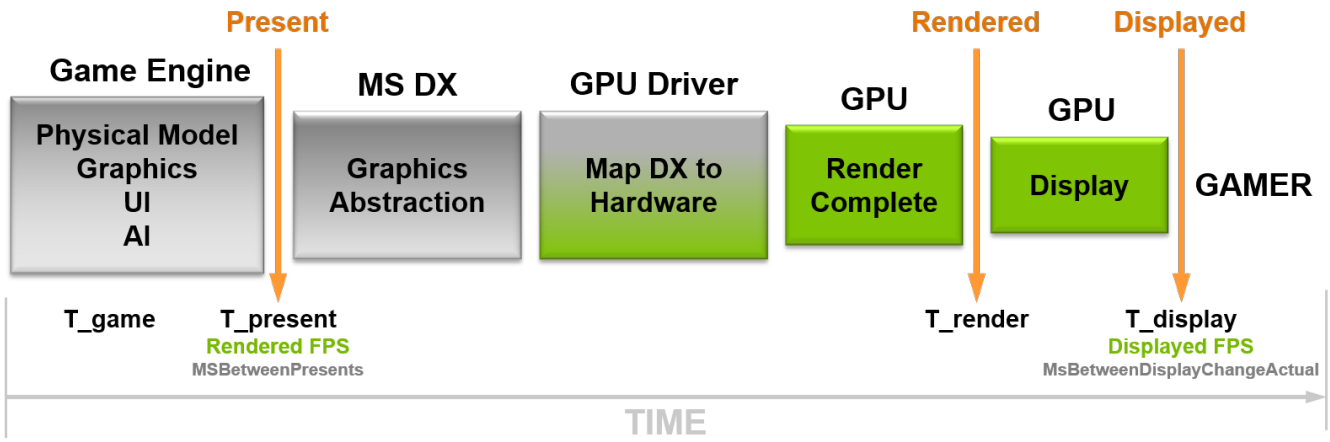


HOW FRAMEVIEW WORKS

To provide performance data for an accurate comparative analysis of GPUs, FrameView measures timestamps at the beginning of the graphics pipeline to provide a metric indicating the smoothness of the animation delivered to the GPU, and at the end of the pipeline to provide an indicator of what the user actually sees displayed on screen.

Frame Rendering Pipeline

The diagram below shows how game frames are created at the beginning of the pipeline and their path to the display.



FrameView provides performance data that is captured in the Present and Displayed portions of the game/graphics pipeline. Data from these two areas will always be reported in the logs. You can choose which you would like shown in the overlay using the FrameView settings in the interface. They are called Rendered FPS and Displayed FPS.

- **Rendered FPS (MsBetweenPresents)** measures timestamps from the beginning of the graphics pipeline and is a metric indicating the smoothness of the animation delivered to the GPU. This is the data that is typically provided by other benchmarking capture tools.

- **Displayed FPS (MsBetweenDisplayChange)** measures timestamps at the end of the game pipeline and is an indicator of what the user actually sees displayed on screen.

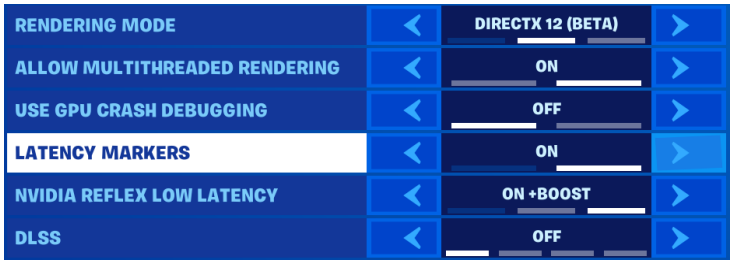
Stutter is the variation between T_game and T_display. This data is also reported by FrameView in the logs. The header is called MsUntilRenderComplete and it measures the time between present start and GPU work completion.

ENABLE LATENCY MARKERS FOR PC LATENCY

If latency markers are disabled, PC Latency might report “N/A” instead of the proper responsiveness of your PC. This section shows you how to enable Latency Markers in *Fortnite* for both NVIDIA and AMD in order for FrameView to properly capture the PCL (PC Latency) metric.

