# NVIDIA® FCAT VR

REVIEWER'S GUIDE

# TABLE OF CONTENTS

Table of Contents ..... 2

FCAT VR ..... 4

Measuring the Quality of your VR Experience ..... 4

FCAT VR Capture ..... 4

FCAT VR Analyzer ..... 5

Installation ..... 6

FCAT VR Directory Structure ..... 6

Unpack FCAT VR Capture ..... 6

Unpack FCAT VR Analyzer ..... 6

FCAT VR Capture ..... 8

Installing FCAT VR Capture ..... 8

Installing on HTC Vive ..... 8

Installing on Oculus Rift ..... 10

Using FCAT VR Capture ..... 11

FCAT VR Data Folder ..... 13

FCAT VR Analyzer ..... 15

Installing Anaconda & pyqtgraph ..... 15

Anaconda ..... 15

pyqtgraph ..... 16

Running FCAT VR Analyzer ..... 18

File Association ..... 18

Launching FCAT VR Analyzer ..... 20

App Regions & Elements ..... 22

Input Files or Directories Region ..... 22

Data Region ..... 23

Log Region ..... 24

Plot Region ..... 24

Manipulating Charts ..... 25

Using the Mouse ..... 25

Plot Menu ..... 26

**FCAT VR Benchmarking Guide.....34**
Game Recommendations & Performance Results.....34

**FCAT VR Capture | How it Works.....35**
Ideal VR Pipeline.....35
Dropped Frames.....36
Synthesized Frame.....37
Warp Misses.....37
FrameTime.....38
Oculus Rift | FCAT VR Capture Details.....38
Asynchronous Spacewarp (ASW).....39
HTC Vive | FCAT VR Capture Details.....42
Reprojection.....43

## Measuring the Quality of your VR Experience

Smoothness is important for VR gaming, and users can evaluate stutter that might affect game fluidity in a few different ways. The most important method is simply to experience VR and evaluate how it feels. Does it hitch? Is there stutter? Pan the view around and feel how smooth it is in motion. However, this method is subjective.

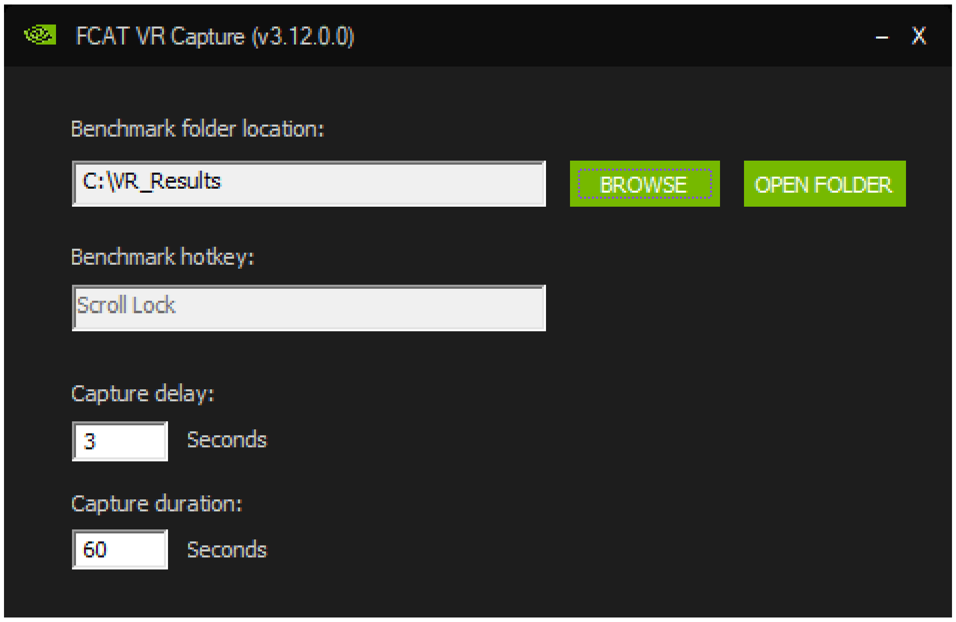
Our newest iteration of FCAT, called FCAT VR, supports Virtual Reality (VR) by adding capture support for Oculus Rift and HTC Vive. FCAT analysis takes the guesswork out of VR performance testing since it is an objective, data-based process.

In the past, NVIDIA has made FCAT freely modifiable and redistributable, and we have seen third-parties adopt FCAT to analyze their own applications. FCAT VR will be just as open.

**FCAT VR is comprised of two packages: FCAT VR Capture and the FCAT VR Analyzer.**

1. **FCAT VR Capture** – Use this software to capture VR gameplay.
2. **FCAT VR Analyzer** – Use this software to turn captured raw data into human-readable charts.

### FCAT VR Capture


FCAT VR Capture (v3.12.0.0)

Benchmark folder location: C:\VR_Results

Benchmark hotkey: Scroll Lock

Capture delay: 3 Seconds

Capture duration: 60 Seconds

FCAT VR Capture is a new frametime capture tool with a user interface similar to other frame capture tools (such as FRAPS), but it uses NVIDIA driver stats, Oculus Event Tracing for Windows (ETW) events, or SteamVR's performance API data (for HTC Vive) to generate precise VR performance data. FCAT VR Capture works for all GPUs independent of GPU vendor.

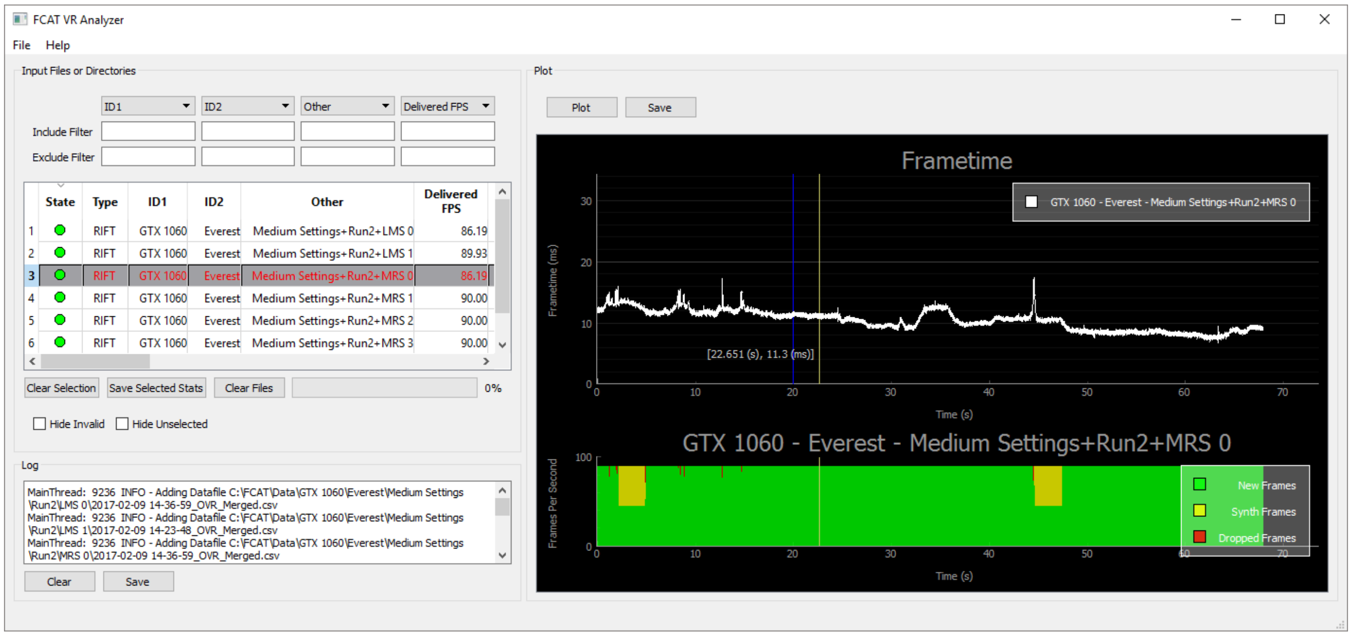
Most importantly, FCAT VR Capture works with popular VR Head-Mounted Displays (HMDs) to capture key performance events:

- Total frametime
- Application dropped frames
- Runtime warp dropped frames
- Asynchronous Space Warp (ASW) synthesized frames

From these events we can draw meaningful conclusions about the GPU performance and the VR experience.

**NOTE:** Using hardware capture requires advanced setup and additional hardware. For more information, please see the **FCAT VR Hardware Capture Guide**.

## FCAT VR Analyzer



| | State | Type | ID1 | ID2 | Other | Delivered FPS |
|---|---|---|---|---|---|---|
| 1 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+LMS 0 | 86.19 |
| 2 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+LMS 1 | 89.93 |
| 3 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+MRS 0 | 86.19 |
| 4 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+MRS 1 | 90.00 |
| 5 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+MRS 2 | 90.00 |
| 6 | ● | RIFT | GTX 1060 | Everest | Medium Settings+Run2+MRS 3 | 90.00 |

```
MainThread:  9236  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings
\Run2\LMS 0\2017-02-09 14-36-59_OVR_Merged.csv
MainThread:  9236  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings
\Run2\LMS 1\2017-02-09 14-23-48_OVR_Merged.csv
MainThread:  9236  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings
\Run2\MRS 0\2017-02-09 14-36-59_OVR_Merged.csv
```

Data generated by FCAT VR Capture can be easily dragged into the FCAT VR Analyzer so that essential data including frametime, dropped frames, synthesized (ASW) frames, and much more can be extracted and easily compared to other VR data. The analyzer also allows for the simple creation of frametime and dropped frame charts that can be saved and exported.

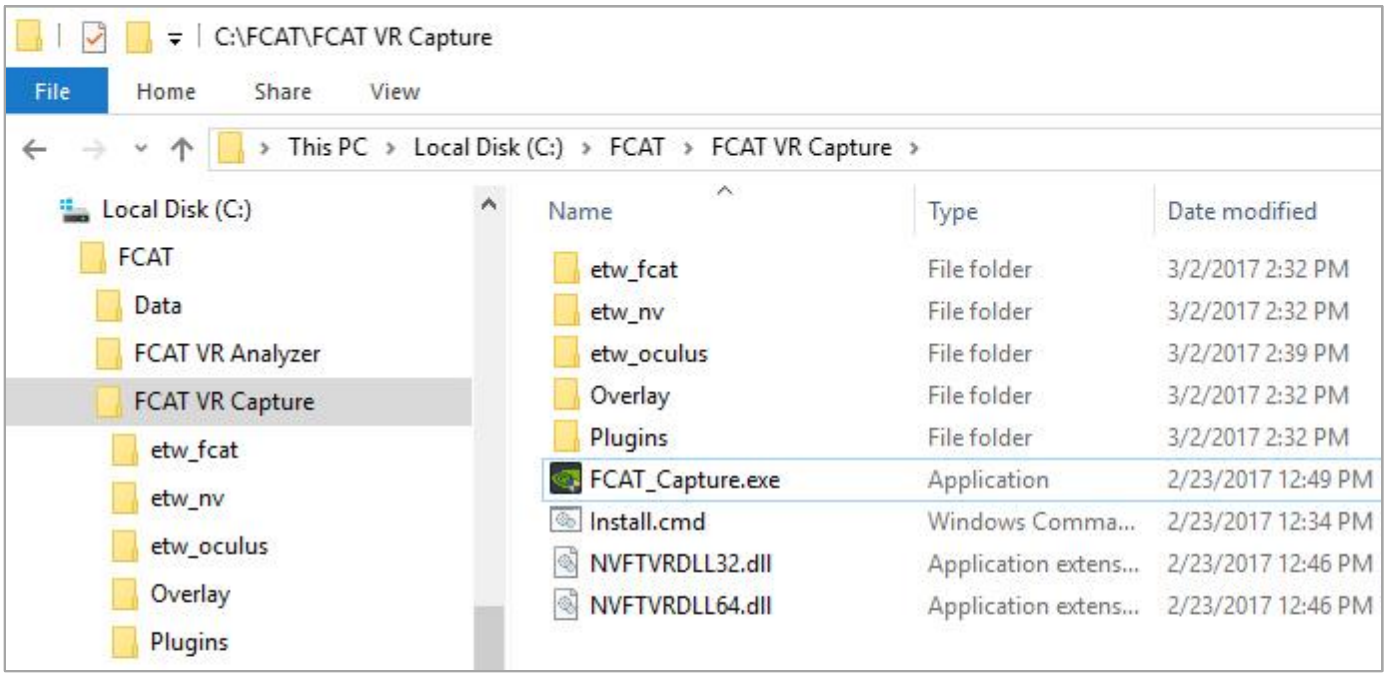
## FCAT VR Directory Structure

Because the capture and analyzer software are both delivered in .ZIP files, we have created this section to help place and organize the files so the installation process can be more easily followed.

### Unpack FCAT VR Capture

The FCAT VR Capture software is delivered in a .ZIP file. Download and unpack that file to the VR capture station where your HMD is installed. The software can run from anywhere.

We recommend placing the files in a folder named **FCAT VR Capture** in **C:\FCAT** as shown below.

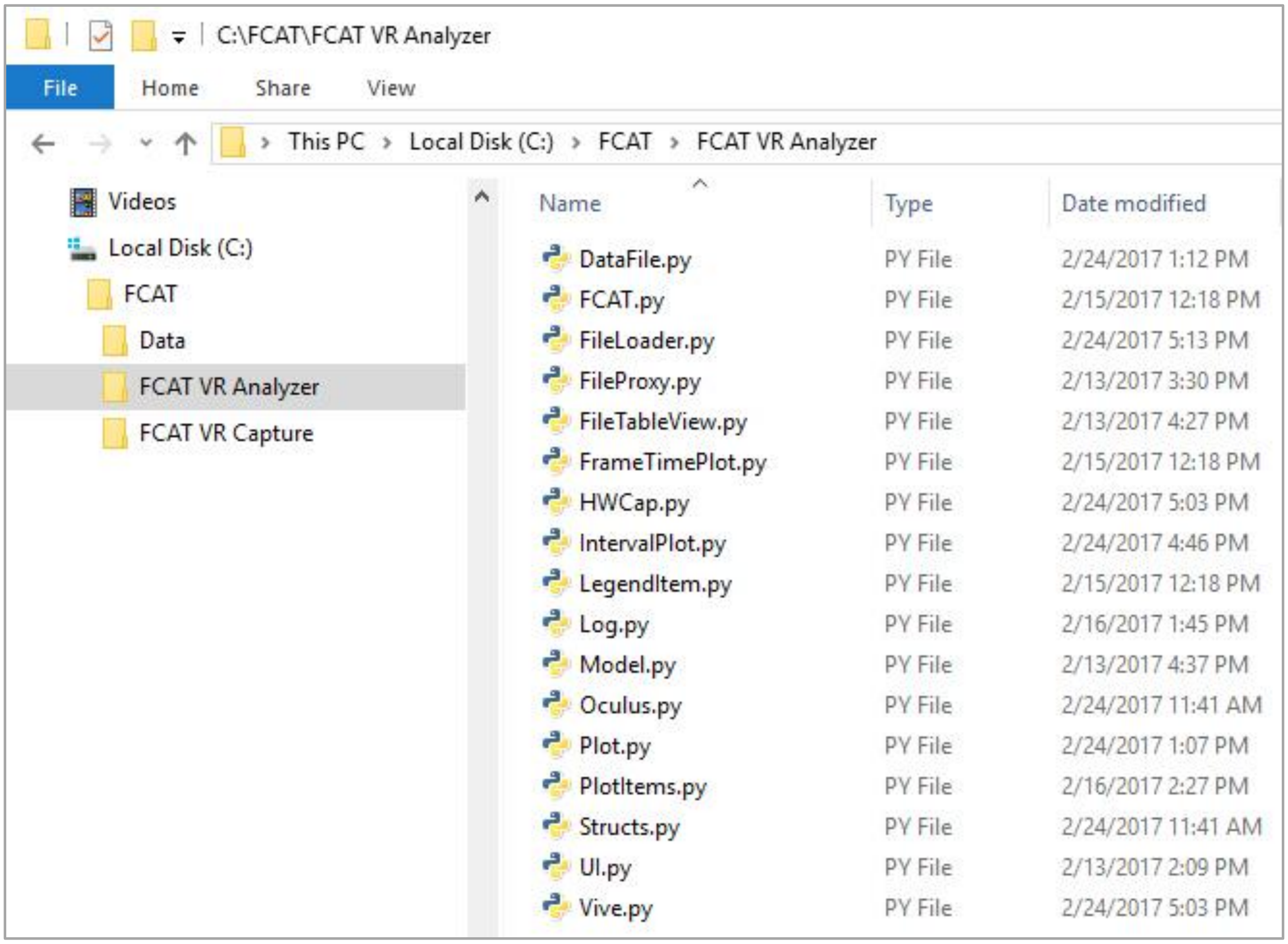


| Name | Type | Date modified |
|---|---|---|
| etw_fcat | File folder | 3/2/2017 2:32 PM |
| etw_nv | File folder | 3/2/2017 2:32 PM |
| etw_oculus | File folder | 3/2/2017 2:39 PM |
| Overlay | File folder | 3/2/2017 2:32 PM |
| Plugins | File folder | 3/2/2017 2:32 PM |
| FCAT_Capture.exe | Application | 2/23/2017 12:49 PM |
| Install.cmd | Windows Comma... | 2/23/2017 12:34 PM |
| NVFTVRDLL32.dll | Application extens... | 2/23/2017 12:46 PM |
| NVFTVRDLL64.dll | Application extens... | 2/23/2017 12:46 PM |

### Unpack FCAT VR Analyzer

The FCAT VR Analyzer is also delivered in a ZIP file. Unpack this file on the system you plan on using to analyze the captured VR data. This can be the same or a separate machine.

We recommend creating an **FCAT VR Analyzer** folder in **C:\FCAT** as shown below.



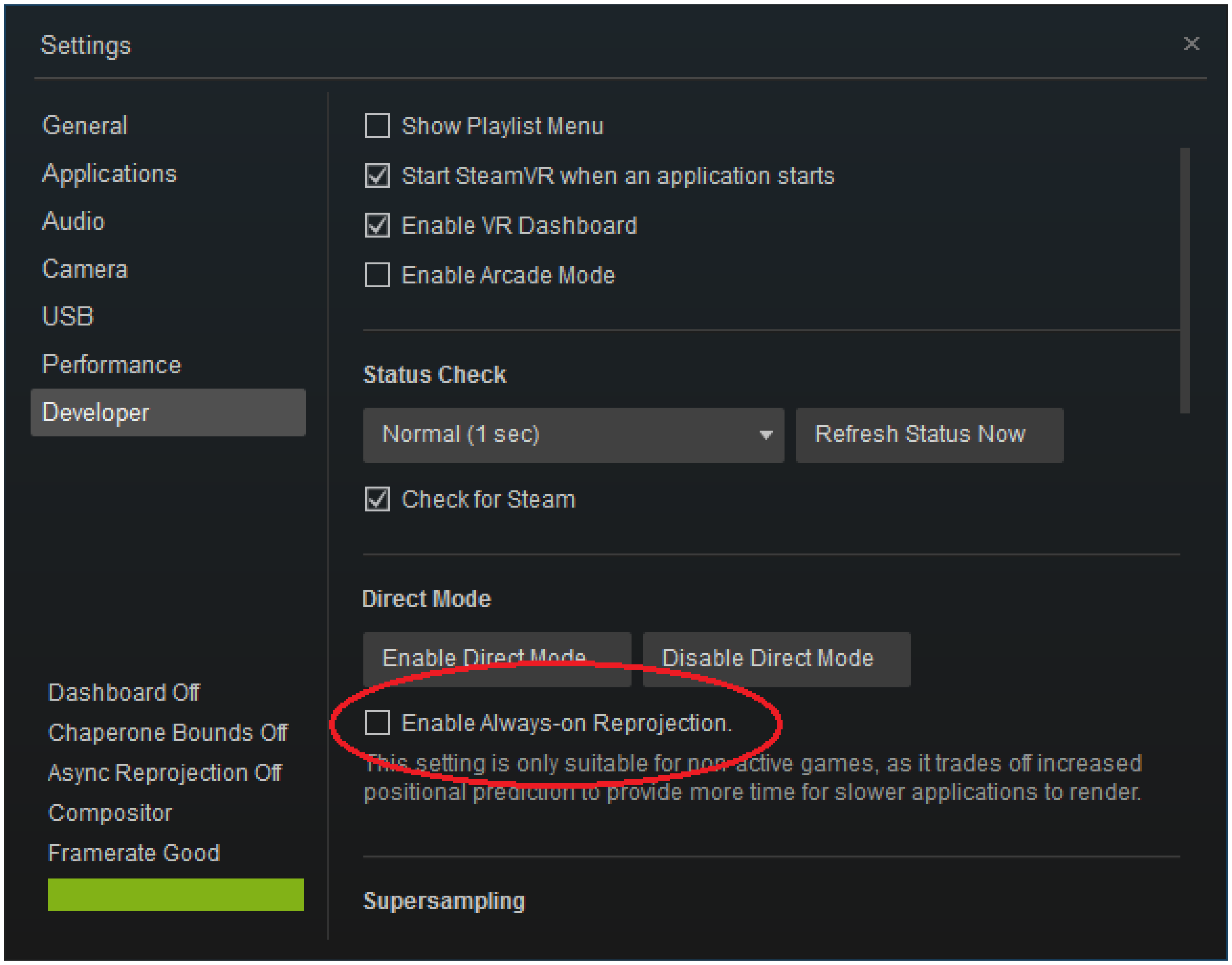
| Name | Type | Date modified |
|---|---|---|
| DataFile.py | PY File | 2/24/2017 1:12 PM |
| FCAT.py | PY File | 2/15/2017 12:18 PM |
| FileLoader.py | PY File | 2/24/2017 5:13 PM |
| FileProxy.py | PY File | 2/13/2017 3:30 PM |
| FileTableView.py | PY File | 2/13/2017 4:27 PM |
| FrameTimePlot.py | PY File | 2/15/2017 12:18 PM |
| HWCap.py | PY File | 2/24/2017 5:03 PM |
| IntervalPlot.py | PY File | 2/24/2017 4:46 PM |
| LegendItem.py | PY File | 2/15/2017 12:18 PM |
| Log.py | PY File | 2/16/2017 1:45 PM |
| Model.py | PY File | 2/13/2017 4:37 PM |
| Oculus.py | PY File | 2/24/2017 11:41 AM |
| Plot.py | PY File | 2/24/2017 1:07 PM |
| PlotItems.py | PY File | 2/16/2017 2:27 PM |
| Structs.py | PY File | 2/24/2017 11:41 AM |
| UI.py | PY File | 2/13/2017 2:09 PM |
| Vive.py | PY File | 2/24/2017 5:03 PM |

# Installing FCAT VR Capture

## Installing on HTC Vive

Please follow these instructions when using FCAT VR Capture with the HTC Vive.

1. Ensure that all graphic drivers are properly installed.
2. Start **SteamVR**, click the **File** menu, and click **Settings**.
3. Click **Developer**, then clear **Enable Always-on Reprojection**



Settings

General
Applications
Audio
Camera
USB
Performance
Developer

- [ ] Show Playlist Menu
- [x] Start SteamVR when an application starts
- [x] Enable VR Dashboard
- [ ] Enable Arcade Mode

Status Check

Normal (1 sec) | Refresh Status Now

- [x] Check for Steam

Direct Mode

Enable Direct Mode | Disable Direct Mode

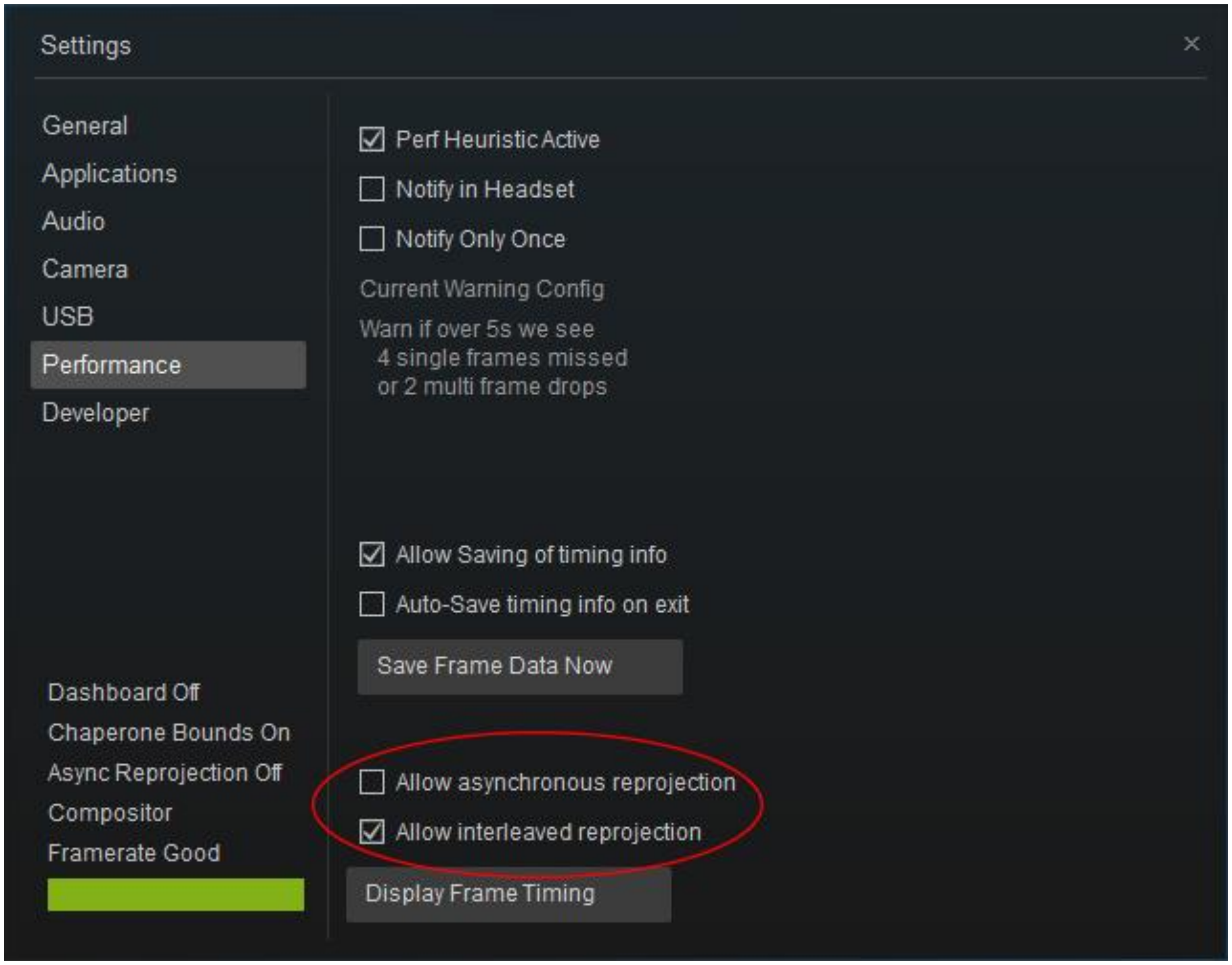
- [ ] Enable Always-on Reprojection.

This setting is only suitable for non-active games, as it trades off increased positional prediction to provide more time for slower applications to render.

Supersampling

Dashboard Off
Chaperone Bounds Off
Async Reprojection Off
Compositor
Framerate Good

4. Click **Performance**, then clear **Allow asynchronous reprojection** and select **Allow interleaved reprojection**.



Settings

- General
- Applications
- Audio
- Camera
- USB
- Performance
- Developer

- [✓] Perf Heuristic Active
- [ ] Notify in Headset
- [ ] Notify Only Once

Current Warning Config
Warn if over 5s we see
4 single frames missed
or 2 multi frame drops

- [✓] Allow Saving of timing info
- [ ] Auto-Save timing info on exit

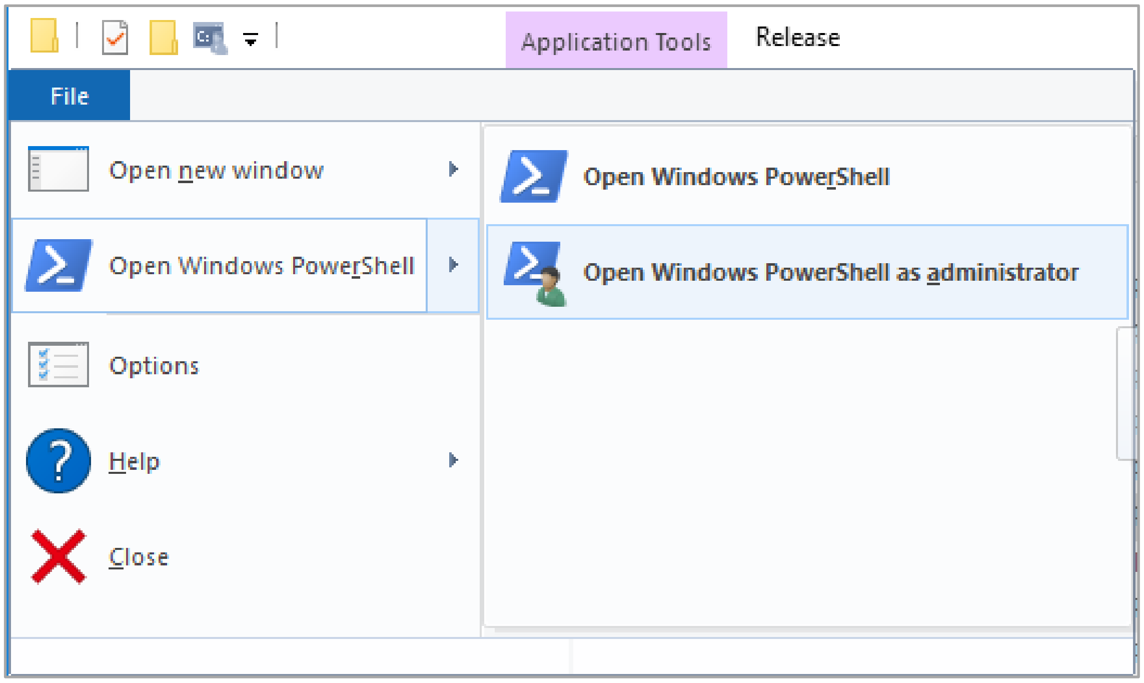
Save Frame Data Now

- [ ] Allow asynchronous reprojection
- [✓] Allow interleaved reprojection

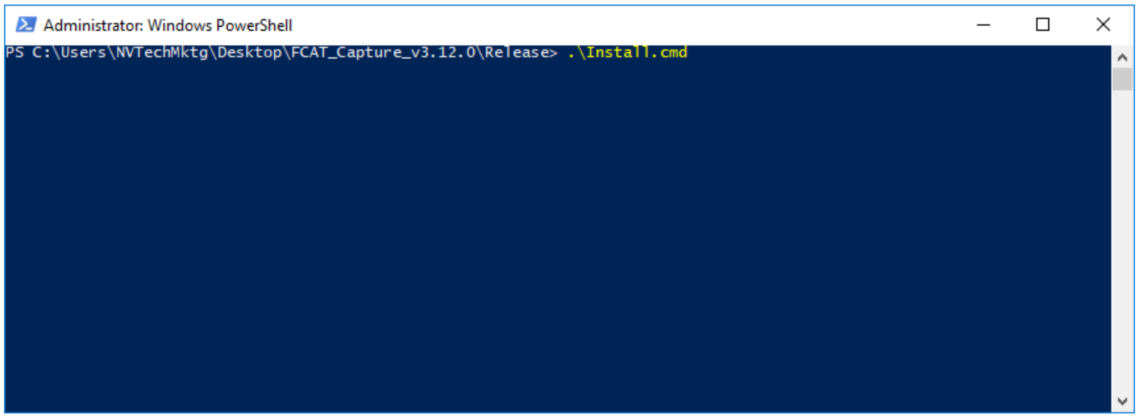
Display Frame Timing

Dashboard Off
Chaperone Bounds On
Async Reprojection Off
Compositor
Framerate Good

5. Close **SteamVR**.
6. **Open a Powershell window with administrator privileges** in the directory of the supplied FCAT VR Capture files.



7. From an administrator Powershell, execute .**Install.cmd**.



```
PS C:\Users\NVTechMktg\Desktop\FCAT_Capture_v3.12.0\Release> .\Install.cmd
```

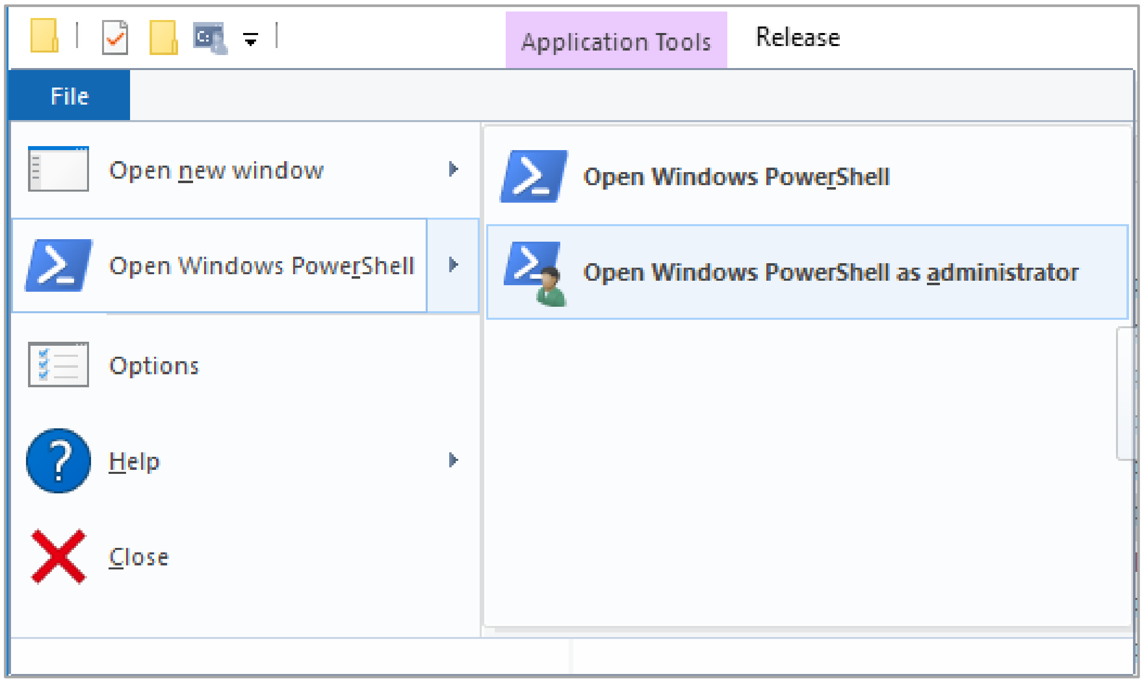
8. Reboot your system.

**NOTE:** You **must** repeat steps 4 – 6 each time you install a new GPU or new graphics driver.

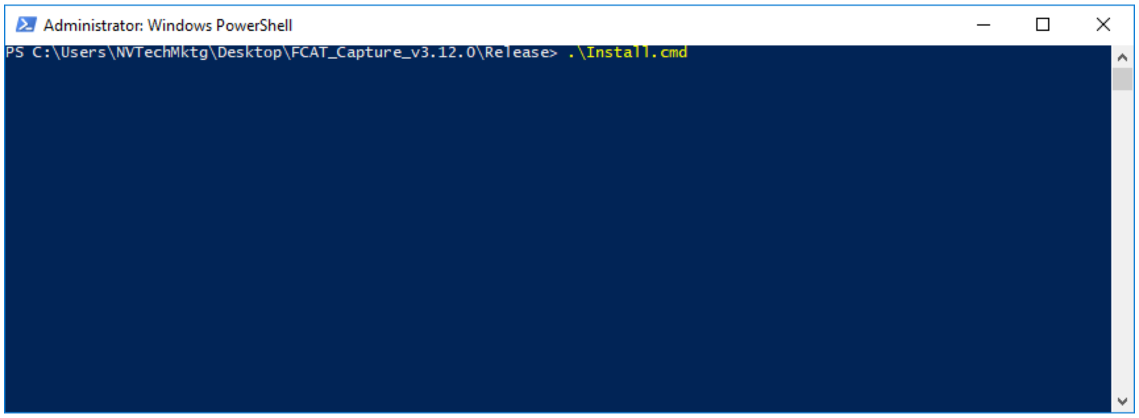
## Installing on Oculus Rift

Please follow these instructions when using FCAT VR Capture with the Oculus Rift.

1. Ensure that all graphic drivers are properly installed.
2. **Open a PowerShell window with administrator privileges** in the directory of the supplied FCAT VR Capture files.



3. From an administrator Powershell, execute .**Install.cmd**.



```
PS C:\Users\NVTechMktg\Desktop\FCAT_Capture_v3.12.0\Release> .\Install.cmd
```

4. Reboot your system

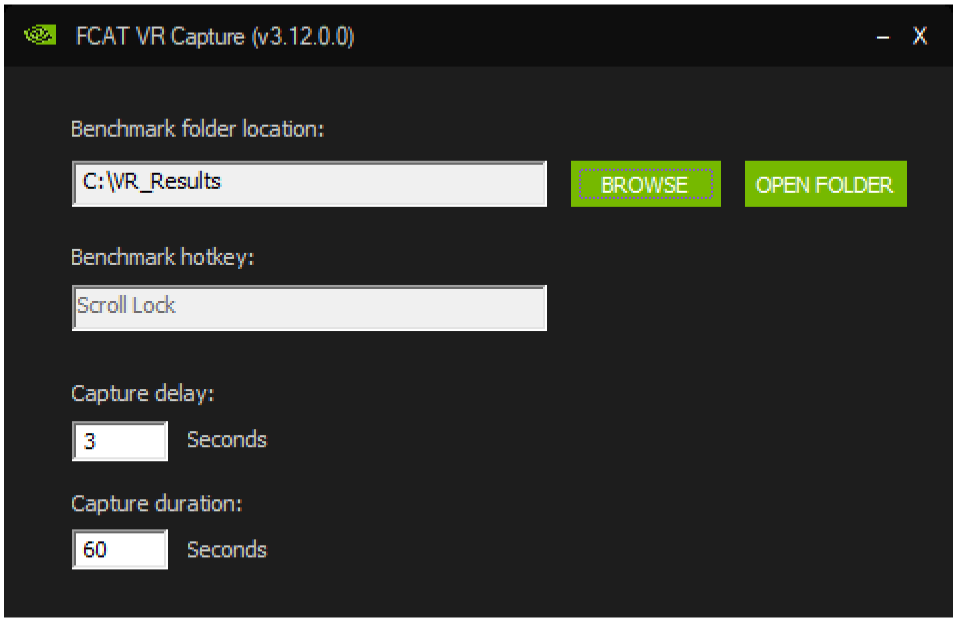
**NOTE:** You **must** repeat steps 2 – 4 each time you install a new GPU or new graphics driver.

## Using FCAT VR Capture

FCAT VR Capture works with all versions of Windows, and all DirectX APIs, all GPUs, and both Oculus Rift and HTC Vive. However, there is no OpenGL support at this time.

1. Right-click on **FCAT_Capture.exe** and then select **Run as administrator.**

**NOTE:** Please ensure that FRAPS is shut down before launching FCAT VR Capture. As an additional step, it's recommended to disable overlays from other applications to ensure they don't step on FCAT VR Capture.



Benchmark folder location: C:\VR_Results

Benchmark hotkey: Scroll Lock

Capture delay: 3 Seconds

Capture duration: 60 Seconds

2. In **Benchmark folder location**, select the **BROWSE** button to select the preferred directory where benchmark results will be stored.
3. Specify Capture delay and duration:

   **Capture delay –** Capture will start after 'delay' seconds. Setting this value to 0 starts the capture immediately.

   **Capture duration –** After 'duration' seconds are elapsed, the capture will stop automatically. Setting this value to 0 disables capture.

4. **Start a VR application.** A red bar will appear along the right side of the HMD to indicate that FCAT VR Capture is currently running.

   **Indicator color legend:**
   **Green** = capture in progress
   **Flashing green and red** = delayed start
   **Red** = capture is stopped

5. Press **SCROLL LOCK** to begin benchmarking. The red bar will turn green to indicate benchmarking is in progress.

**NOTE:** At this time, FCAT VR Capture only supports **SCROLL LOCK** as the benchmarking hotkey. At any point in time, if a capture is scheduled or running (indicator flashing or green), pressing SCROLL LOCK will stop the capture. To start the capture again, press SCROLL LOCK.

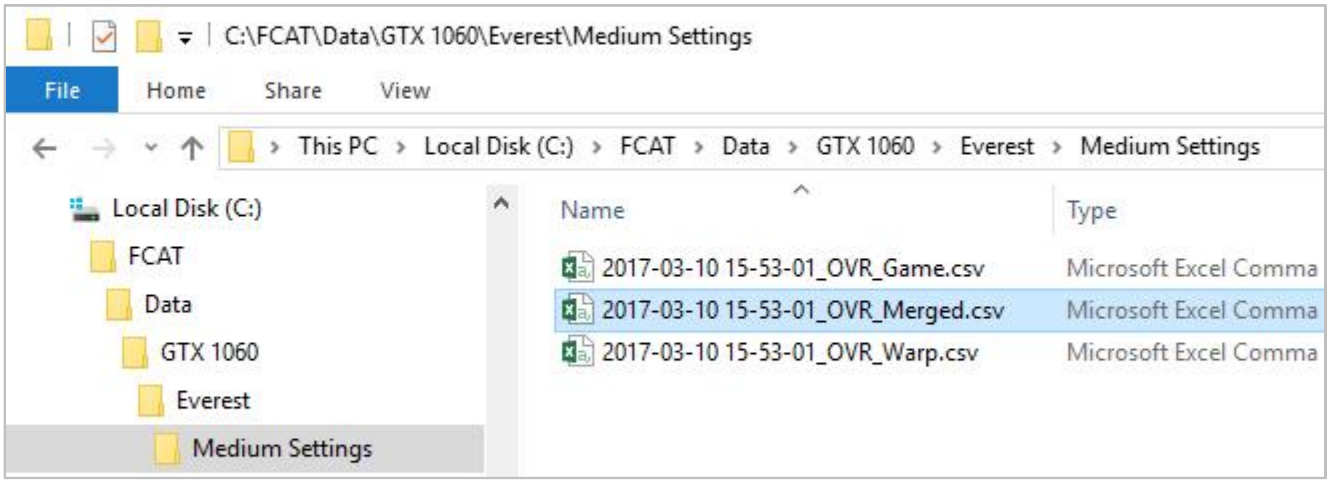
6. Press **SCROLL LOCK** again to stop benchmarking.
7. Exit the VR application and return to FCAT VR Capture. Click **OPEN FOLDER** to view benchmark results.
8. Results generated by FCAT VR Capture will be saved as a directory with a timestamp name. Consider renaming the directory to reflect the GPU, game, and settings tested.

**NOTE:** We STRONGLY recommend using the directory structure outlined in the next section for your data.

9. Within the results directory, the file which includes the word "merged" in the filename needs to be used to generate data with the new FCAT VR Analyzer.

## FCAT VR Data Folder

FCAT VR Capture creates three files. Always used the file with "Merged" in the name in the Analyzer.


C:\FCAT\Data\GTX 1060\Everest\Medium Settings
File
Home
Share
View
This PC › Local Disk (C:) › FCAT › Data › GTX 1060 › Everest › Medium Settings
Local Disk (C:)
FCAT
Data
GTX 1060
Everest
Medium Settings

| Name | Type |
|---|---|
| 2017-03-10 15-53-01_OVR_Game.csv | Microsoft Excel Comma |
| 2017-03-10 15-53-01_OVR_Merged.csv | Microsoft Excel Comma |
| 2017-03-10 15-53-01_OVR_Warp.csv | Microsoft Excel Comma |

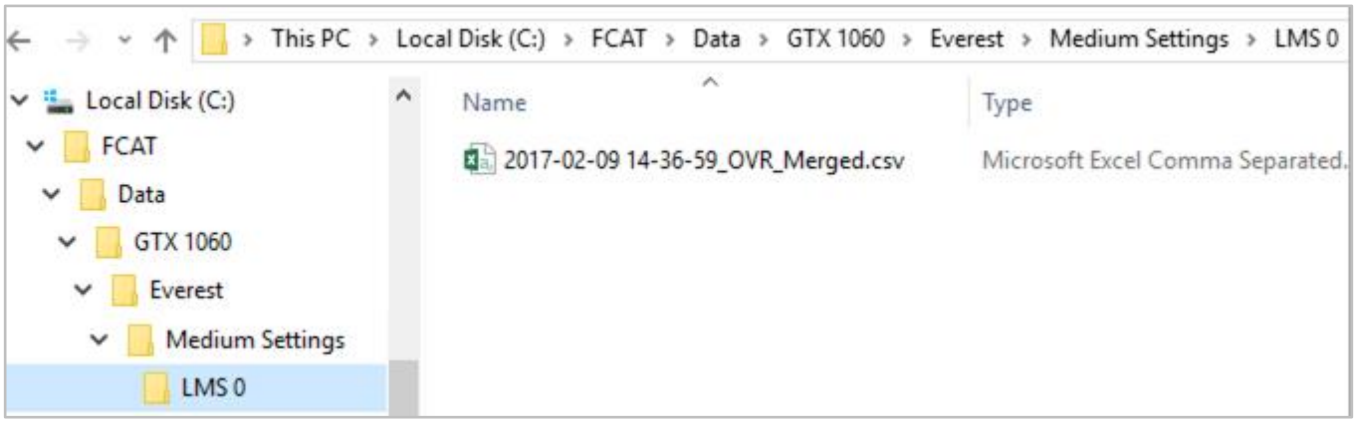
It is recommended that the data captured using FCAT VR Capture are placed in directory structure shown below.

Use the **C:\FCAT** directory as your base directory, and then create a **\Data** folder to place your captured VR data into.

Use the folder naming method for your captured VR data as follows:

C:\FCAT\DATA**\<GPU>\<GAME>\<SETTINGS>\<OTHER>**

**NOTE:** The folders used for <OTHER> in this example are for LMS and MRS captures.


This PC > Local Disk (C:) > FCAT > Data > GTX 1060 > Everest > Medium Settings > LMS 0
Local Disk (C:)
FCAT
Data
GTX 1060
Everest
Medium Settings
LMS 0
Name
Type

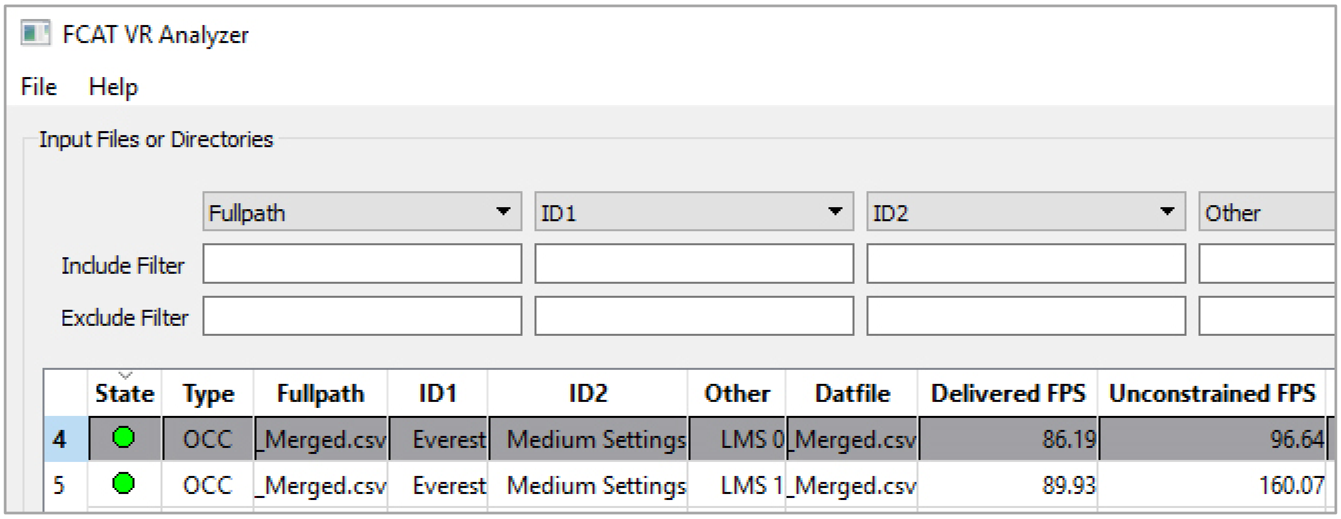
2017-02-09 14-36-59_OVR_Merged.csv — Microsoft Excel Comma Separated...

This is the folder structure using the example above:

C:\FCAT\DATA\**GTX 1060\Everest\Medium Settings\LMS 0**

**NOTE:** You can use dashes, underlines, or spaces in the folder names.

Using this folder structure will more easily allow the FCAT VR Analyzer software to capture the GPU, Game, Settings, and other info that you would like included in your charts.



| | Fullpath | ID1 | ID2 | Other |
|---|---|---|---|---|

| | State | Type | Fullpath | ID1 | ID2 | Other | Datfile | Delivered FPS | Unconstrained FPS |
|---|---|---|---|---|---|---|---|---|---|
| 4 | ● | OCC | _Merged.csv | Everest | Medium Settings | LMS 0 | _Merged.csv | 86.19 | 96.64 |
| 5 | ● | OCC | _Merged.csv | Everest | Medium Settings | LMS 1 | _Merged.csv | 89.93 | 160.07 |

**Figure 1:** Using the folder structure above will make it easier for the FCAT VR Analyzer to properly fill in the elements based on your folder structure.

Smoothness is important for VR gaming, and users can evaluate stutter that might affect game fluidity in a few different ways. The most important method is simply to experience VR and evaluate how it feels. Does it hitch? Is there stutter? Pan the view around and feel how smooth it is in motion.

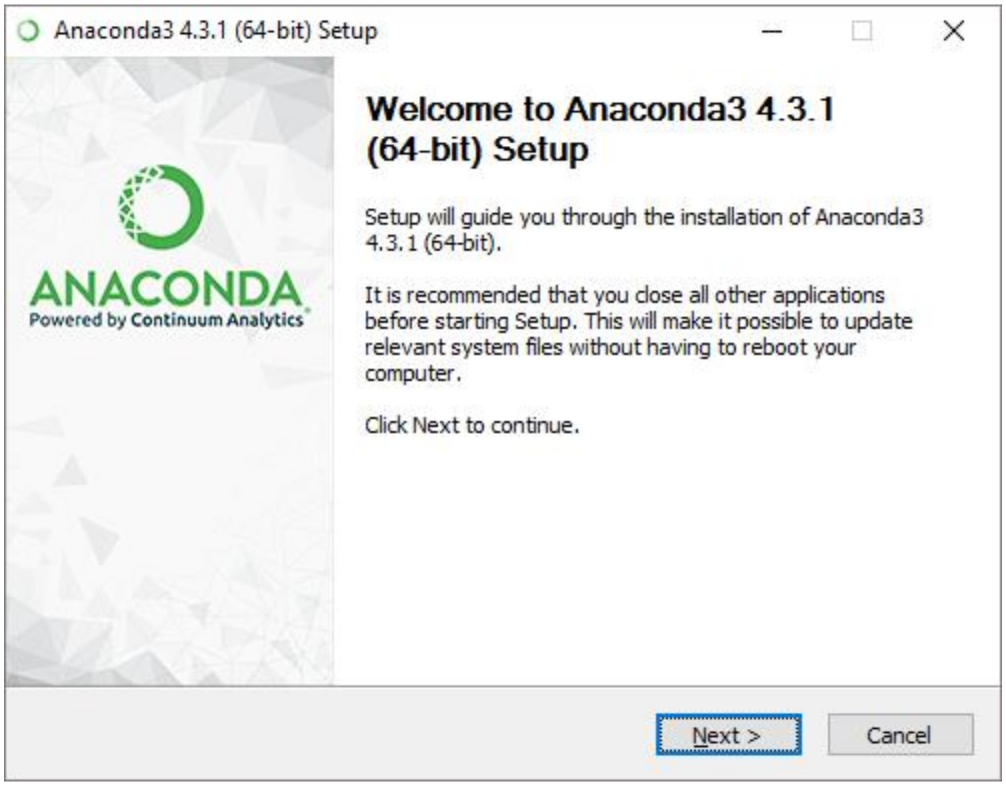
FCAT VR works on FRAPS and the new FCAT VR Capture frame time data. This section describes how to use FCAT VR.

## Installing Anaconda & pyqtgraph

FCAT VR is written with Python. As such, a few applications are needed.

### Anaconda

The first piece of independent software needed for FCAT VR to work is the Python software Anaconda. Download it here: https://www.continuum.io/downloads


Anaconda3 4.3.1 (64-bit) Setup

Welcome to Anaconda3 4.3.1 (64-bit) Setup

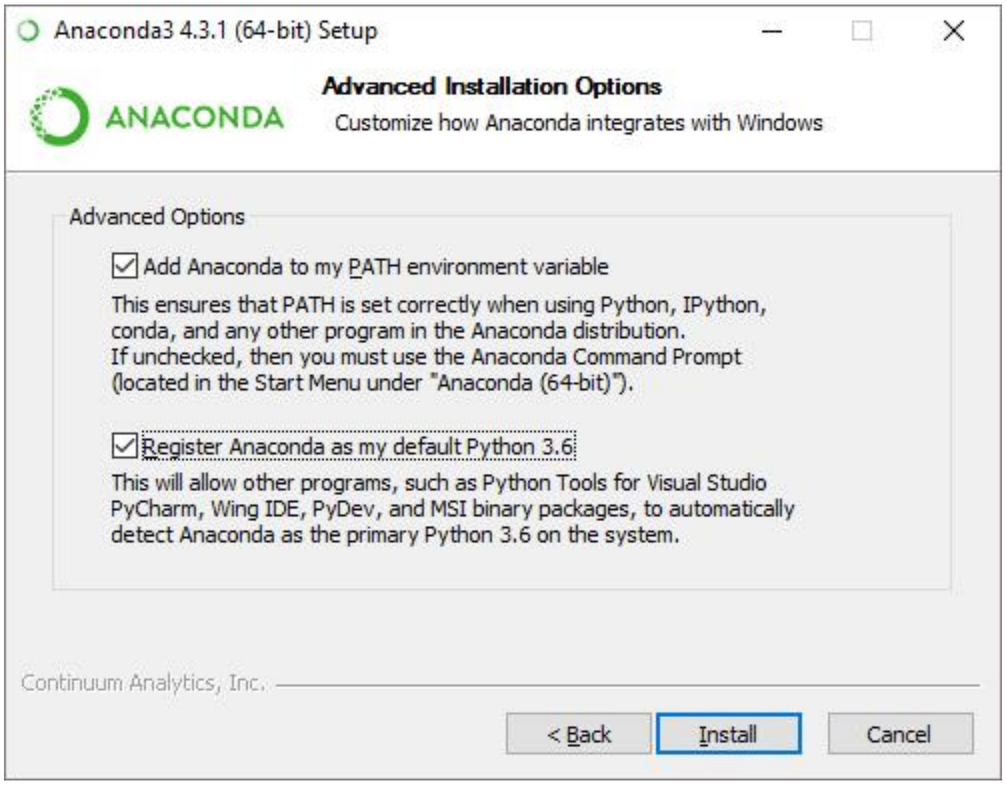
Setup will guide you through the installation of Anaconda3 4.3.1 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

**Figure 2:** Anaconda

During installation, make sure to select **Add Anaconda to the PATH environment variable** and also select **Register Anaconda as the default Python 3.6** (as shown below).



Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☑ Add Anaconda to my PATH environment variable

This ensures that PATH is set correctly when using Python, IPython, conda, and any other program in the Anaconda distribution. If unchecked, then you must use the Anaconda Command Prompt (located in the Start Menu under "Anaconda (64-bit)").

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Continuum Analytics, Inc.

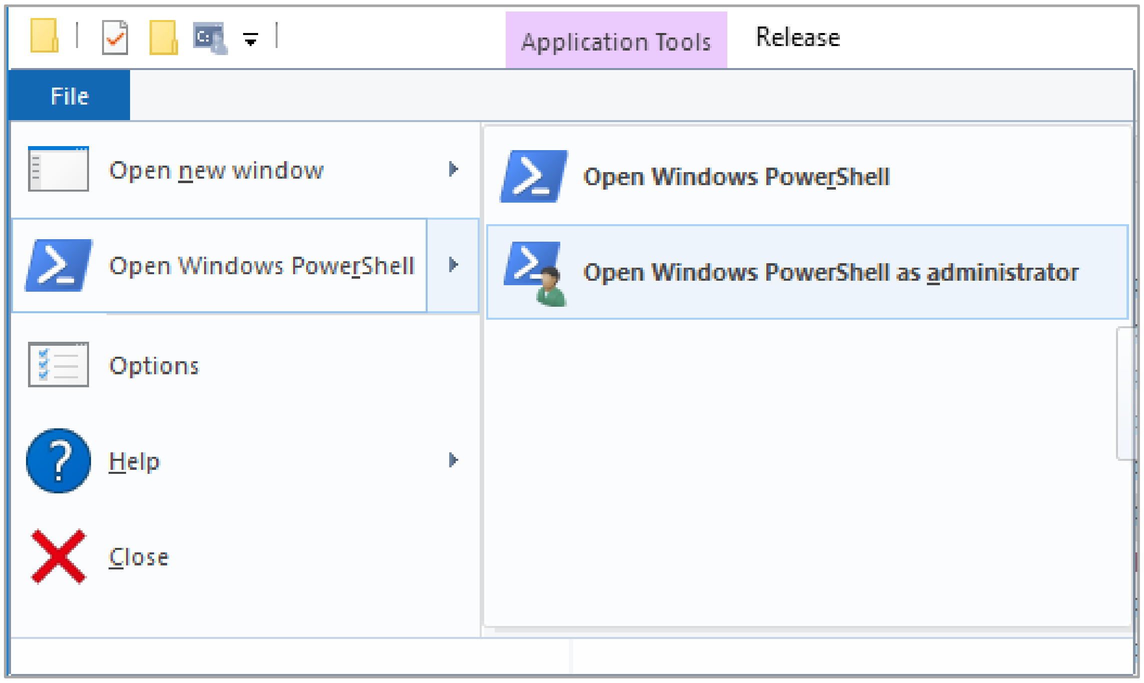
## pyqtgraph

The second piece of independent software used by the scripts to generate charts and plots is called pyqtgraph.

To install pyqtgraph:

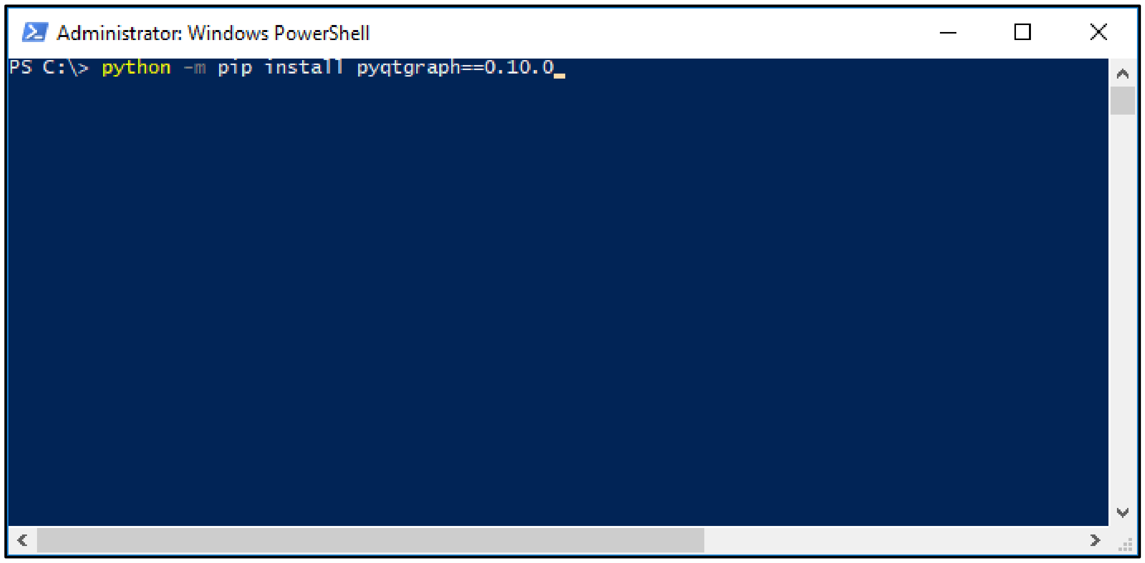
1. After installing Anaconda, **open a Powershell window with administrator privileges** from any location.

2.



3. From an administrator Powershell prompt, type

   **python -m pip install pyqtgraph==0.10.0**

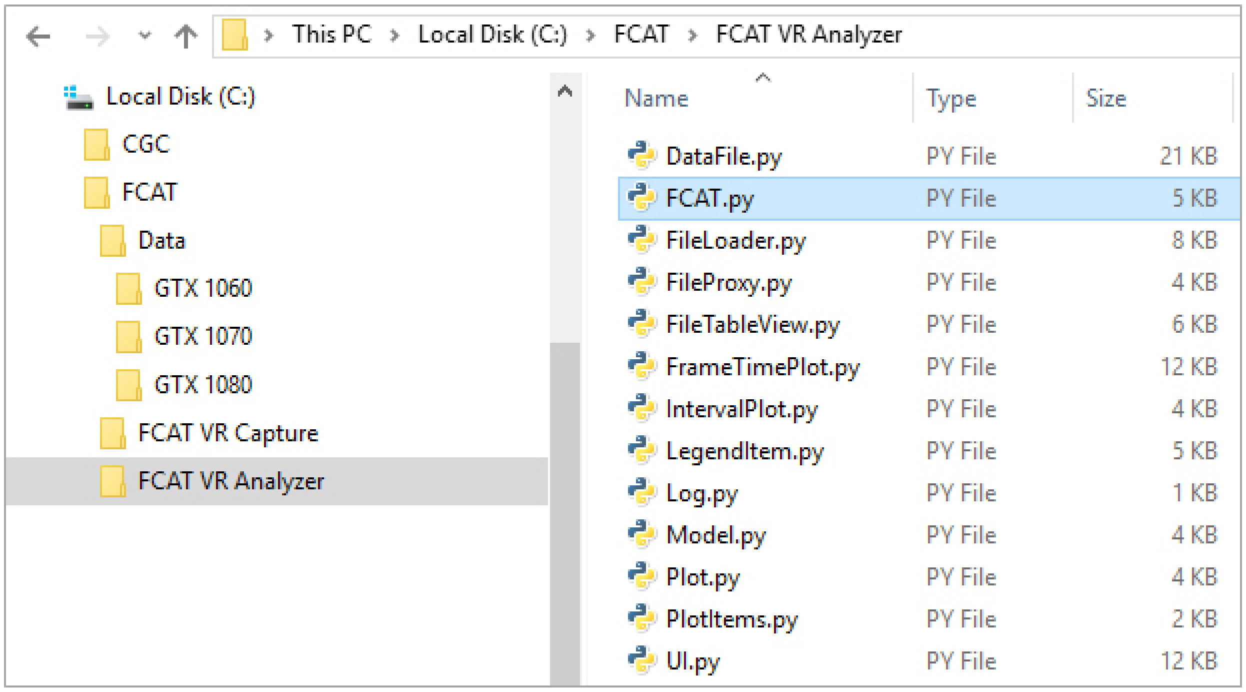


```
PS C:\> python -m pip install pyqtgraph==0.10.0_
```

**NOTE:** You may need to reboot if Python does not work after installation.

## Running FCAT VR Analyzer

After installing the software above, you should be able to run the FCAT VR Analyzer by double-clicking (opening) FCAT.py as shown below.



This PC > Local Disk (C:) > FCAT > FCAT VR Analyzer

Local Disk (C:)
- CGC
- FCAT
  - Data
    - GTX 1060
    - GTX 1070
    - GTX 1080
  - FCAT VR Capture
  - FCAT VR Analyzer

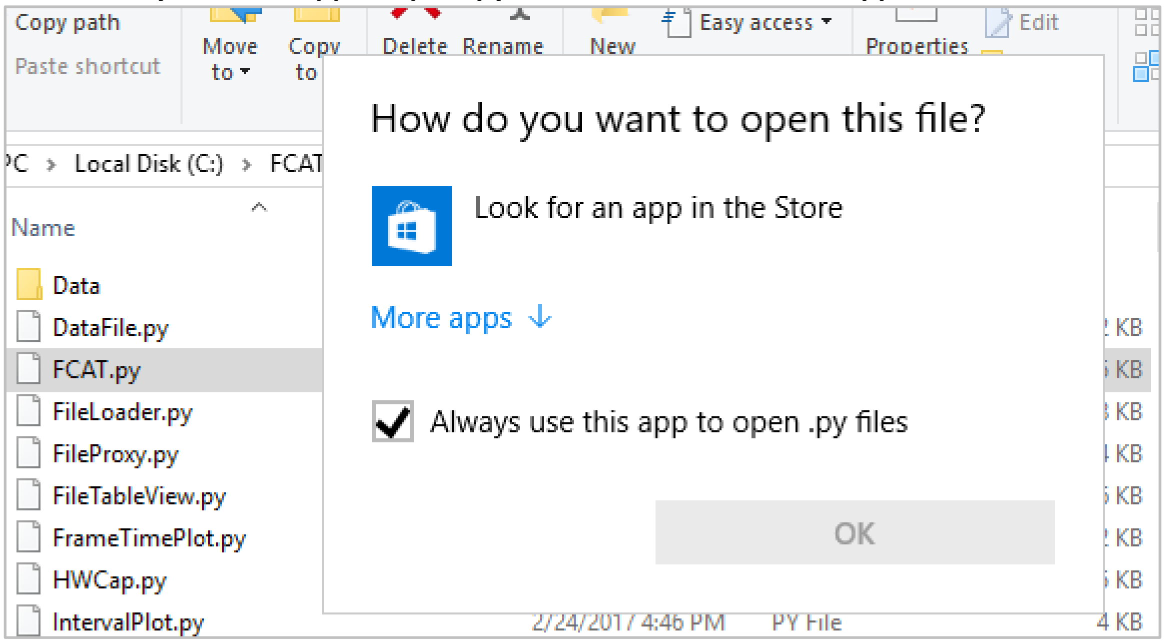
| Name | Type | Size |
|---|---|---|
| DataFile.py | PY File | 21 KB |
| FCAT.py | PY File | 5 KB |
| FileLoader.py | PY File | 8 KB |
| FileProxy.py | PY File | 4 KB |
| FileTableView.py | PY File | 6 KB |
| FrameTimePlot.py | PY File | 12 KB |
| IntervalPlot.py | PY File | 4 KB |
| LegendItem.py | PY File | 5 KB |
| Log.py | PY File | 1 KB |
| Model.py | PY File | 4 KB |
| Plot.py | PY File | 4 KB |
| PlotItems.py | PY File | 2 KB |
| UI.py | PY File | 12 KB |

### File Association

If FCAT.py is not associated properly with Python, you will need to set up the associate manually. Follow these instructions for doing this:

1. **Right-click on FCAT.py** in Windows File Explorer and select **Open with…**

2. Select **Always use this app to open .py files** and then click **More apps**.



How do you want to open this file?

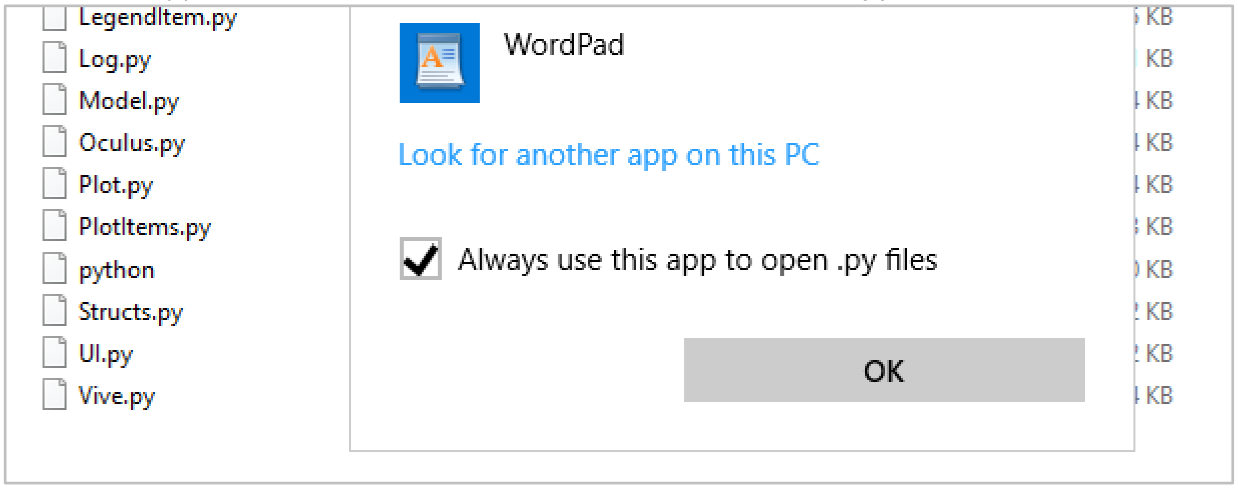
Look for an app in the Store

More apps ↓

✓ Always use this app to open .py files

| | | | |
|---|---|---|---|
| Data | | | |
| DataFile.py | | | KB |
| FCAT.py | | | KB |
| FileLoader.py | | | KB |
| FileProxy.py | | | KB |
| FileTableView.py | | | KB |
| FrameTimePlot.py | | | KB |
| HWCap.py | | | KB |
| IntervalPlot.py | 2/24/2017 4:46 PM | PY File | 4 KB |

3. A bunch of applications will be shown. Click on **Look for another app on this PC**.



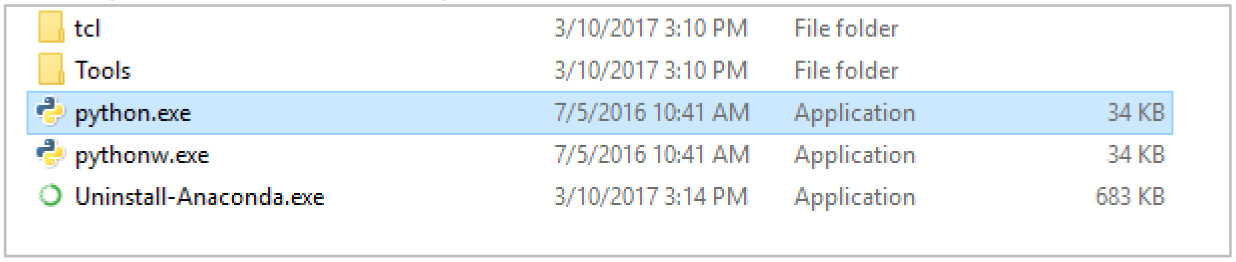
Look for another app on this PC

✓ Always use this app to open .py files

| | |
|---|---|
| LegendItem.py | KB |
| Log.py | KB |
| Model.py | KB |
| Oculus.py | KB |
| Plot.py | KB |
| PlotItems.py | KB |
| python | KB |
| Structs.py | KB |
| UI.py | KB |
| Vive.py | KB |

4. You will now need to browse to the location of Python.exe in the Anaconda3 folder.

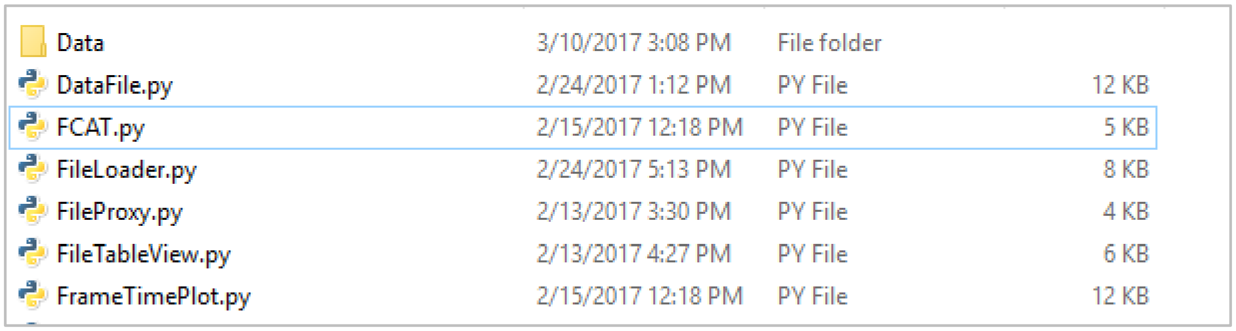
5. **Select Python.exe** and click the **Open** button.



| | | | |
|---|---|---|---|
| tcl | 3/10/2017 3:10 PM | File folder | |
| Tools | 3/10/2017 3:10 PM | File folder | |
| python.exe | 7/5/2016 10:41 AM | Application | 34 KB |
| pythonw.exe | 7/5/2016 10:41 AM | Application | 34 KB |
| Uninstall-Anaconda.exe | 3/10/2017 3:14 PM | Application | 683 KB |

6. FCAT.py should now be associated to Python.exe. All of the .py files will now have the Anaconda icon associated with them (as shown below).

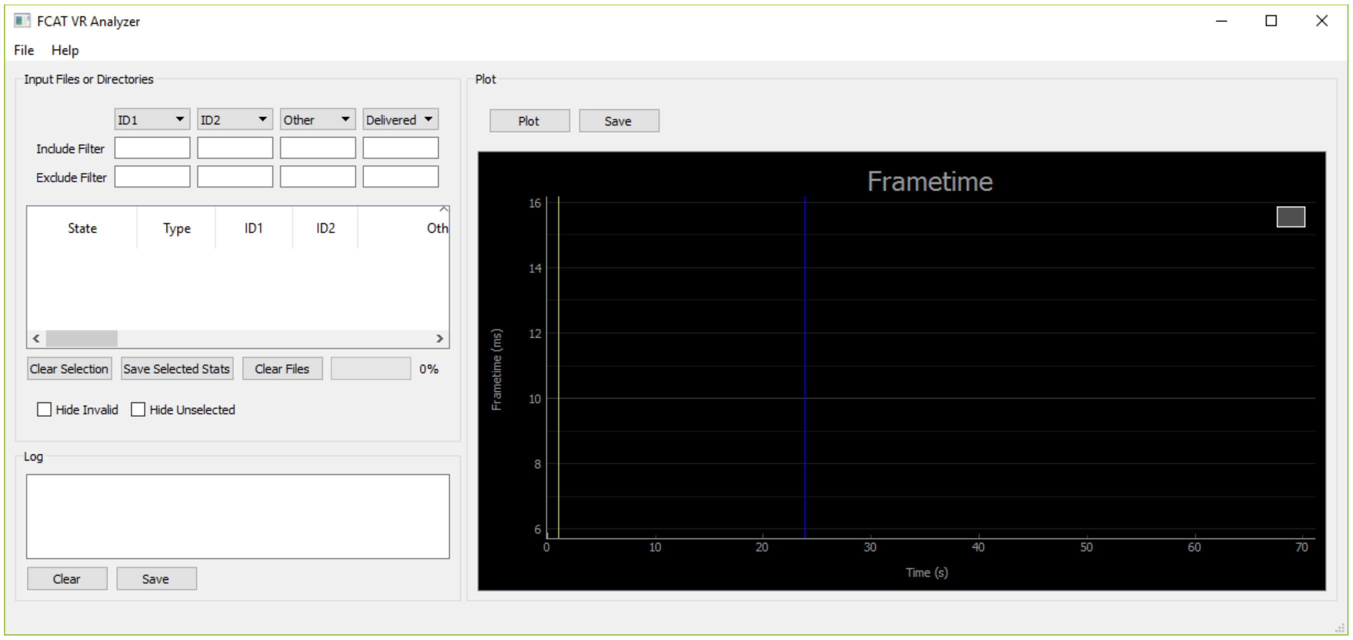
4.



| | | | |
|---|---|---|---|
| Data | 3/10/2017 3:08 PM | File folder | |
| DataFile.py | 2/24/2017 1:12 PM | PY File | 12 KB |
| FCAT.py | 2/15/2017 12:18 PM | PY File | 5 KB |
| FileLoader.py | 2/24/2017 5:13 PM | PY File | 8 KB |
| FileProxy.py | 2/13/2017 3:30 PM | PY File | 4 KB |
| FileTableView.py | 2/13/2017 4:27 PM | PY File | 6 KB |
| FrameTimePlot.py | 2/15/2017 12:18 PM | PY File | 12 KB |

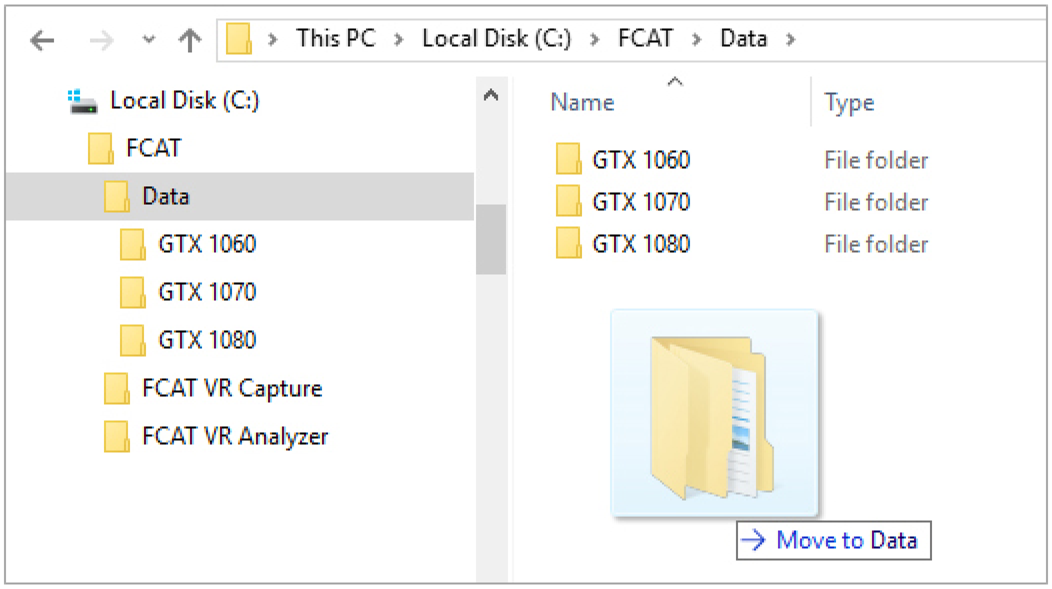
## Launching FCAT VR Analyzer

Once launched, FCAT VR Analyzer looks like this:



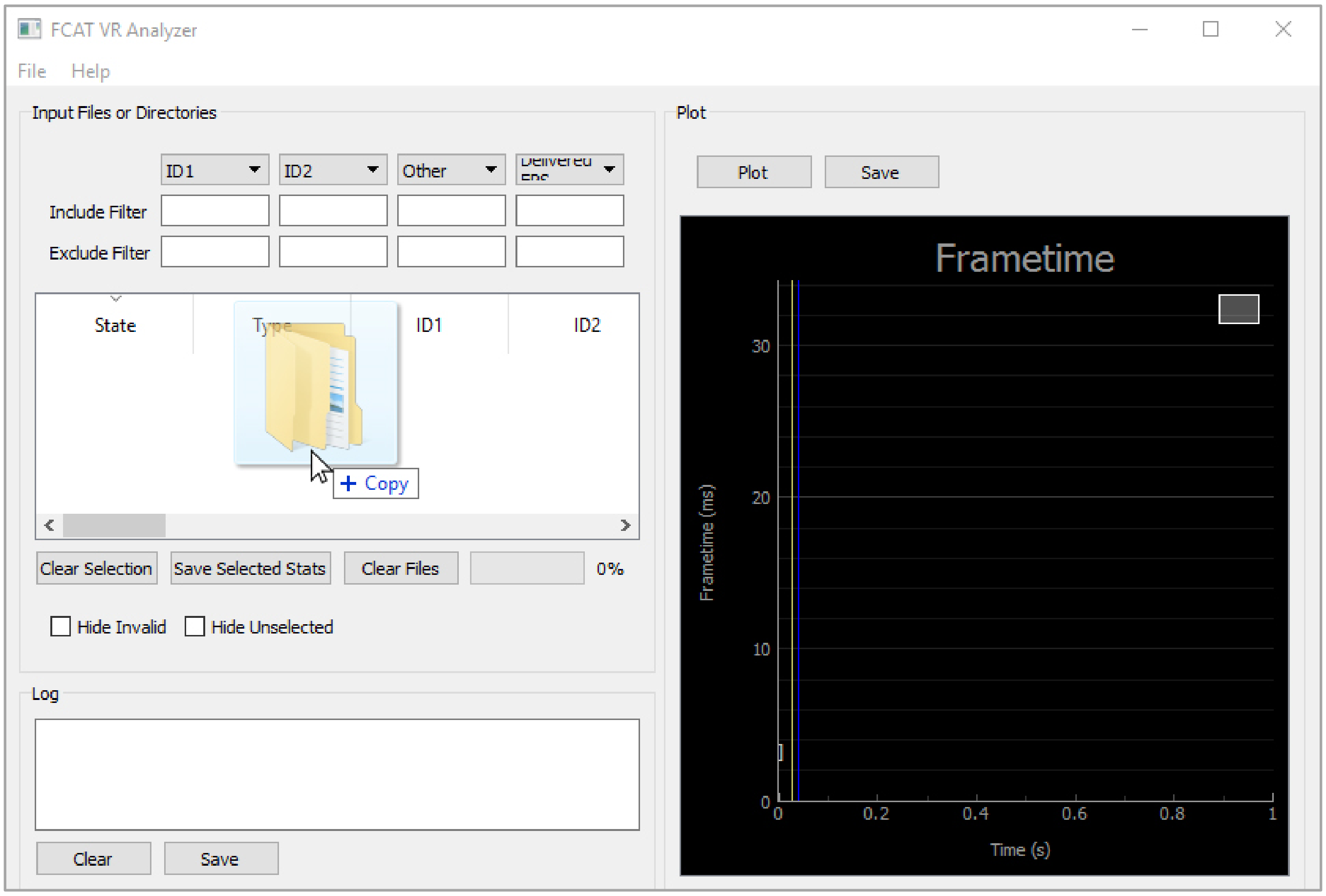
To get data in, select the \data folder from Windows Explorer and drag it with the mouse into the FCAT Analyzer program. Always drag from the \data folder. This is the folder that contains the subfolders as outlined above. You can see in the example below:

C:\FCAT\**DATA**\<GPU>\<GAME>\<SETTINGS>\<OTHER>

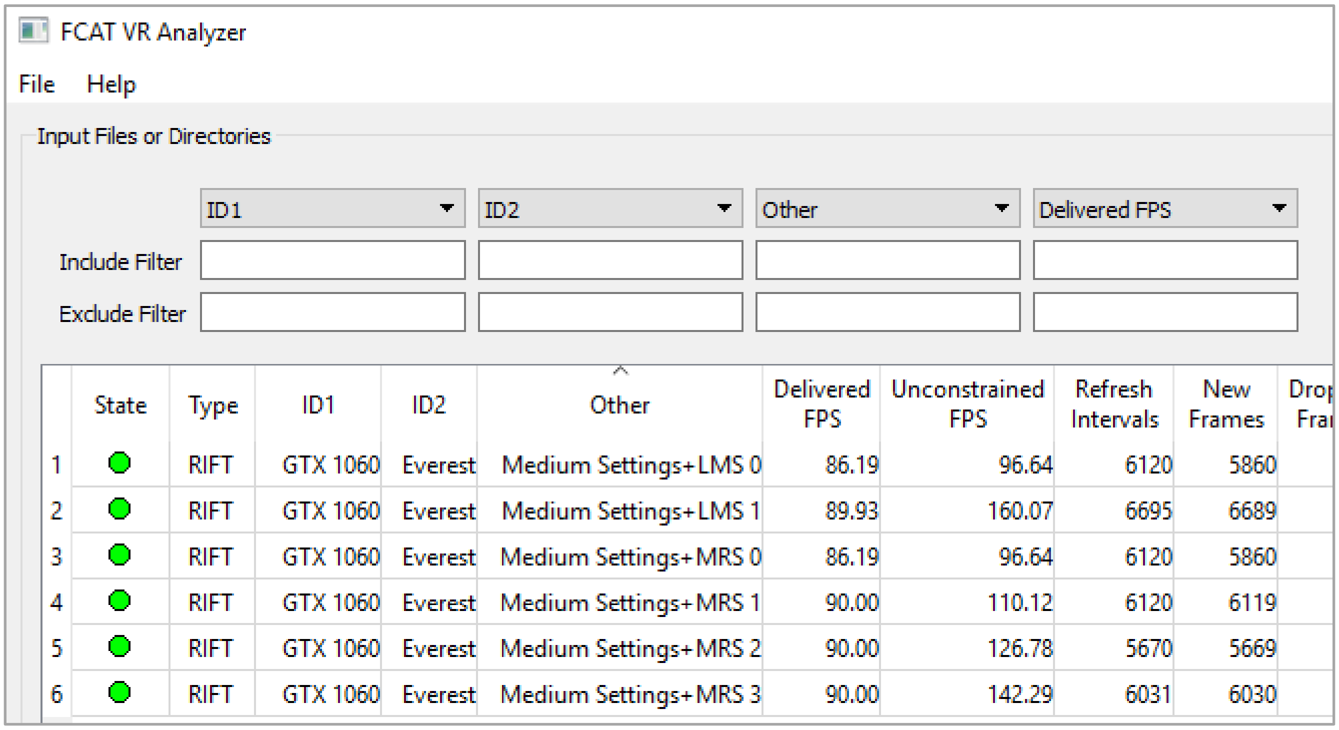


**NOTE:** You can drag multiple GPU folders into FCAT VR Analyzer. In fact, there is no limit.

Drag the folder(s) into the top-left region as shown below:



Your data should look like this:



| | ID1 | ID2 | Other | Delivered FPS |
|---|---|---|---|---|
| Include Filter | | | | |
| Exclude Filter | | | | |

| | State | Type | ID1 | ID2 | Other | Delivered FPS | Unconstrained FPS | Refresh Intervals | New Frames | Drop Fra |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ● | RIFT | GTX 1060 | Everest | Medium Settings+LMS 0 | 86.19 | 96.64 | 6120 | 5860 | |
| 2 | ● | RIFT | GTX 1060 | Everest | Medium Settings+LMS 1 | 89.93 | 160.07 | 6695 | 6689 | |
| 3 | ● | RIFT | GTX 1060 | Everest | Medium Settings+MRS 0 | 86.19 | 96.64 | 6120 | 5860 | |
| 4 | ● | RIFT | GTX 1060 | Everest | Medium Settings+MRS 1 | 90.00 | 110.12 | 6120 | 6119 | |
| 5 | ● | RIFT | GTX 1060 | Everest | Medium Settings+MRS 2 | 90.00 | 126.78 | 5670 | 5669 | |
| 6 | ● | RIFT | GTX 1060 | Everest | Medium Settings+MRS 3 | 90.00 | 142.29 | 6031 | 6030 | |

## App Regions & Elements

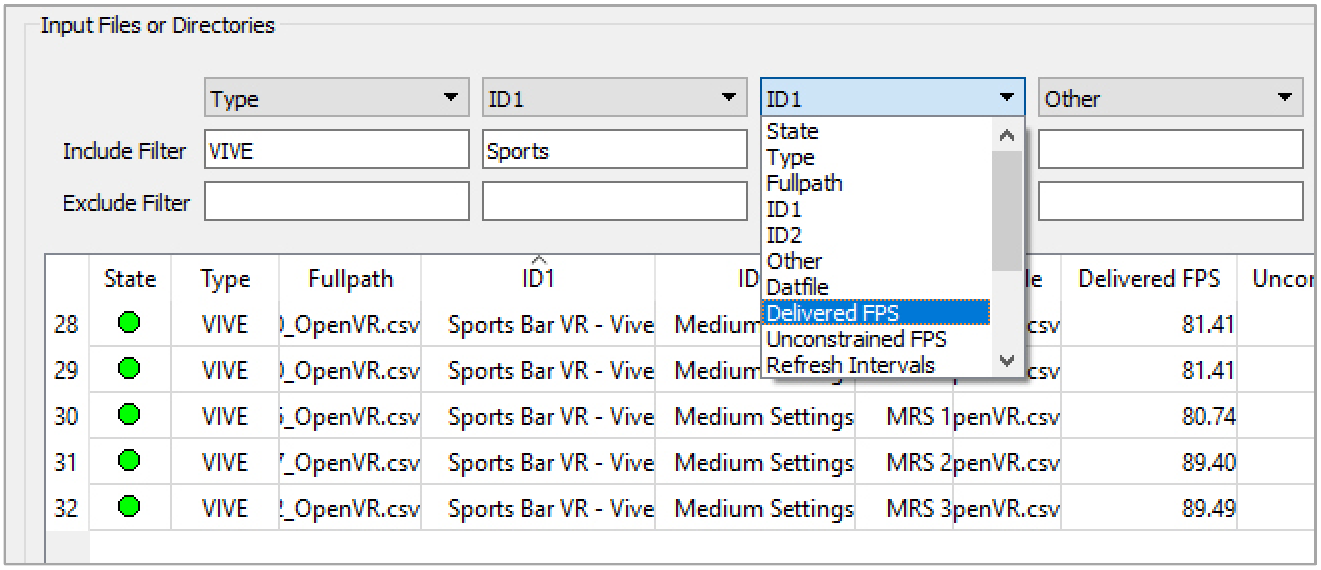
The FCAT VR Analyzer application contains three regions, and each of these regions can be resized by using the mouse as needed.

### Input Files or Directories Region

The main section allows for the organization and manipulation of captured VR data. This is where you can filter, sort, and rename data.

#### Filters

Using filters is a great method to sort through vast amounts of captured data. There are filters to include and exclude data from any of the data columns. The drop-down menus allow for up to four columns of data to be filtered. These column headers can be changed by selecting the drop down menu and selecting another column header to be filtered (as shown below).



Input Files or Directories

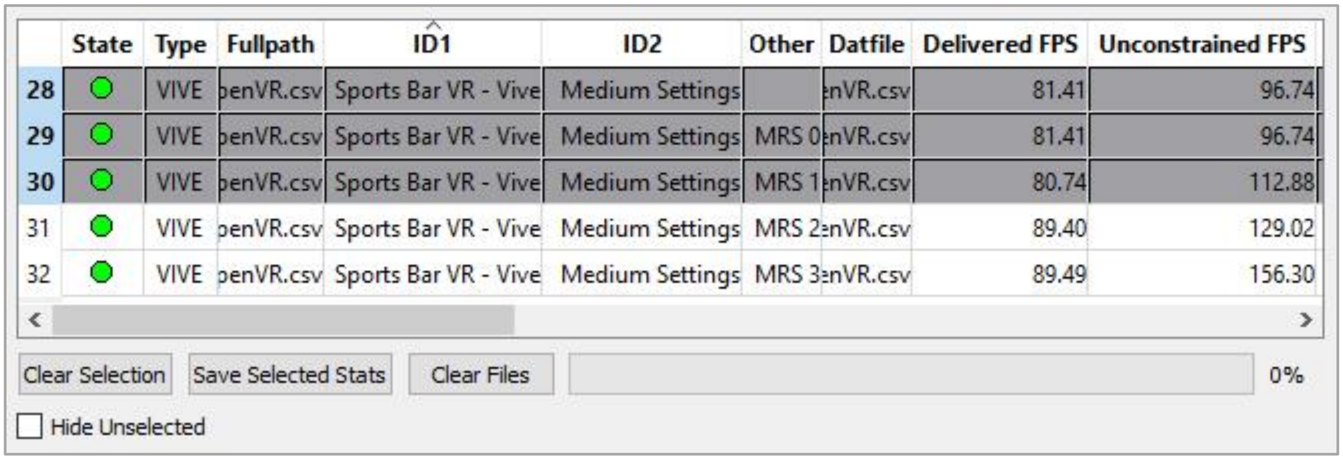
| | Type | ID1 | ID1 | Other |
|---|---|---|---|---|
| Include Filter | VIVE | Sports | | |
| Exclude Filter | | | | |

ID1 dropdown: State, Type, Fullpath, ID1, ID2, Other, Datfile, Delivered FPS, Unconstrained FPS, Refresh Intervals

| | State | Type | Fullpath | ID1 | ID | | le | Delivered FPS | Uncor |
|---|---|---|---|---|---|---|---|---|---|
| 28 | ● | VIVE | )_OpenVR.csv | Sports Bar VR - Vive | Medium | | csv | 81.41 | |
| 29 | ● | VIVE | )_OpenVR.csv | Sports Bar VR - Vive | Medium | | csv | 81.41 | |
| 30 | ● | VIVE | ;_OpenVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 1 | penVR.csv | 80.74 | |
| 31 | ● | VIVE | '_OpenVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 2 | penVR.csv | 89.40 | |
| 32 | ● | VIVE | !_OpenVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 3 | penVR.csv | 89.49 | |

## Data Region

Click on a data row to select it, and click it again to deselect it. Multiple rows can be selected by clicking and dragging the mouse down over the data, and all data can be selected by pressing Ctrl+A.

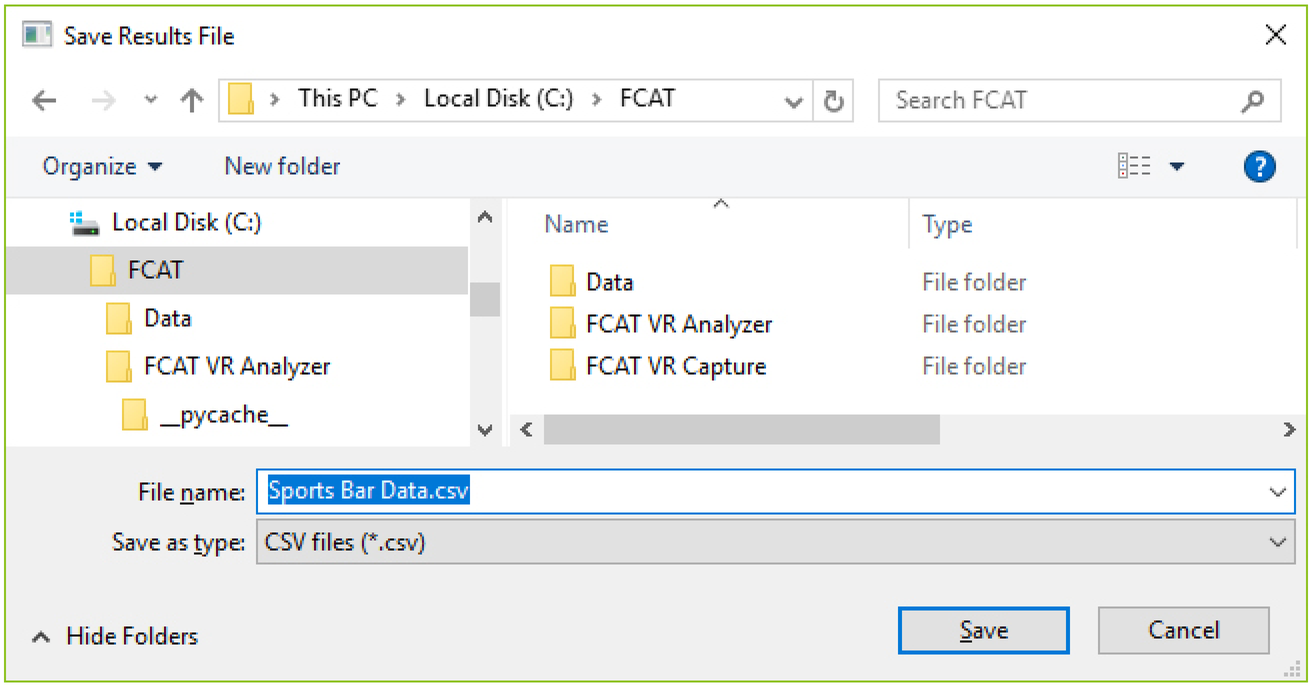


| | State | Type | Fullpath | ID1 | ID2 | Other | Datfile | Delivered FPS | Unconstrained FPS |
|---|---|---|---|---|---|---|---|---|---|
| 28 | ● | VIVE | penVR.csv | Sports Bar VR - Vive | Medium Settings | | nVR.csv | 81.41 | 96.74 |
| 29 | ● | VIVE | penVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 0 | nVR.csv | 81.41 | 96.74 |
| 30 | ● | VIVE | penVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 1 | nVR.csv | 80.74 | 112.88 |
| 31 | ● | VIVE | penVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 2 | nVR.csv | 89.40 | 129.02 |
| 32 | ● | VIVE | penVR.csv | Sports Bar VR - Vive | Medium Settings | MRS 3 | nVR.csv | 89.49 | 156.30 |

Clicking **Clear Selection** will clear any of the data that has already been selected, and clicking **Clear Files** will remove all data from the data window. Clicking **Hide Unselected** will hide data that is currently not selected.

### Save Selected Stats

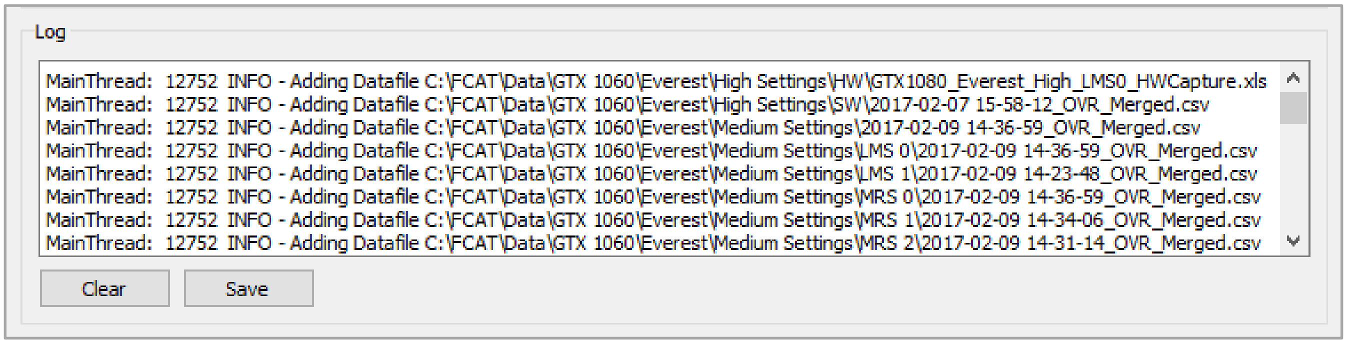
Clicking **Save Selected Stats** will open a window where the selected data can be saved as a .CSV file (see below).



File name: Sports Bar Data.csv
Save as type: CSV files (*.csv)

## Log Region

The logging window provides information in case of error. This data can be saved and sent to NVIDIA to assist with troubleshooting.

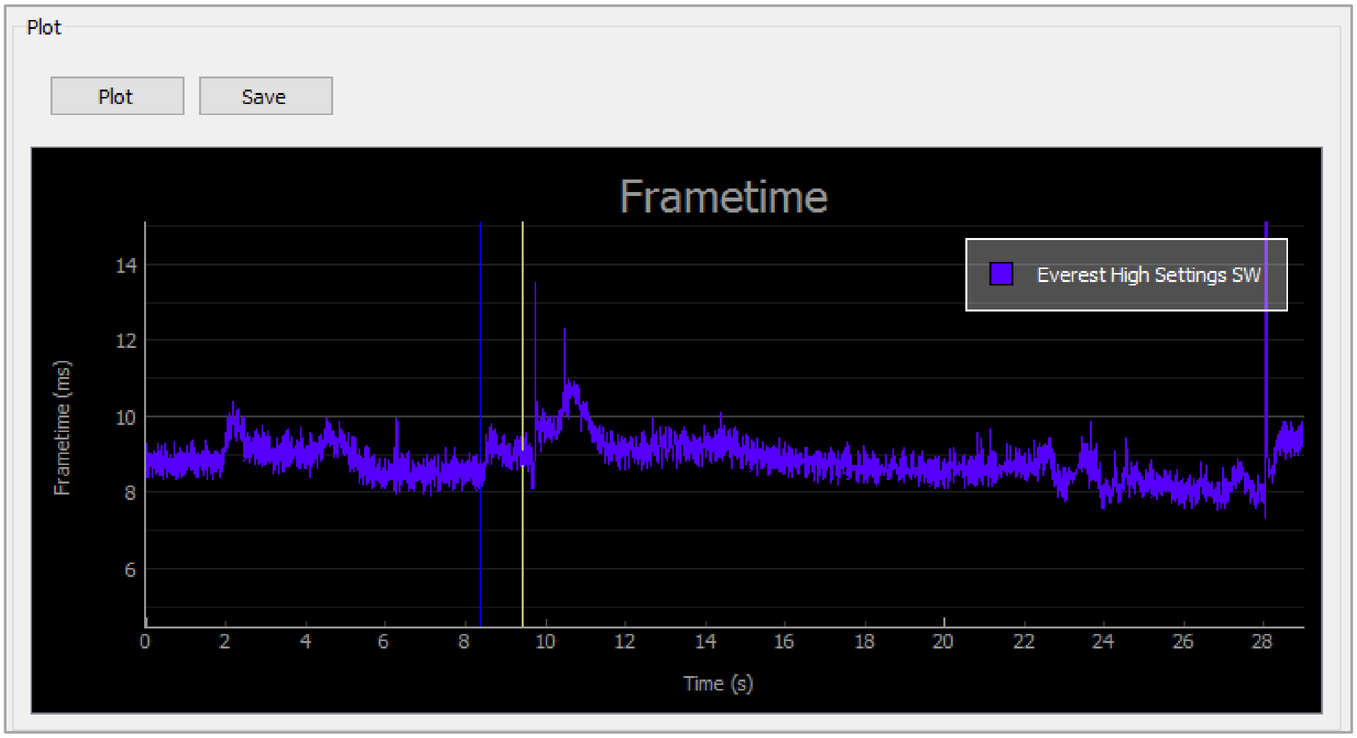


```
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\High Settings\HW\GTX1080_Everest_High_LMS0_HWCapture.xls
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\High Settings\SW\2017-02-07 15-58-12_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\2017-02-09 14-36-59_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\LMS 0\2017-02-09 14-36-59_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\LMS 1\2017-02-09 14-23-48_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\MRS 0\2017-02-09 14-36-59_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\MRS 1\2017-02-09 14-34-06_OVR_Merged.csv
MainThread:  12752  INFO - Adding Datafile C:\FCAT\Data\GTX 1060\Everest\Medium Settings\MRS 2\2017-02-09 14-31-14_OVR_Merged.csv
```

## Plot Region

Clicking the **Plot** button will create charts for all of the data that has been selected. Clicking the **Save** button will allow the plot to be saved to a .PNG file.

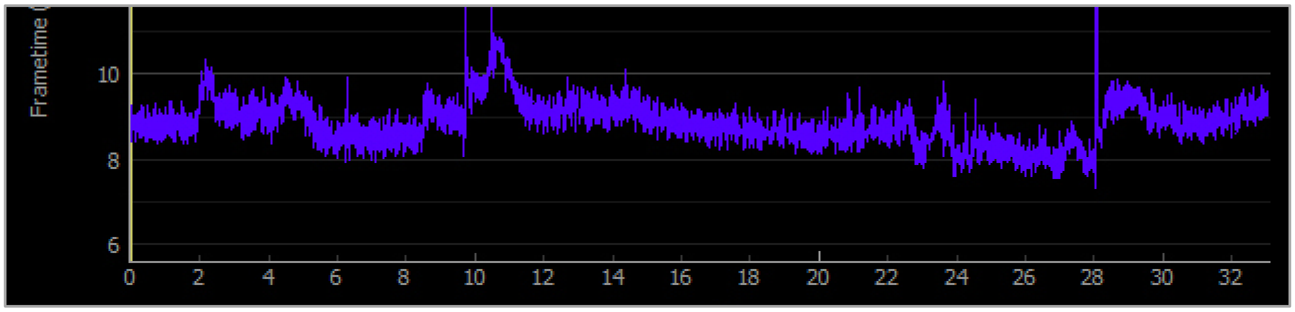
**NOTE:** Up to eight datasets are currently supported.



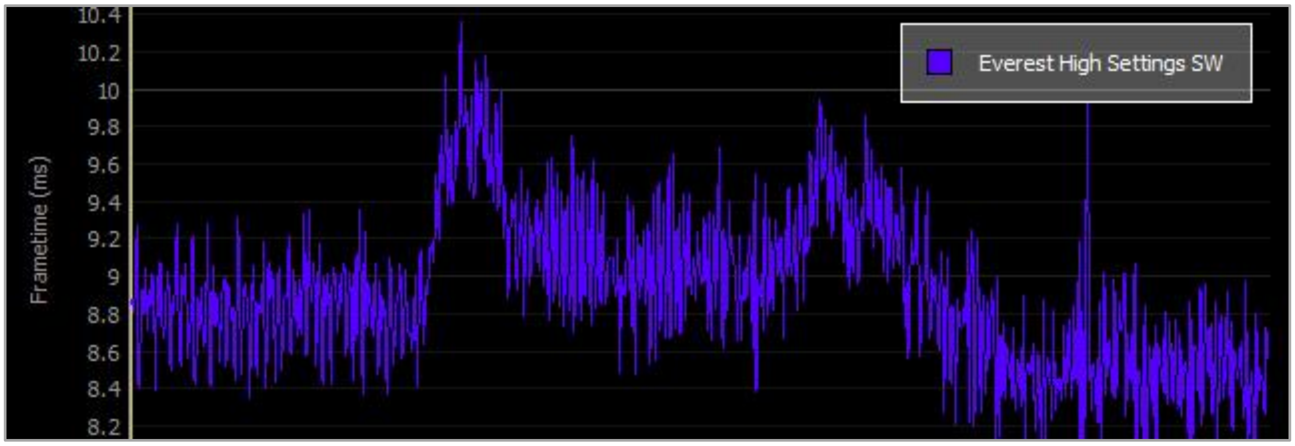
## Manipulating Charts

### Using the Mouse

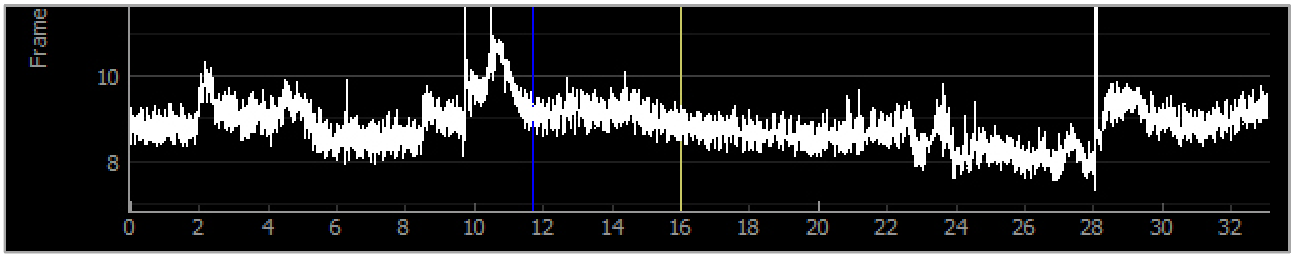
The plot lines can be easily moved around using the mouse. This allows the chart to be centered and zoomed as required.



Use the middle mouse wheel to zoom in and out:



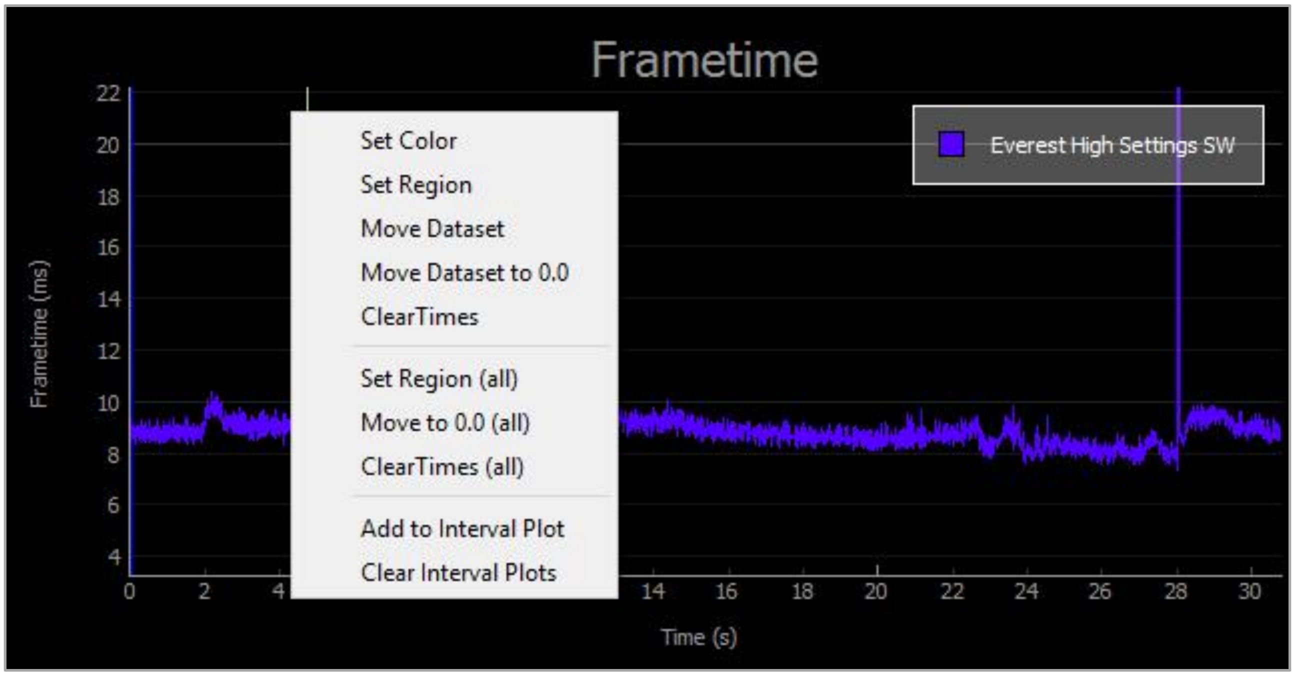
Selecting a line will turn it white to show that it has been selected:



**NOTE:** The blue vertical line is where the chart line was selected using the mouse, and the yellow vertical line will continuously follow the mouse as it's moved around the plot.

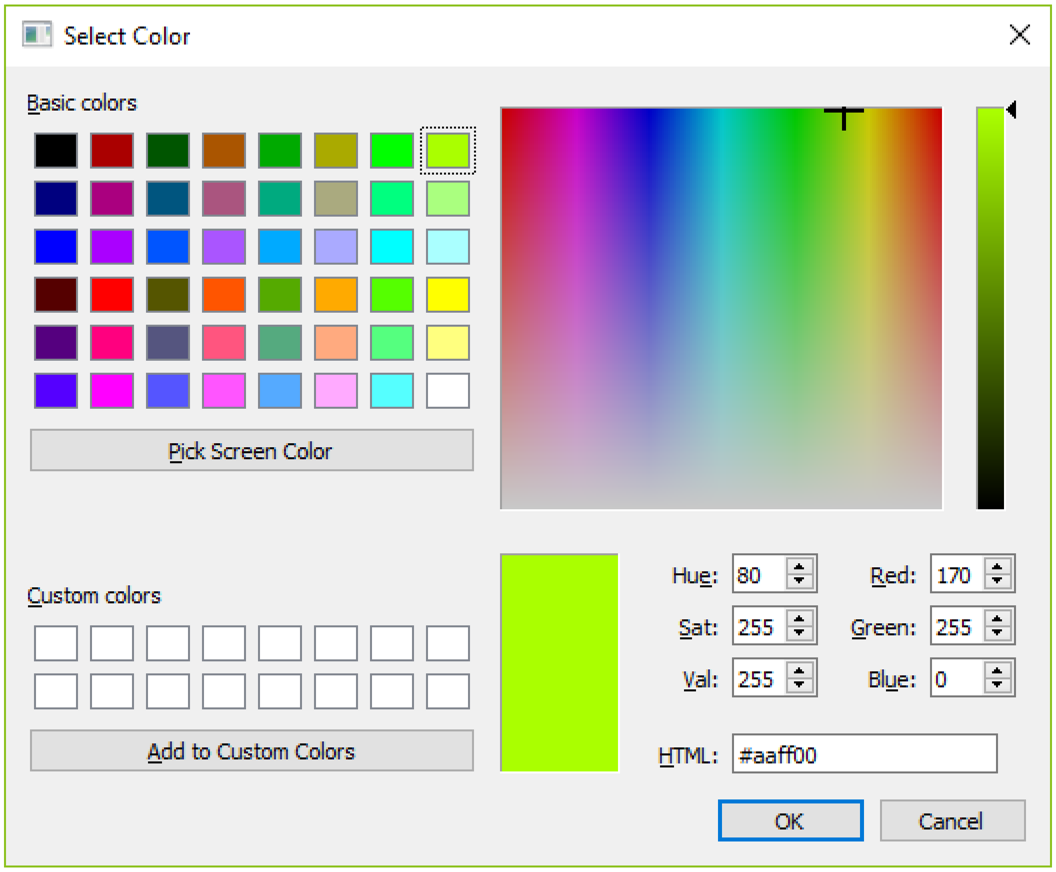
## Plot Menu

Right-clicking on the plot will expose the **Plot Menu**. This will allow you to manipulate the plot lines that you have selected.



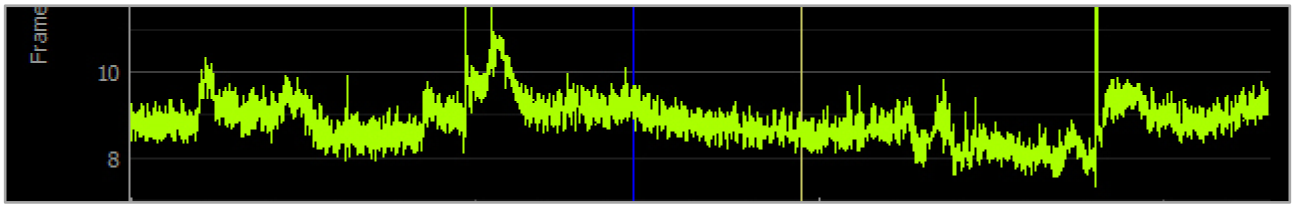
## Set Color

Chart line colors can be changed using this option. First, select the chart line by clicking on it (which turns it white), and then right-click on the plot and select **Set Color**.



Hue: 80
Sat: 255
Val: 255
Red: 170
Green: 255
Blue: 0
HTML: #aaff00

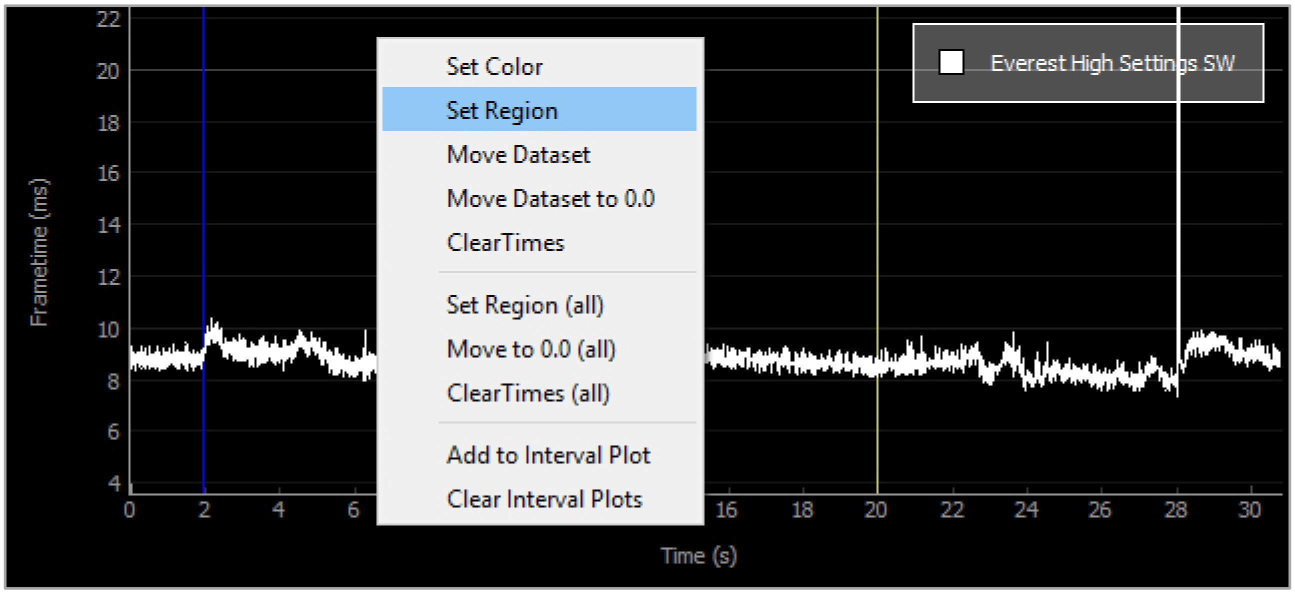
In this example, we selected green as our new line color.



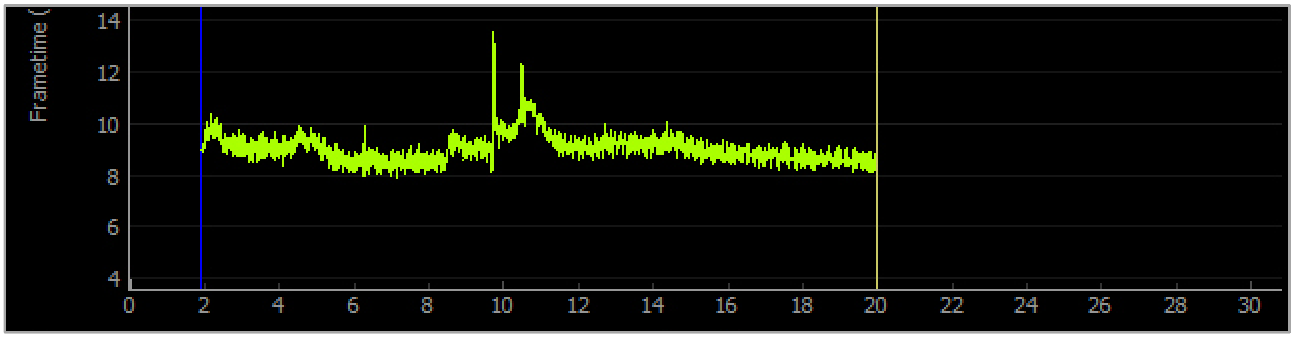
## Set Region

Use this option to trim the beginning and ends of the chart line. This is useful when you have data outside the benchmark area such as menus. This extra data not only looks wrong on the chart, but it can negatively affect the FPS data as well (and other data including a number of dropped and synthetic frames).

Use the blue and yellow lines to set the region, which allows for trimming of the start and end data from the chart. First, select the starting point by selecting the chart line (left-click) where you want the beginning data trimmed, then select the end point with a right-click.



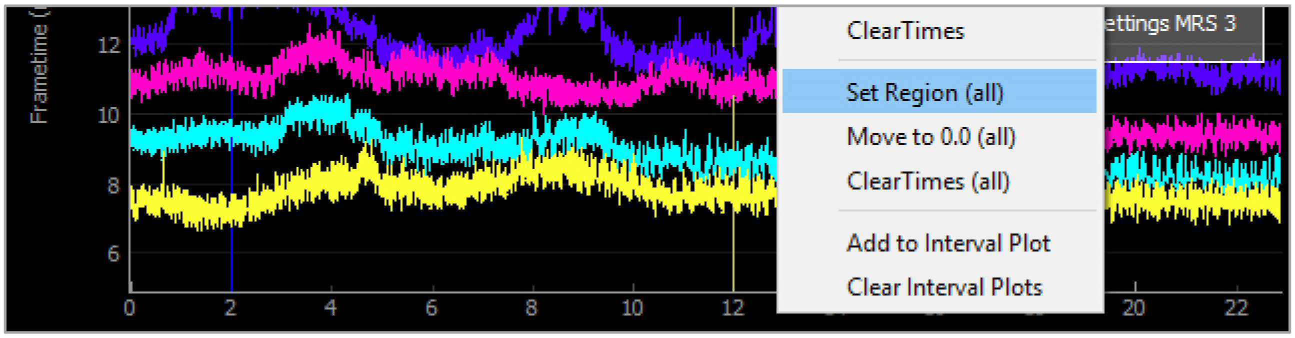
Select **Set Region** when the desired area is selected to trim your data.



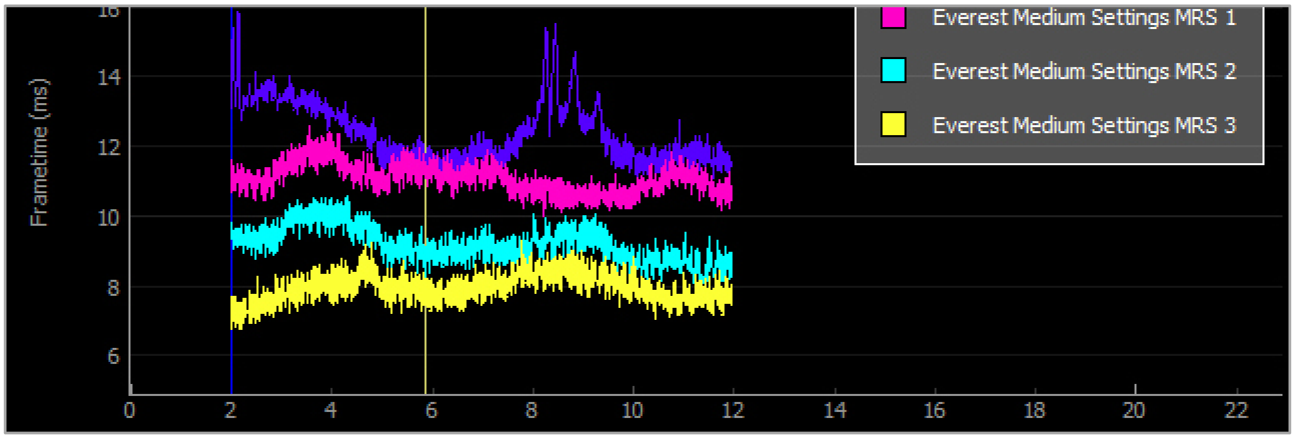
**NOTE:** The FPS and other data will be affected by the new region that you set.

## Set Region (all)

This allows multiple chart lines to be trimmed together.

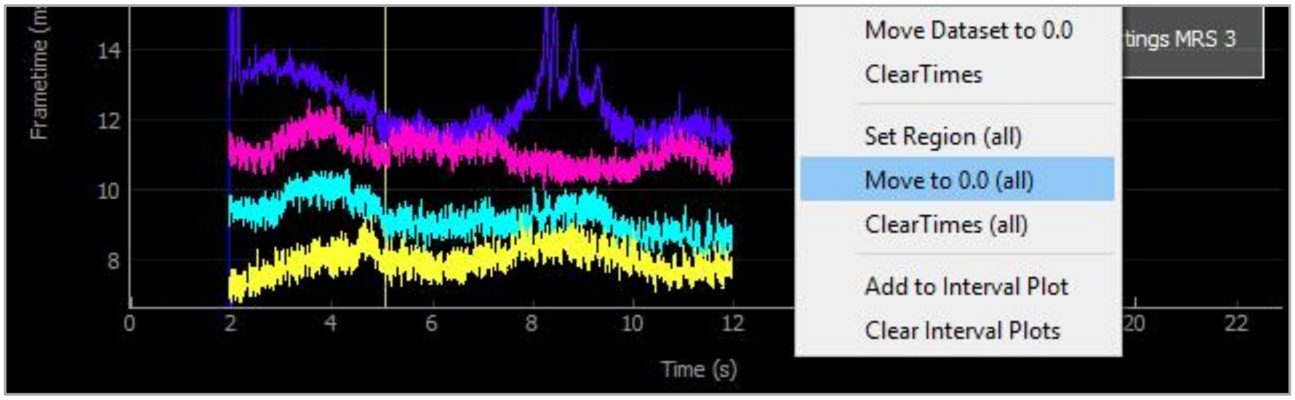


We decided to set the region between 2 seconds and 12 seconds.

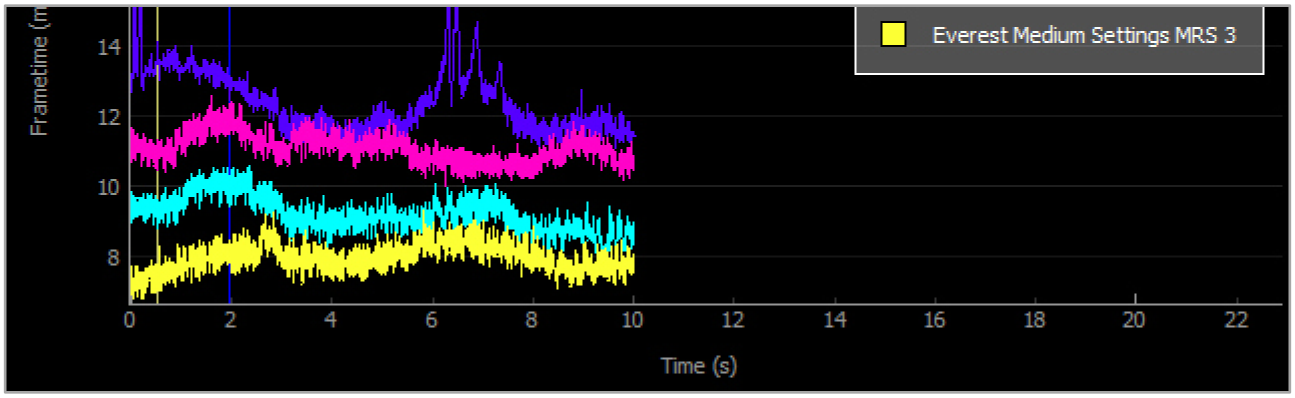

Frametime (ms)
16
14
12
10
8
6
0
2
4
6
8
10
12
14
16
18
20
22
Everest Medium Settings MRS 1
Everest Medium Settings MRS 2
Everest Medium Settings MRS 3

## Move to 0.0 (all)

This setting allows all line charts to be realigned to the 0 second starting time on the chart. Once these regions have been set using the blue and yellow vertical lines, right-click on the plot and select **Move to 0.0 (all)**.



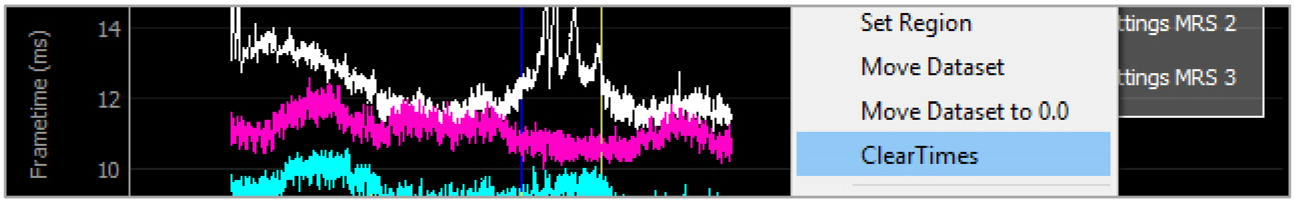
Once complete, all of the lines should begin at the 0 second point on the X-axis.

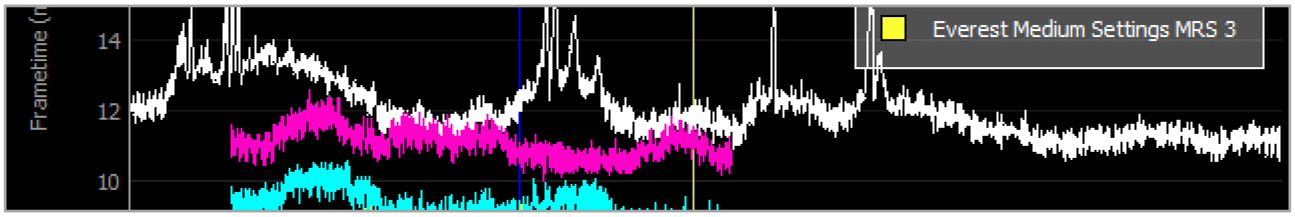


## Clear Times

Notice how the chart lines used in the example above are not properly aligned? It would have been better to move the lines first and then trim them by setting the region.

To undo the trim, select the chart line, right-click, and then select **Clear Times**.





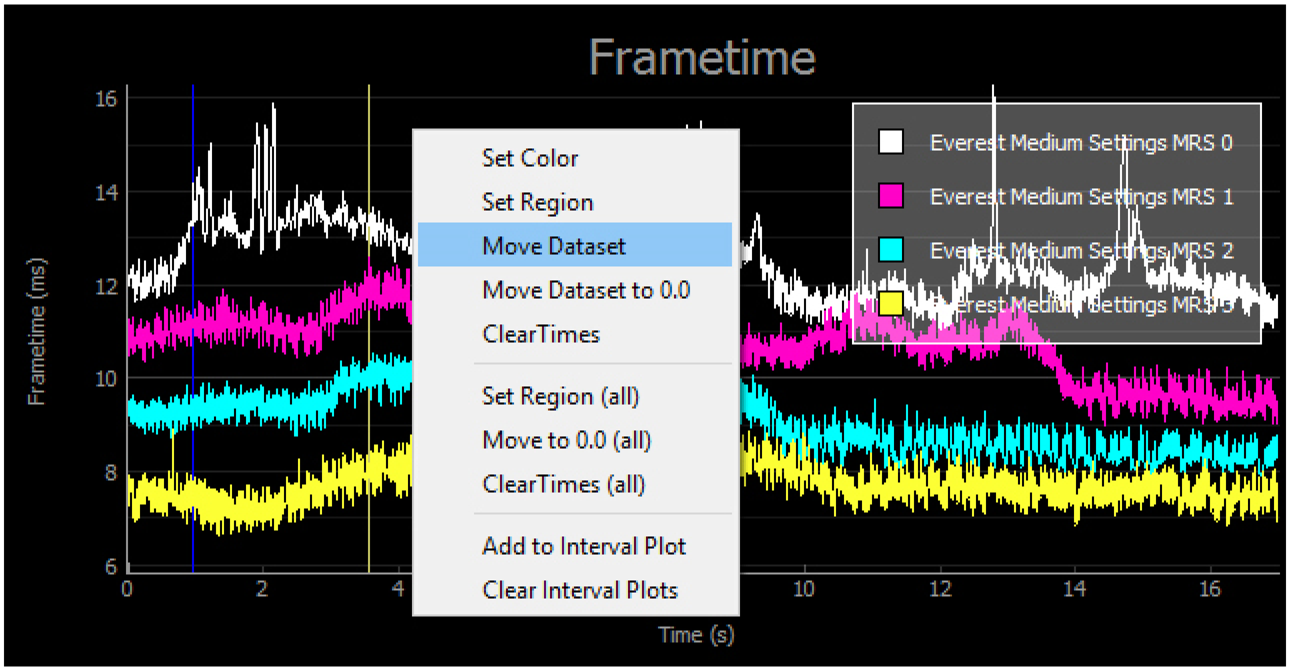
**NOTE:** You will need to do this individually for each chart line.

## Clear Times (all)

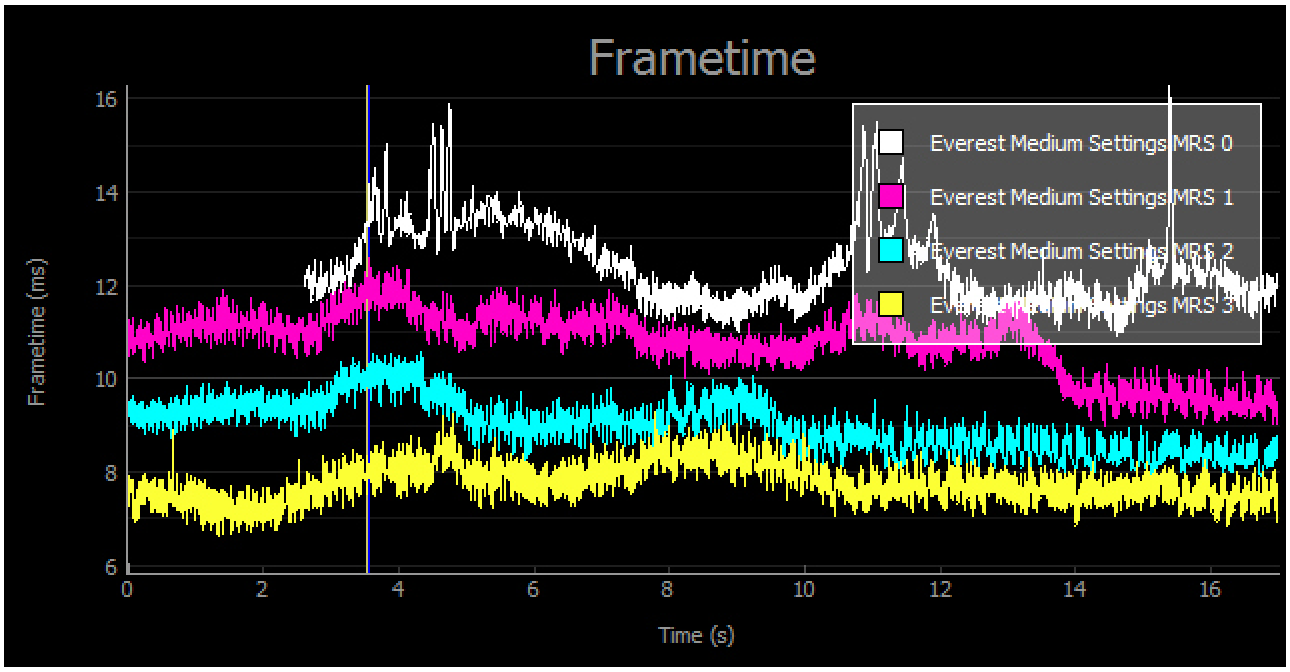
This works like Clear Times, but affects all chart lines in the plot.

## Move Dataset

To move a chart line, select the new area you would like to move it using the left-mouse button. This will place a blue vertical line. Then click on the chart using the right-mouse button to drop the yellow vertical line where you want the chart to be moved.



After clicking **Move Dataset**, the chart line will move to the area. If you accidentally moved it to the wrong spot, then simply select the line, right-click on the plot, and select **Clear Times** to reset the chart line.

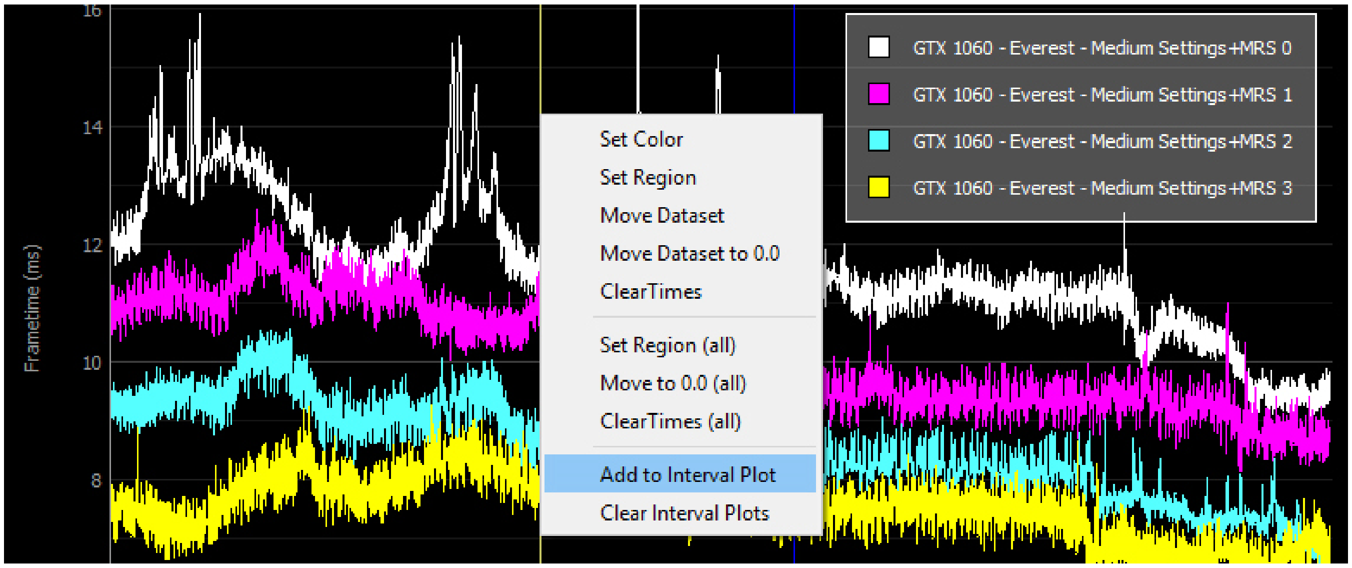


## Move Dataset to 0.0

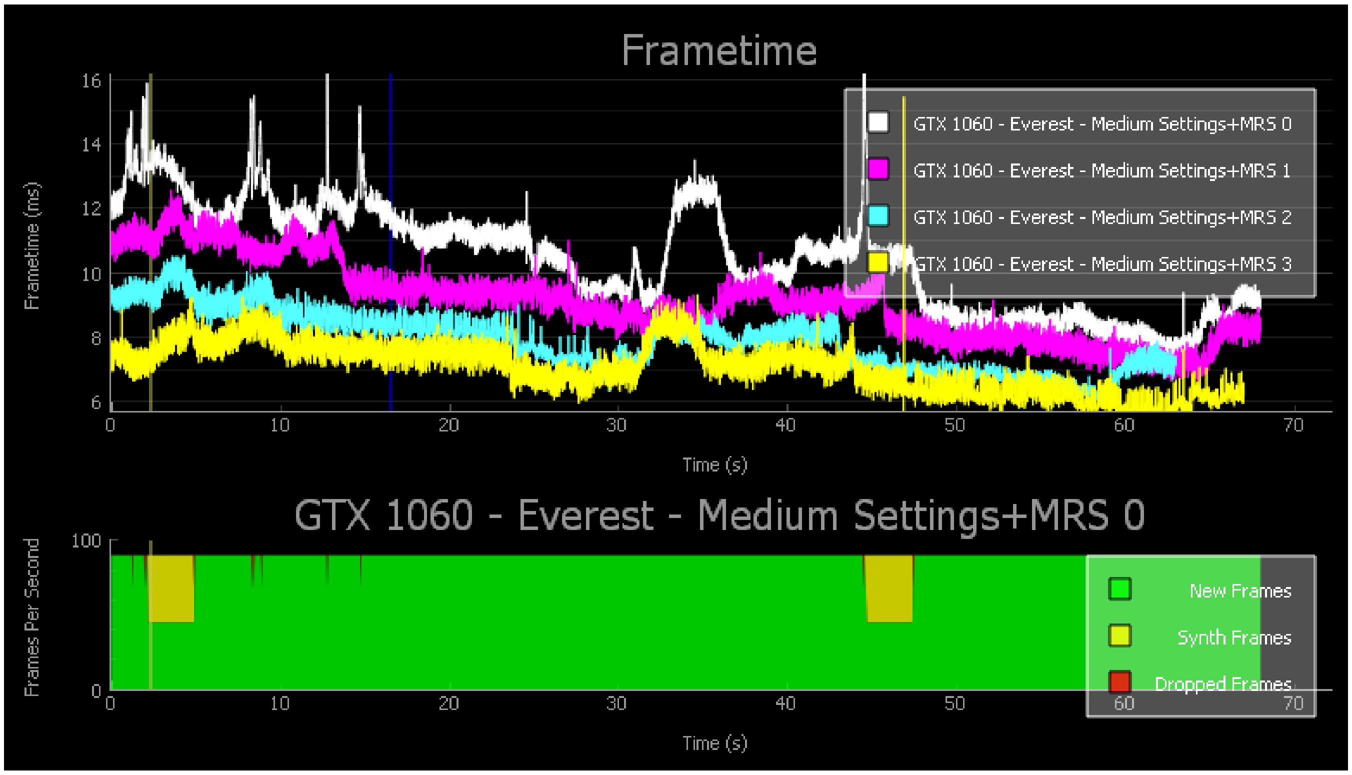
This will realign all chart lines in the plot back to 0 seconds on the X-axis.

## Add to Interval Plot

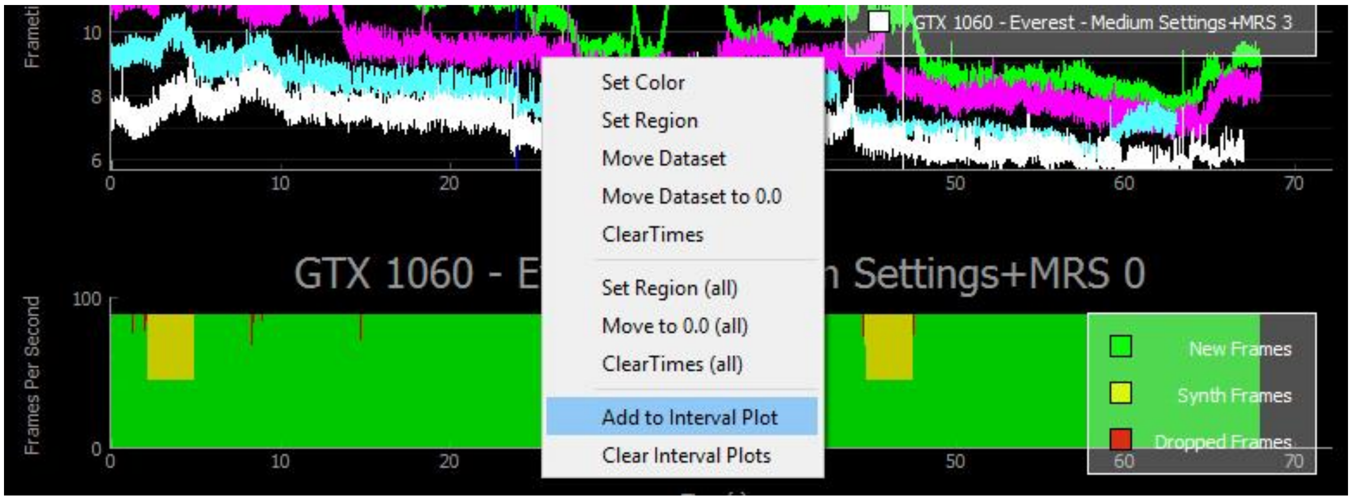
The Interval Plot shows data for new frames, synthetic frames, and dropped frames. To create an Interval Plot, select a chart line, right-click on the plot, and then select **Add to Interval Plot**.



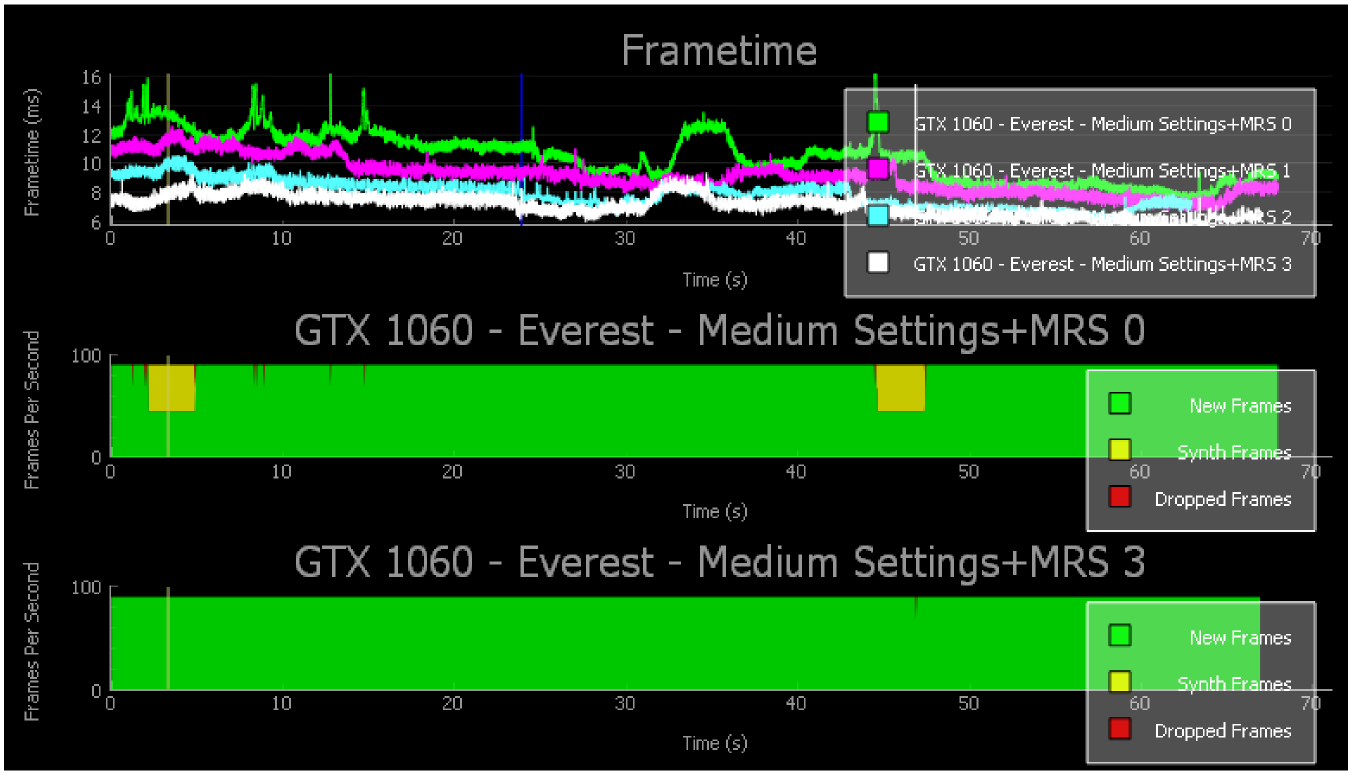
An Interval Plot makes it easy to see the synthetic and dropped frames that occurred in the VR application during SW capture. You can easily compare the chart line to the Interval Plot to see where the synthetic frames occurred in the VR capture below. Everest with MRS 0 shows synthetic frames between 2.5 and 6 seconds where the chart line strays above 11.1 ms.



To add another Interval Plot, simply select another chart line, right-click in the plot area, and select **Add to Interval Plot**.



This time we selected the MRS 3 chart line which shows that no synthetic or dropped frames occurred when using MRS 3 in VR with Everest using Medium Settings.



## Game Recommendations & Performance Results

Please refer to the **FCAT VR Benchmarking Guide** for information on the following:

- Recommended GeForce GTX GPUs for VR
- Architecture details for VRWorks technologies including Multi-Res Shading (MRS), Lens-Matched Shading (LMS), and VR SLI
- Recommended VR Games and Settings
- Information about testing MRS and LMS games
- Information about testing VR SLI games

Contact your local NVIDIA PR representative for a copy of the **FCAT VR Benchmarking Guide**.

Today's leading high-end VR headsets, the Oculus Rift and HTC Vive, both refresh their screen at a fixed interval of 90 Hz, which equates to one screen refresh every ~11.1 milliseconds (ms). VSYNC is enabled to prevent tearing, since tearing in the HMD can cause major discomfort to the user.

VR software for delivering frames can be divided into two parts: the **VR Game** and the **VR Runtime**. When timing requirements are satisfied and the process works correctly, the following sequence is observed:

1. The **VR Game** samples the current headset position sensor and updates the camera position in a game to correctly track a user's head position.
2. The game then establishes a graphics frame, and the GPU renders the new frame to a texture (not the final display).
3. The **VR Runtime** reads the new texture, modifies it, and generates a final image that is displayed on the headset display. Two of these interesting modifications include color correction and lens correction, but the work done by the VR Runtime can be much more elaborate.

The following figure shows what this looks like in a timing chart.

## Ideal VR Pipeline

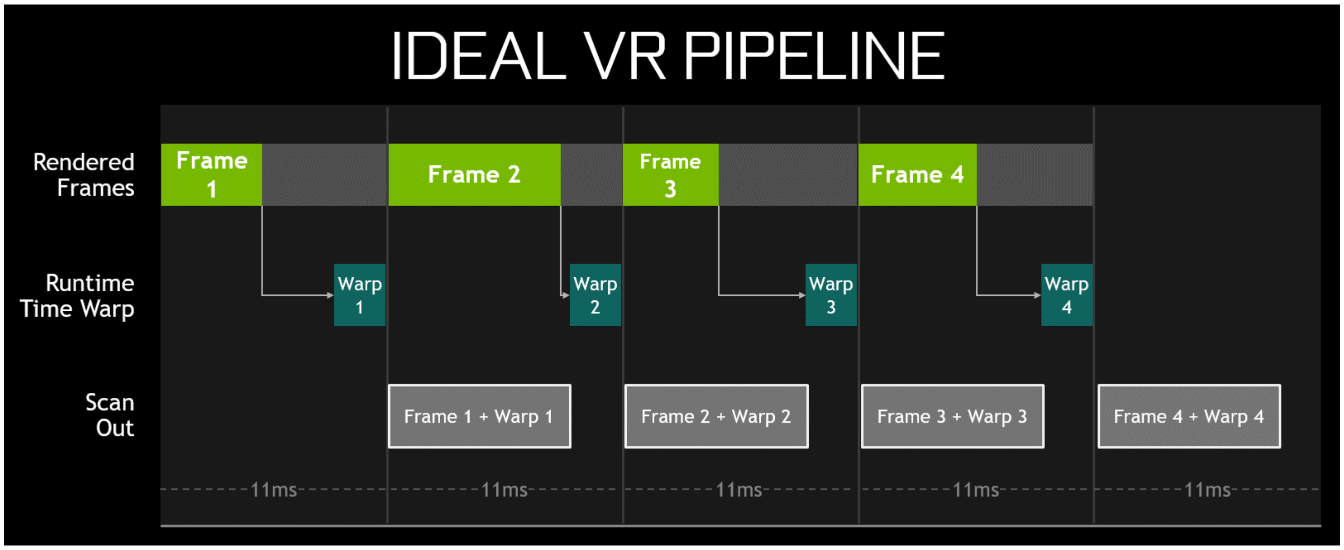


**Figure 3**: The ideal VR pipeline

The job of the Runtime becomes significantly more complex if the time to generate a frame exceeds the refresh interval. In that case, the total elapsed time for the combined VR Game and VR Runtime is too long, and the frame will not be ready to display at the beginning of the next scan.

In this case, the HMD would typically redisplay the prior rendered frame from the Runtime, but for VR that experience is unacceptable because repeating an old frame on a VR headset display ignores head motion and results in a poor user experience.

Runtimes use a variety of techniques to improve this situation, including algorithms that synthesize a new frame rather than repeat the old one. Most of the techniques center on the idea of reprojection, which uses the most recent head sensor location input to adjust the old frame to match the current head position. This does not improve the animation embedded in a frame—which will suffer from a lower frame rate and judder—but a more fluid visual experience that tracks better with head motion is presented in the HMD.

**FCAT VR Capture captures four key performance metrics for Rift and Vive:**

- Dropped Frames (also known as App Miss or App Drop)
- Warp Misses
- Frametime data
- Asynchronous Space Warp (ASW) synthesized frames

## Dropped Frames

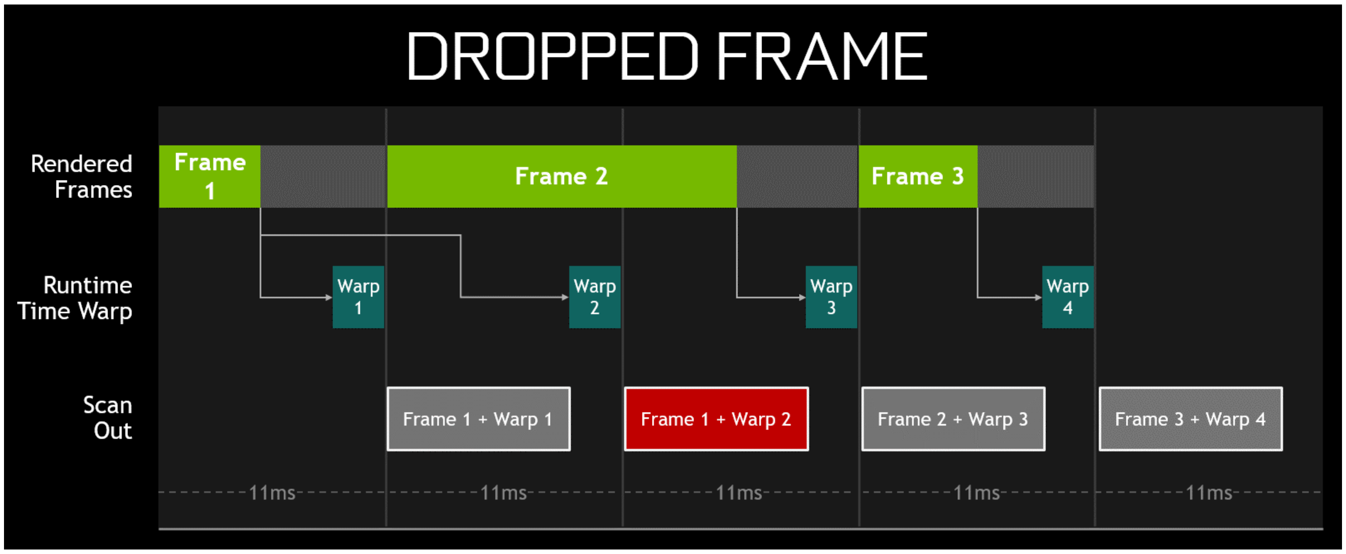


**Figure 4**: Application Dropped frame (App miss)

Whenever the frame rendered by the VR Game arrives too late to be displayed in the current refresh interval, a **Frame Drop** occurs and causes the game to stutter. Understanding these drops and measuring them provides insight into VR performance.

## Synthesized Frame

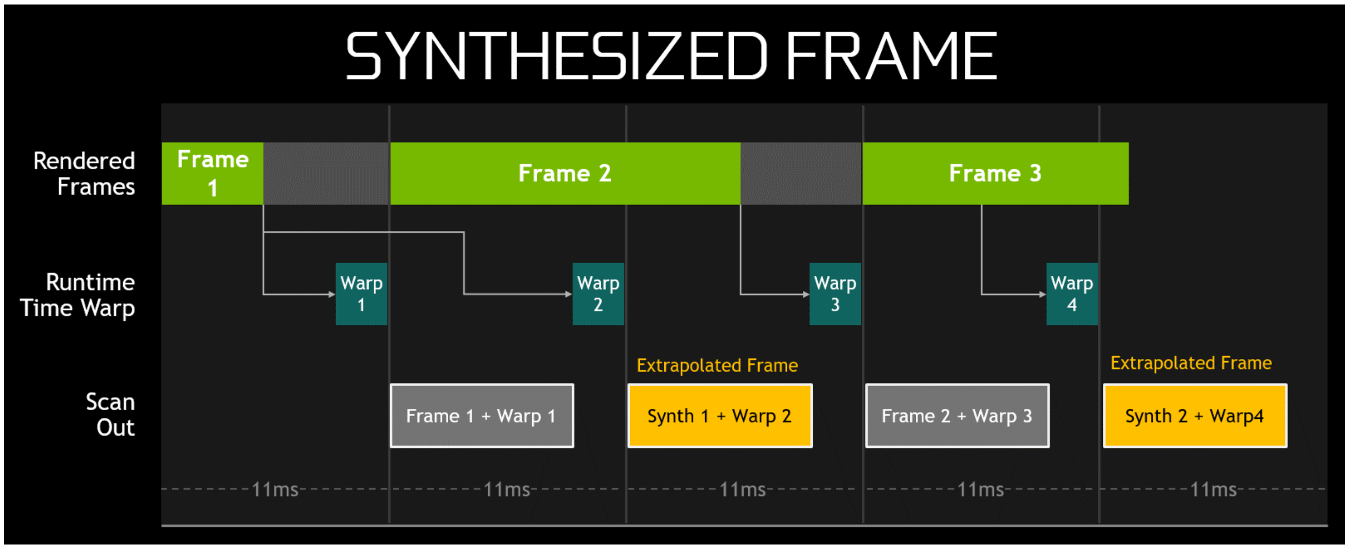


**Figure 5:** A Synthesized frame (ASW)

**Synthesized Frames:** Asynchronous Spacewarp (ASW) is a process that applies animation detection from previously rendered frames in order to synthesize a new, predicted frame.

See the **Asynchronous Spacewarp (ASW) section** below for more information on how it works.

## Warp Misses

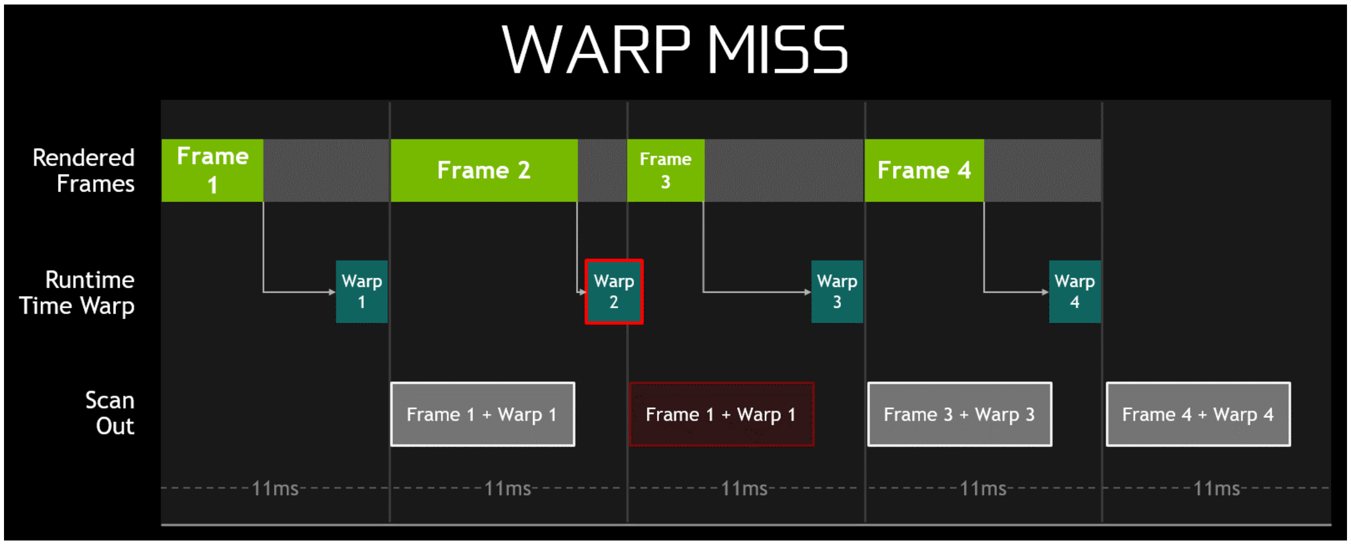


**Figure 6:** A Runtime Dropped frame (Warp miss)

**Warp misses** are a more significant issue for the VR experience. A warp miss occurs whenever the runtime fails to produce a new frame (or a reprojected frame) in the current refresh interval. In the preceding figure, a prior warped frame is reshown by the GPU. The VR user experiences this frozen time as a significant stutter.

### FrameTime

Since FCAT VR provides detailed timing, it is also possible to accurately measure unconstrained FPS for any title. By examining how long the system takes to render each frame, we see just how quickly the system *could have* displayed that frame if not for the fixed 90 Hz refresh cadence. By using this information, FCAT VR Capture looks inside the frame to calculate estimated headroom and is able to truly measure relative GPU performance of demanding VR content within the fixed refresh VR ecosystem.

## Oculus Rift | FCAT VR Capture Details

FCAT VR Capture directly accesses performance information provided by the **Oculus** runtime logged to ETW. When the hotkey is pressed, FCAT VR Capture captures the required events on the fly, converts these events into readable timestamps, and then logs it to a CSV file.

The following timestamps are generated today with Oculus:

- App Render Begin
- App Render Completion
- App Miss
- Warp Render Begin
- Warp Render Completion
- Warp Miss

The current SDK version support is <= 1.11.

### Oculus — FCAT VR Capture Columns Explained

The table below shows FCAT VR Capture columns and corresponding Oculus events.
**NOTE:** N/A represents "Not Available", and some ETW events do not have fields.

| FCAT VR Capture Fields | Units | Oculus Event Name (id) | Oculus Event Field | Description |
|---|---|---|---|---|
| **Frame Index** | integer counter | Return(1) and EndFrameAppTiming(44) | FrameID from Return event and FrameIndex from EndFrameAppTiming event | Game specifies this frame index in ovr_SubmitFrame. |
| **Game Start** | milliseconds, absolute | PhaseSyncEnd(36) | BeginFrameTime | Timestamp when Game starts preparing frame. |
| **Game Complete CPU** | milliseconds, absolute | PhaseSyncEnd(36) | EndFrameTime | Timestamp when Game has prepared frame (i.e. all CPU side work has been done). |

| FCAT VR Capture Fields | Units | Oculus Event Name (id) | Oculus Event Field | Description |
|---|---|---|---|---|
| **Game Complete GPU** | milliseconds, absolute | PhaseSyncEnd(36) | CompletionTime | Timestamp when frame finished on GPU. |
| **Queue Ahead** | milliseconds | PhaseSyncEnd(36) | QueueAhead | Amount of queue ahead that was allowed for the frame. For adaptive queue ahead please refer to following link: https://developer.oculus.com/documentation/pcsdk/latest/concepts/dg-render/#dg-queue-ahead. |
| **Runtime Sample** | milliseconds, absolute | CompositionEndSpinWait (53) | NA - Timestamp when event was generated | Timestamp for warp start. Usually fixed amount of time before VSYNC. |
| **Runtime Complete** | milliseconds, absolute | CompositionEnd(49) | NA - Timestamp when event was generated | Warp finished on GPU. |
| **VSYNC** | milliseconds, absolute | NV specific event, not available on non-NV HW | NA - Timestamp when event was generated | VSYNC interrupt for HMD (NVIDIA only). |
| **App Miss** | integer counter | ClientFrameMissed(47) | Not used | Incremented for each ClientFrameMissed event and reset after warp complete event |
| **Warp Miss** | integer counter | CompositionMissedCompositorFrame(56) | Not used | N/A |
| **ASW Status** | binary 0 or 1 | Computed value based on call(0) and return(1) events | Call(0) Return (1) | Call(0) and Return(1) events contain FrameID field. Interpolated FrameID's are 0, if interpolated frames are present, ASW status is reported as 1, otherwise 0. |

## Asynchronous Spacewarp (ASW)

Asynchronous Spacewarp (ASW) is a technology developed by Oculus aimed at improving smoothness on mainstream GPUs. At this time, ASW is enabled by default in the public runtime.

### How does ASW Work?

In order to understand ASW, we must first understand Asynchronous Timewarp (ATW). ATW is a process that is separate from the main rendering thread and runs within the Oculus Runtime where the HMD position is sampled very close to the VSYNC interval, the difference from the previous position is calculated, the most recently completed frame is translated (shifted without full re-rendering) based on the position difference, and the new translated frame is displayed on the HMD.

ASW is a process that applies animation detection from previously rendered frames in order to synthesize a new, predicted frame. Colloquially, we can refer to this as an ASW synthesized frame.

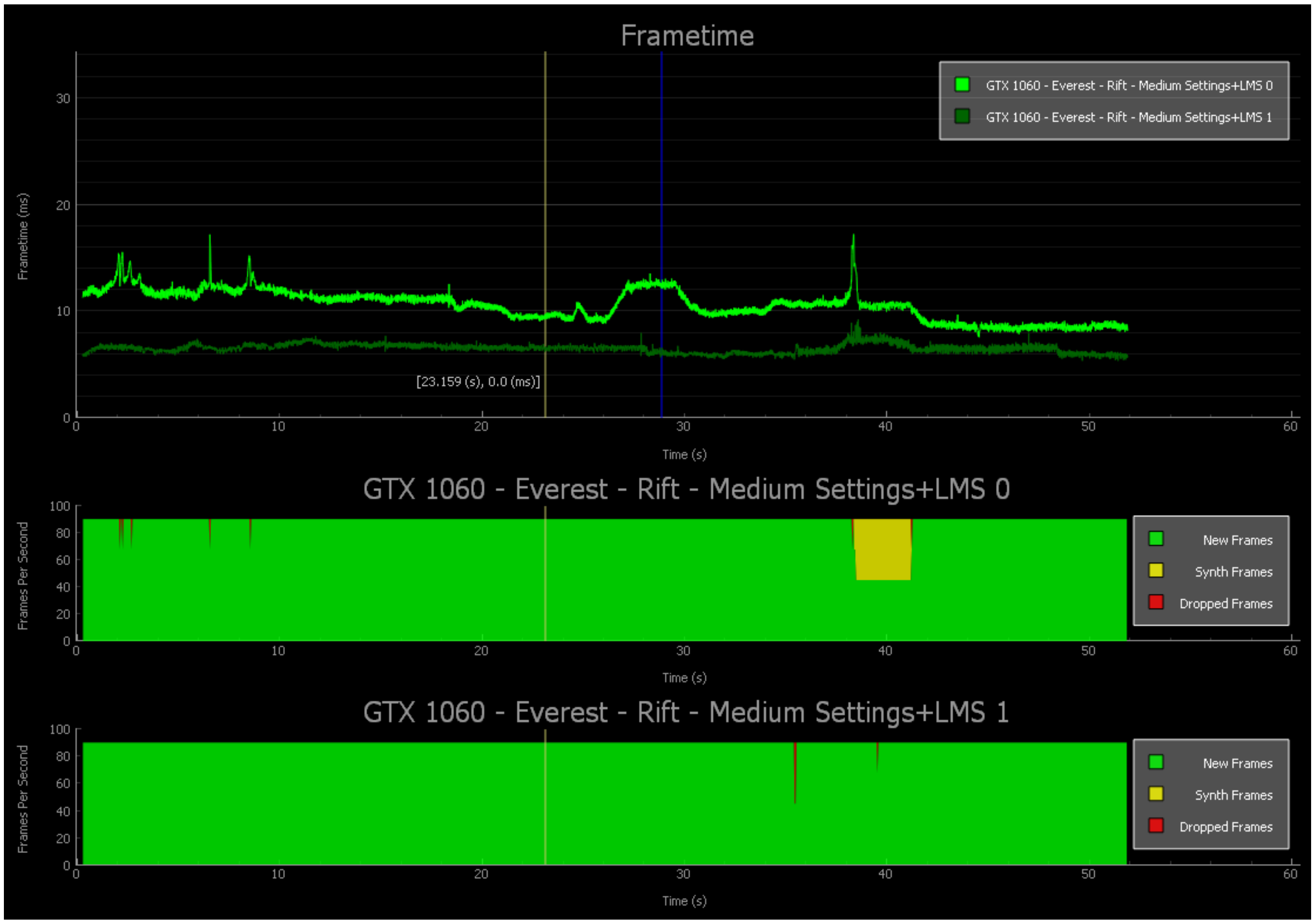
If the application is able to consistently render at 90 Hz, the synthesized frames are never displayed in the HMD. ASW is "activated" when a frame cannot be rendered as per usual, on time. Predicting a synthesized frame based on motion detection from previously rendered frames is less demanding than rendering a new frame.

If ASW is disabled and an application fails to submit frames to the Oculus Runtime at 90 Hz, the Runtime will select the most recently completed frame and apply ATW to it.

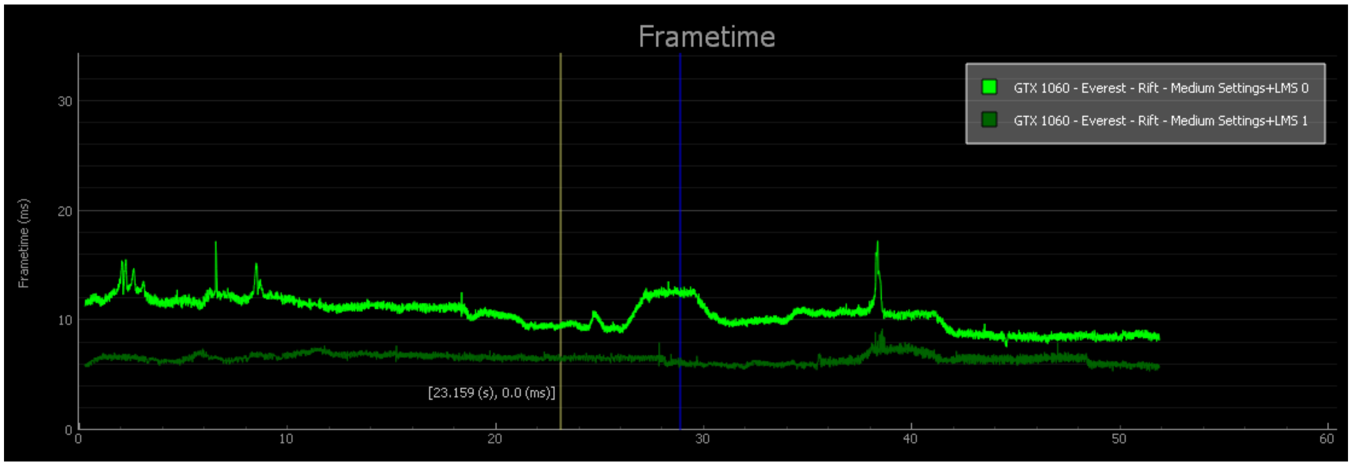
If ASW is enabled and an application fails to submit frames to the Oculus Runtime at 90 Hz, the Runtime renders the application at 45 FPS and applies ATW to both regularly rendered frames and ASW synthesized frames. These ASW synthesized frames act as intermediary frames between the regularly rendered frames. The end result is that the viewer sees smoother animation, rendered at 45 FPS, but presented at 90 FPS.

## Output Graphs

FCAT VR Analyzer generates graphs which clearly show this behavior seen in the HMD. The topmost chart represents the time it took to render the frame which appears in the HMD. The bottom two charts (green rectangles) show dropped frames, synthesized frames, and warp misses (which are shown as dropped frames).



Looking closer at the frametime graph (below), we see a comparison of LMS settings in the game Everest on the Oculus Rift. Notice that the bright-green capture line with LMS set to 0 (Off) usually stays at 11.1 ms, which correlates to 90 FPS. However, there are minor spikes that go above 11ms.



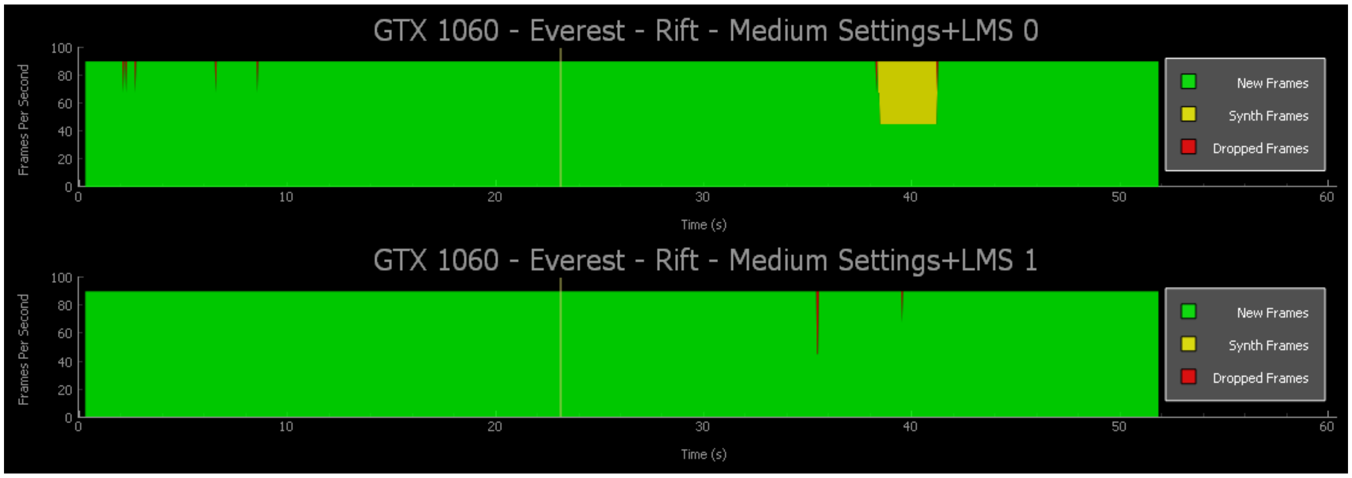
The colors below the frametime graph (green rectangles) represent the instantaneous percentage of frame types within any given second within 90 FPS. This data answers these questions:

- Over the previous second, **how many frames were real and fully rendered at 90 FPS?**
- **How many frames were synthesized instead of fully rendered?**
- And **how many frames were dropped?**

Continuing to look at the Everest data, we see that that the spikes above 11ms resulted in several dropped frames, and a rather large spike at approximately 38-39 seconds, and several synthesized frames over ~5 seconds due to that large spike. However, using NVIDIA LMS (1) fixes most of the issues.

**These charts show that FCAT VR Capture allows for a more accurate understanding of what is happening behind the scenes.**

A cursory analysis of these graphs can be accomplished by considering how much green is seen in the graph. The greener the graph, the more real frames are presented, and the better the experience.



# HTC Vive | FCAT VR Capture Details

**HTC Vive** uses the SteamVR-based OpenVR SDK. FCAT VR Capture uses a performance API exposed by SteamVR to generate the timestamps in the same format used for Oculus timestamps. When the hotkey is pressed, FCAT VR Capture records these events on the fly, converts these events into readable timestamps, and then logs the event into a CSV file. OpenVR SDK versions 0.5 to 0.19 (both included) are supported.

## OpenVR — FCAT VR Capture Columns Explained

The following table shows FCAT VR Capture columns and corresponding HTC events.

| FCAT VR Capture Fields | Units | OpenVR Performance API field name | Description |
|---|---|---|---|
| **Frame Index** | integer counter | m_nFrameIndex | |
| **Frame Start** | milliseconds, absolute | m_flSystemTimeInSeconds | Absolute time reference for comparing frames. This aligns with the VSYNC that running start is relative to. |
| **Render End** | milliseconds, absolute | m_flSystemTimeInSeconds + m_flPreSubmitGpuMs + m_flPostSubmitGpuMs | Absolute time for render finish on GPU. |
| **Warp End** | milliseconds, absolute | m_flSystemTimeInSeconds + m_flTotalRenderGpuMs | Absolute time for warp finish on GPU. |
| **HMD Sample Start** | milliseconds, absolute | m_flSystemTimeInSeconds + m_flWaitGetPosesCalledMs | Absolute time for HMD sample. |
| **Number Times Presented** | integer counter | m_nNumFramePresents | Number of times this frame was presented. |
| **Number Miss Presented** | integer counter | m_nNumMisPresented | Number of times this frame was presented on a vsync other than it was originally predicted to. |
| **Number Dropped Frames** | integer counter | m_nNumDroppedFrames | Number of additional times previous frame was scanned out. |
| **Reprojection Flags** | integer | m_nReprojectionFlags | 4 - Asynchronous reprojection is enabled. |
| **Scene Render GPU Time** | milliseconds, | m_flPreSubmitGpuMs | Time spent rendering the scene (GPU work submitted between WaitGetPoses and second Submit). |
| **Total GPU Render Time** | milliseconds | m_flTotalRenderGpuMs | Time between work submitted immediately after present (ideally VSYNC) until the end of compositor submitted work. |
| **Compositor GPU Render Time** | milliseconds | m_flCompositorRenderGpuMs | Time spend performing distortion correction, rendering chaperone, overlays, etc. |
| **Compositor CPU Render Time** | milliseconds | m_flCompositorRenderCpuMs | Time spent on CPU submitting the above work for this frame. |

| FCAT VR Capture Fields | Units | OpenVR Performance API field name | Description |
|---|---|---|---|
| **Compositor Idle Time** | milliseconds | m_flCompositorIdleCpuMs | Time spent waiting for running start (application could have used this much more time). |
| **Client Frame Interval** | milliseconds | m_flClientFrameIntervalMs | Time between calls to WaitGetPoses. |
| **Present Call CPU** | milliseconds | m_flPresentCallCpuMs | Time blocked on call to present (usually 0.0, but can go long). |
| **Wait for Present CPU** | milliseconds | m_flWaitForPresentCpuMs | Time spent spin-waiting for frame index to change (not near-zero indicates wait object failure). |
| **Submit Frame** | milliseconds | m_flSubmitFrameMs | Time spent in IVRCompositor::Submit (not near-zero indicates driver issue). |
| **New Poses Ready** | milliseconds | m_flNewPosesReadyMs | |
| **New Frame Ready** | milliseconds | m_flNewFrameReadyMs | Second call to IVRCompositor::Submit. |
| **Compositor Update Start** | milliseconds | m_flCompositorUpdateStartMs | |
| **Compositor Update End** | milliseconds | m_flCompositorUpdateEndMs | |
| **Compositor Render Start** | milliseconds | m_flCompositorRenderStartMs | |
| **ProcessID** | integer | | Process ID of the Game/Application. |
| **API Version** | integer | | OpenVR API version. |

## Reprojection

SteamVR uses two modes of reprojection:

- Interleaved Reprojection
- Asynchronous Reprojection

Both modes are supported on NVIDIA GPUs and are enabled by default. Reprojection can only correct for rotation, similar to Oculus' Asynchronous Time Warp (ATW).

For a detailed explanation of Reprojection, please refer to the presentation by Alex Vlachos from Valve: https://youtu.be/DdL3WC_oBO4?t=21m52s.