Enable Latency Markers in Fortnite

1. **IMPORTANT:** Enable Latency Markers in Fortnite.
    a. NVIDIA GPU



    b. AMD GPU
        i. If you first tested on NVIDIA (Latency Markers set to On), PC Latency will automatically work on AMD. If not, please do the following:
        ii. Exit the game and navigate to: *C:\Users**UserName**\AppData\Local\FortniteGame\Saved\Config\WindowsClient*
        iii. Open GameUserSettings.ini and set Latency Markers to True. Save the file.
            1. **bLatencyTweak1=True**

```
CustomVoiceChatInputDevice=
CustomVoiceChatOutputDevice=
CustomVoiceChatInputDeviceId=
CustomVoiceChatOutputDeviceId=
bMotionBlur=False
bShowGrass=True
bShowFPS=True
bUseGPUCrashDebugging=False
bLatencyTweak1=True
LatencyTweak2=0
bLatencyFlash=True
DLSSQuality=0
```

TROUBLESHOOTING

The FrameView overlay is not being displayed over a game

Overlay information will not be shown in DX9/10. However, data capture is supported and the information will be properly logged after the completion of a benchmark or at the end of a session.

If another performance tool such as Fraps or RTSS is running, the FrameView overlay may not display.

Running FrameView and RTSS/FRAPS simultaneously

Since FrameView and RTSS/FRAPS both are hooking into similar application processes, FrameView might not work with x64-bit applications if RTSS/FRAPS is already running in the background. To remedy this and run these applications together, launch each process in the following order:

- Launch FrameView
- Launch Game
- Let FrameView overlay appear
- Launch RTSS/FRAPS
- Close RTSS/FRAPS before closing the game, and then follow steps 1-4 for the next run

FrameView reporting invalid or NA data for NVIDIA GPU specific metrics

After any new driver installation or update, FrameView may report invalid or NA data. To workaround this issue, reboot the system. An upcoming version of FrameView will address this known issue.

FrameView overlay is appearing on or logging for the wrong applications

If the overlay is appearing on multiple unintended applications, the user simply has to disable the application by editing the exclude list. To disable the overlay for a particular process, add the executable name to %ProgramData%\NVIDIA Corporation\FrameView\ExcludeList.overlay.txt

To disable logging for a particular process, add executable name to %ProgramData%\NVIDIA Corporation\FrameView\ExcludeList.txt. This will also disable overlay for the process.

PCL is showing as NA on some hardware configurations for supported title

First, please ensure latency markers are turned on in the settings, if the setting is available. If the issue persists, hang tight–we are working with developers to enable PC latency measurement for all vendors and will improve compatibility in a future version

NVIDIA CONTACT INFORMATION

NVIDIA North/Latin America Public Relations

Bryan "BDR" Del Rizzo Senior PR Manager, Desktop & Notebook GeForce Mobile: 510 331 8824 bdelrizzo@nvidia.com	**Brandon Bell** Senior Technical Marketing Manager – Desktop GeForce GPUs Mobile: 512 363 6698 branbell@nvidia.com
Matthew Widener Technical Marketing Senior Manager, GeForce Mobile: 831 419 9253 mwidener@nvidia.com	**Sean Cleveland** Director, Technical Marketing, GeForce Mobile: 831 402 0145 scleveland@nvidia.com
Rajal Maharaj Technical Marketing Manager, GeForce Mobile: 510 579 9111 rmaharaj@nvidia.com	**Anthony Giurbino** Technical Marketing Manager, GeForce – Monitor Technology and Reflex Mobile: 408 242 3317 agiurbino@nvidia.com
Alexandre Ziebert Technical Marketing Manager, Latin America Mobile: +55 11 96630 1074 aziebert@nvidia.com	**Andre Forte** PR Manager, Latin America Mobile: +55 11 97148 1061 aforte@nvidia.com

NVIDIA Europe Public Relations

Benjamin Berraondo Senior Product PR Manager – GeForce & Gaming EMEA Office: +44 118 918 4350 Mobile: +44 7979 384482 bberraondo@nvidia.com	**Christian Beer** PR & Technical Product Manager - SHIELD and GeForce Now, EMEA Office: +49 6283 50055 Cell: +49 162 2164644 cbeer@nvidia.com
Boris Böhles PR Manager, DACH & BENELUX Office: +49 6283 50059 Cell: +49 151 41938777 bboehles@nvidia.com	**Lars Weinand** Senior Technical Product Manager, DACH & BENELUX Mobile: +49 173 7311540 lweinand@nvidia.com
Jen Andersson PR Manager, UK and Nordics Office: +44 (0)118 9184358 Mobile: +44 (0)7799 483 329 jandersson@nvidia.com	**Rick Napier** Senior Technical Product Manager, UK and Nordics Mobile: +44 (0)7917) 630172 rnapier@nvidia.com

Stephane Quentin Senior PR Manager - France Office : +33 155 638 493 Mobile: +33 6 825 68322 squentin@nvidia.com	**Sébastien Januario** Senior Technical Product Manager – Laptops, EMEA Office: +33 (0) 1 55 63 16 51 Mobile: +33 (6) 65 44 91 03 sjanuario@nvidia.com
Michele Gennari PR Manager- Italy, Greece & Israel Office: +39 0200618577 Mobile: +39 3395630576 mgennari@nvidia.com	**Luciano Ballerano** PR Manager- Italy, Greece & Israel Office: +39 0200618577 Mobile: +39 3666760288 lballerano@nvidia.com
Juan Carlos González PR Manager - Spain and Portugal Mobile: +34 670034506 juang@nvidia.com	**Jan Adryański** Community PR Manager, Central Eastern Europe Mobile: +48 574201494 Skype: soushiboyt jadryanski@nvidia.com
Igor Stanek Senior Technical Product Manager EMEAI – Gaming Cell: +420602135136 Istanek@nvidia.com	**Oleg Shkoda** Tech Marketing RU Office: +7 495 981 03 00 ext. 10795 Mobile: +7 929 663 60 56 oshkoda@nvidia.com

NVIDIA APAC Public Relations

Jeff Yen Director, Technical Marketing, APAC Office: +886 987 263 193 jyen@nvidia.com NVIDIA 8, Kee Hu Road, Neihu Taipei 114, TAIWAN	**Searching Shi** Sr. Technical Marketing Manager, China Office: +86-10 5866 1500 seshi@nvidia.com NVIDIA China 1/F, Productivity Building, #5 Hi-Tech Middle 2nd Road, Shenzhen High-Tech IND Park Nanshan District, Shenzhen CHINA
Roy Zhu Technical Marketing Manager, China Office: +86-10 5866 1322 royz@nvidia.com NVIDIA China Fortune Financial Center Level 40, Units: 01, 02, 03-1 Building #5, Middle Road, East 3rd Ring Chaoyang District, Beijing CHINA	**Jade Li** PR Manager, China Office: +86-10 5866 1322 Jadli@nvidia.com NVIDIA China Fortune Financial Center Level 40, Units: 01, 02, 03-1 Building #5, Middle Road, East 3rd Ring Chaoyang District, Beijing CHINA

Masaki Sawai Technical Marketing Manager, Japan Office: +81 3 6743 8699 msawai@nvidia.com ATT New Tower 13F 2-11-7 Akasaka,Minato-ku, Tokyo 107-0052 JAPAN	**Kaori Nakamura** Head of Public Relations, Japan Office: +81 3 6743 8712 knakamura@nvidia.com ATT New Tower 13F 2-11-7 Akasaka,Minato-ku, Tokyo 107-0052 JAPAN
Kyle Kim Sr. Technical Marketing Manager, Korea Office: +82 2 6001 7186 kylek@nvidia.com NVIDIA Korea #2101, COEX Trade Tower, 159-1 Samsung-dong Kangnam-gu, Seoul 135-729 KOREA	**Sunny Lee** Marketing Director, Korea Office: +82 2 6001 7123 slee@nvidia.com NVIDIA Korea #2101, COEX Trade Tower, 159-1 Samsung-dong Kangnam-gu, Seoul 135-729 KOREA
John Gillooly Technical Marketing Manager, Asia Pacific South Office : +65 8322 3075 jgillooly@nvidia.com NVIDIA Singapore Regus Galaxis 1 Fusionopolis Place , #3-20 Galaxis (West Lobby) SINGAPORE	**Mary Chin** PR Manager, ROAP (TW/AU/SEA) Office: +886 2 6605 5323 mchin@nvidia.com NVIDIA 8, Kee Hu Road, Neihu Taipei 114 TAIWAN
Titus Su Technical Marketing Engineer, TASA Office : +886 (2) 66055430 tisu@nvidia.com NVIDIA Taiwan 8, Kee Hu Road, Neihu Taipei 114 TAIWAN	

LEGAL

Notice

ALL INFORMATION PROVIDED IN THIS REVIEWER'S GUIDE, INCLUDING COMMENTARY, OPINION, NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, Battery Boost, GeForce Experience, FrameView, RTX, and GeForce are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright