

EUC Score

Hands-on - How to record and visualize Windows performance counters and screen videos

E2EVC 2024, Madrid

Benny Tritsch | info@drtritsch.com | [@drtritsch](https://twitter.com/drtritsch)



Performance Data Scientist
EUC Documentary Cameraman
MVP | CTP | EUC Expert | NGCA



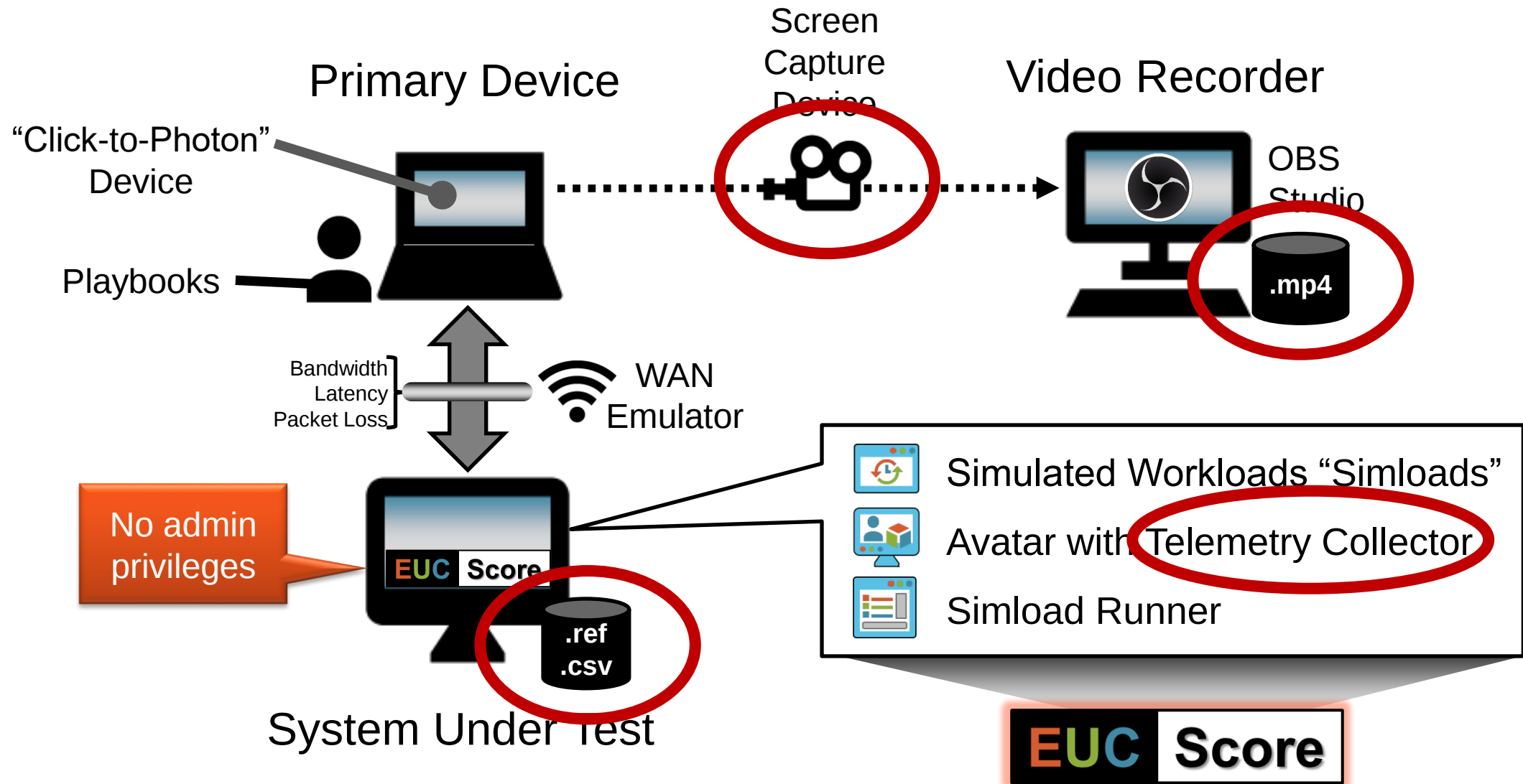
info@drtritsch.com
@drtritsch



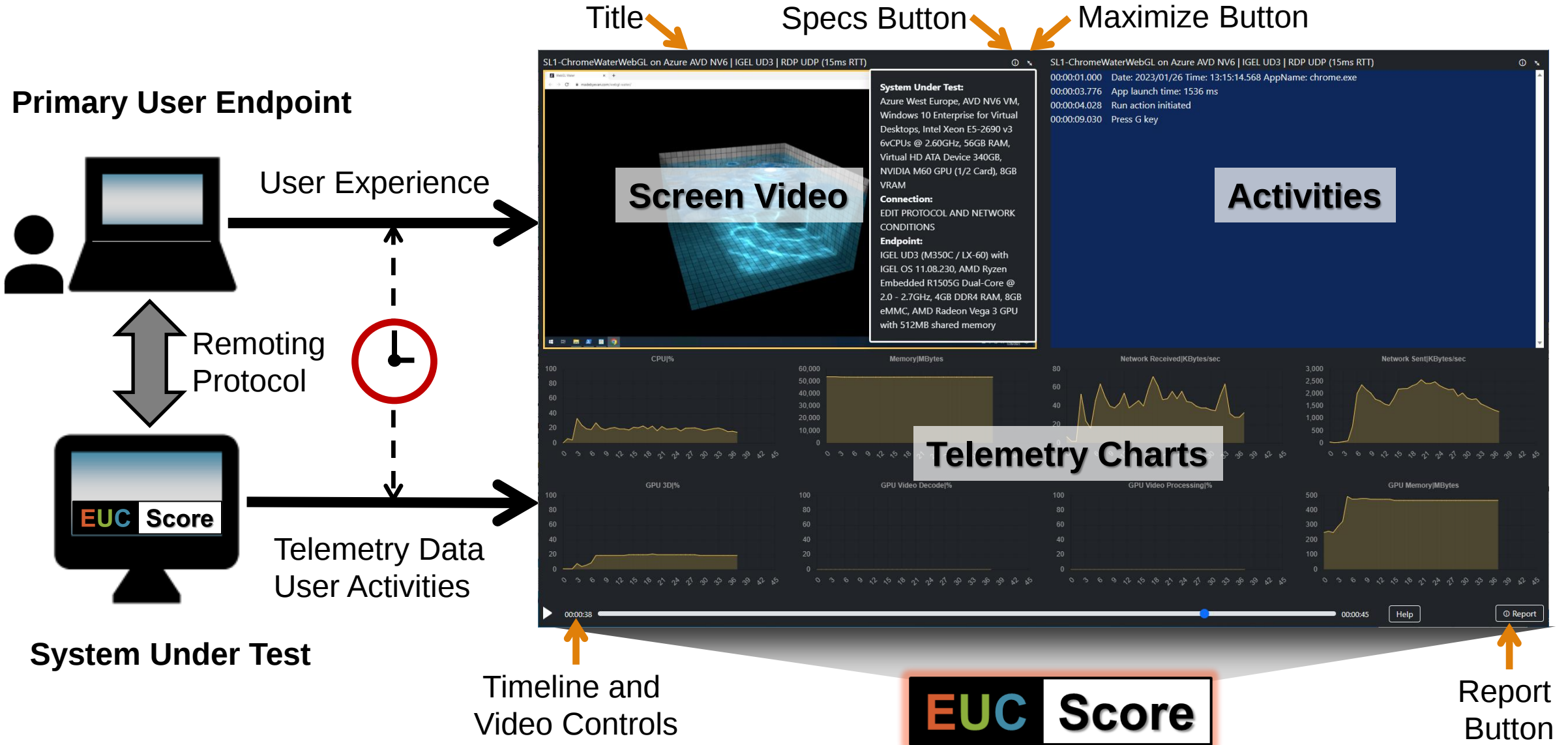
Agenda

- Collecting Windows Performance Counters
- Recording Screen Videos with Frame Grabbers
- Visualizing Data Using HTML5

Build a DEX Test Lab with EUC Score



Visual Data Analytics – Sync Player



Collecting Windows Performance Counters

Windows Performance Counters

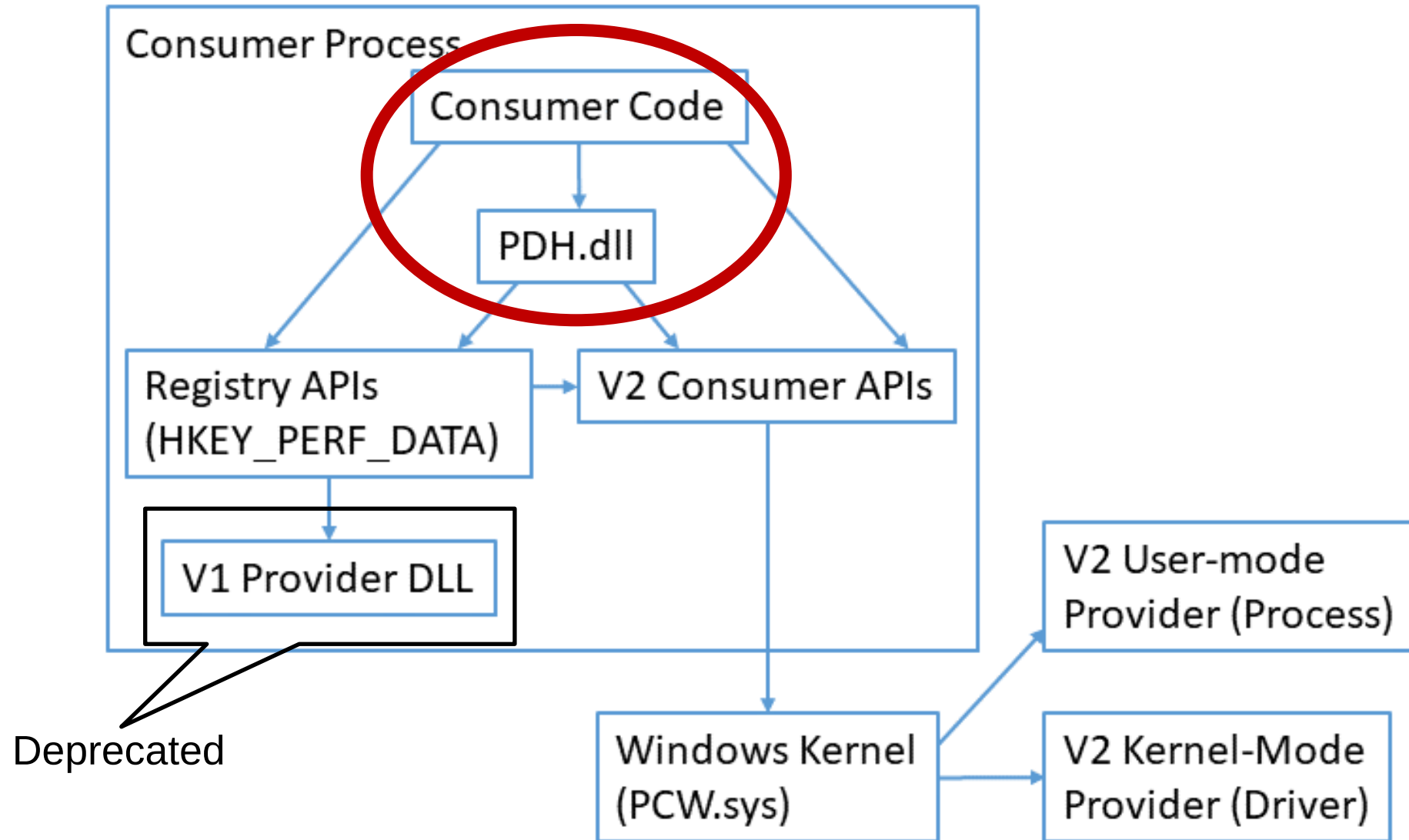
Consistent interface for collecting various kinds of system data

- A **provider** is a software component that generates and publishes performance data
- A **counterset** (or **object**) is a grouping of performance data within a provider
- A **counter** is the definition of single piece of performance data
- An **instance** is an entity about which performance data is reported
- A **counter value** is the value of a single piece of performance counter data
- The **counter type** indicates the type of the counter's raw value and indicates what the counter's raw value represents

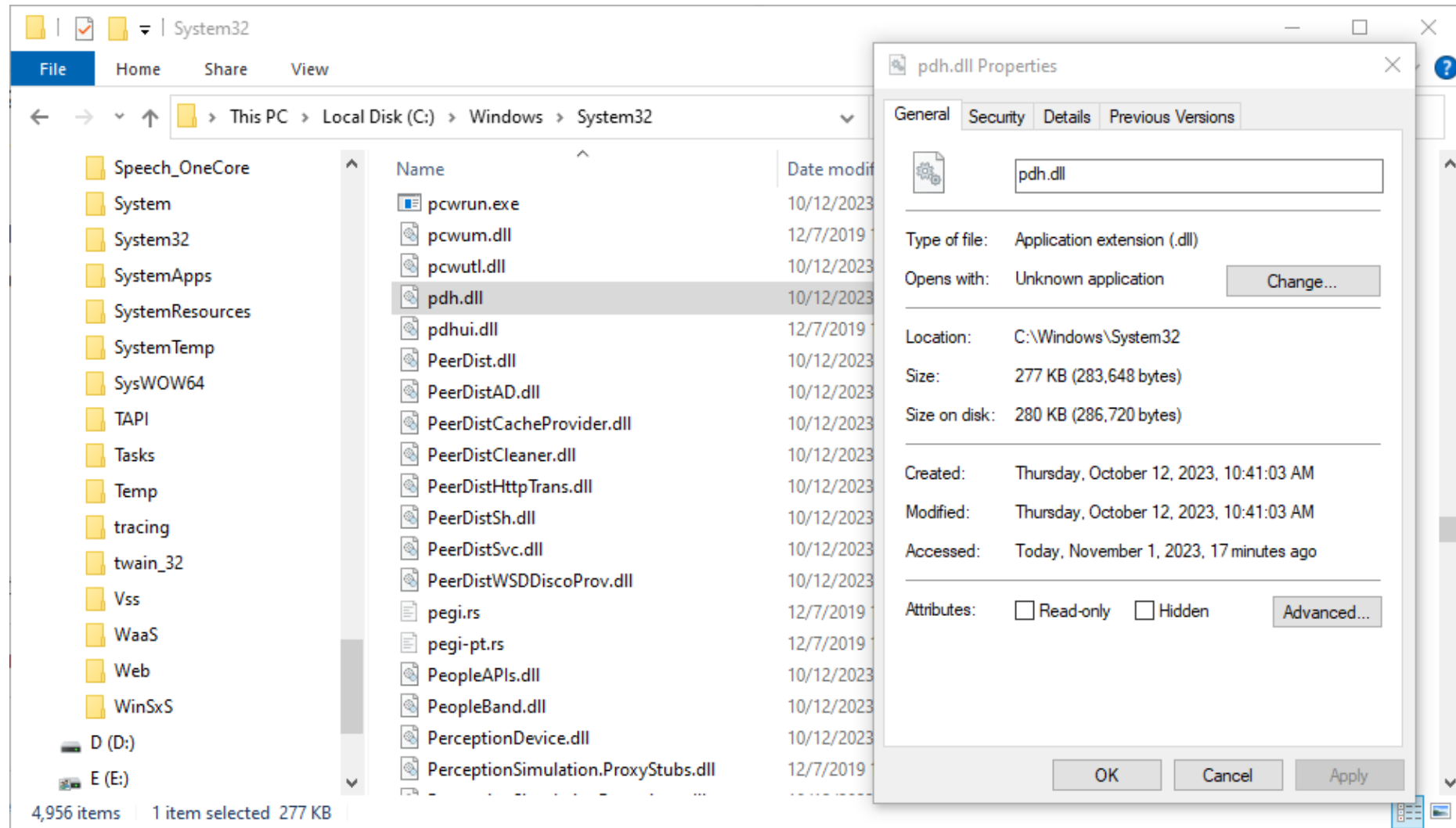
Windows Performance Counters

- **Single-instance countersets** always contain data for exactly one instance
- **Multi-instance countersets** contain data for a variable number of instances
- A **consumer** is a software component that makes use of performance data. It periodically collects and records the data from a provider's counterset:
 - GUI: Task Manager, Resource Monitor, Performance Monitor, and Sysinternals Process Explorer
 - CMD: Typeperf.exe, Logman.exe, and Relog.exe
 - EUC Score: Simload Base Counters, Telemetry Collector, Data Miner

Performance API Architecture

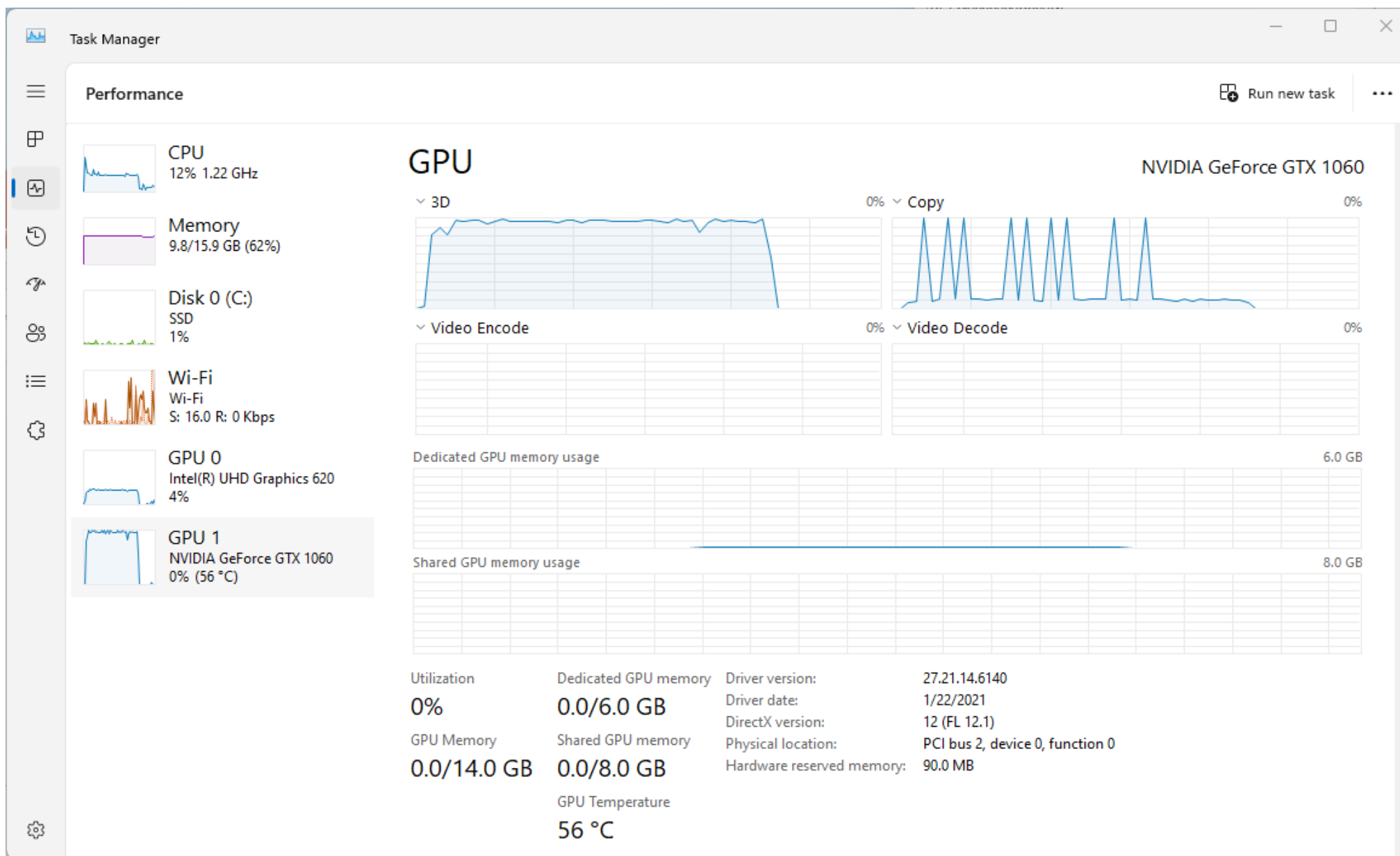


Windows Performance Data Helper DLL – PDH.dll



API: <https://learn.microsoft.com/en-us/windows/win32/api/pdh/>

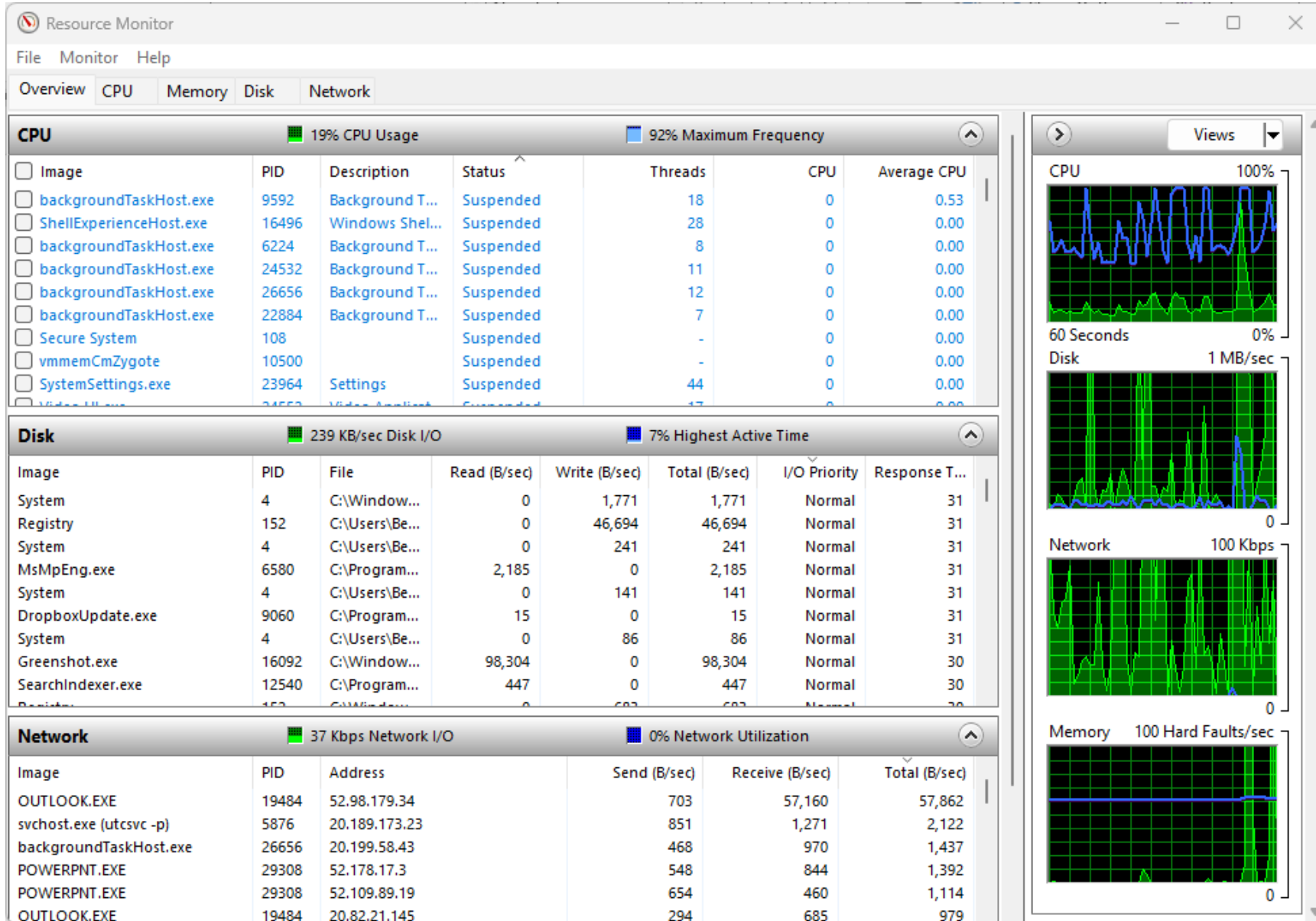
Task Manager – Performance Tab



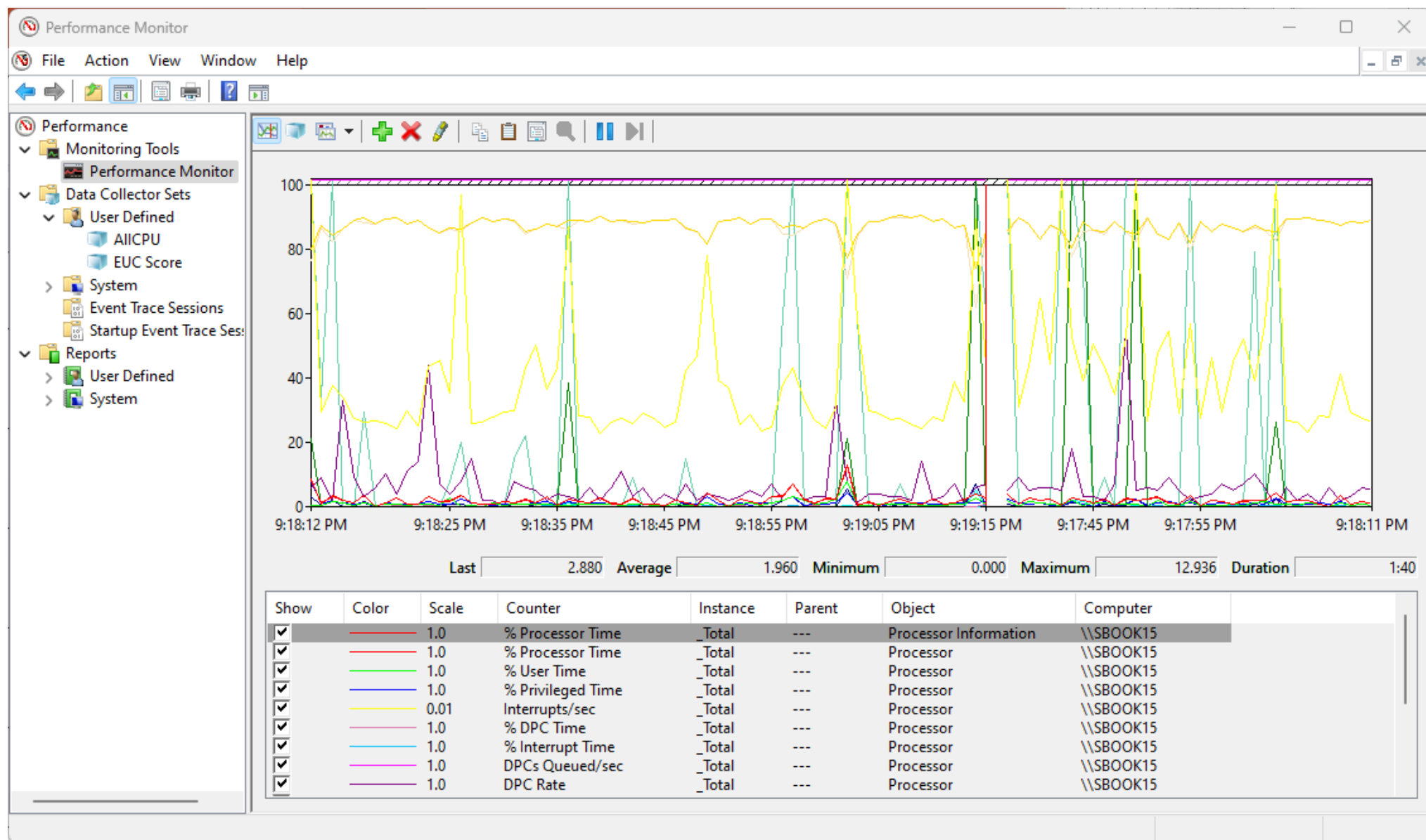
Task Manager – Detail Tab

Name	PID	Status	User name	CPU	Memory (a...	UAC virtualizat...	GPU	GPU engine
System Idle Process	0	Running	SYSTEM	97	8 K		00	
Taskmgr.exe	17940	Running	Benny	01	19,232 K	Not allowed	00	
Greenshot.exe	13732	Running	Benny	01	40,016 K	Disabled	00	
dwm.exe	1852	Running	DWM-1	00	79,524 K	Disabled	00	
chrome.exe	12008	Running	Benny	00	89,148 K	Disabled	00	
svchost.exe	1860	Running	NETWORK...	00	3,620 K	Not allowed	00	
csrss.exe	1012	Running	SYSTEM	00	1,480 K	Not allowed	00	
chrome.exe	16080	Running	Benny	00	120,872 K	Disabled	00	
POWERPNT.EXE	10020	Running	Benny	00	360,752 K	Disabled	00	
OUTLOOK.EXE	21672	Running	Benny	00	186,696 K	Disabled	00	
g2mlauncher.exe	14868	Running	Benny	00	17,144 K	Disabled	00	
PowerToys.exe	10764	Running	Benny	00	2,896 K	Disabled	00	
StreamDeck.exe	1612	Running	Benny	00	69,428 K	Disabled	00	
System interrupts	-	Running	SYSTEM	00	0 K		00	
PowerToys.Peek.UI.exe	13408	Running	Benny	00	12,972 K	Disabled	00	
ctfmon.exe	5024	Running	Benny	00	3,920 K	Disabled	00	
ControlCenter.exe	11940	Running	Benny	00	52,388 K	Disabled	00	
msedge.exe	7464	Running	Benny	00	173,968 K	Disabled	00	
explorer.exe	2232	Running	Benny	00	57,292 K	Disabled	00	
Skype.exe	10256	Running	Benny	00	69,572 K	Disabled	00	
ElgatoAudioControl...	15472	Running	Benny	00	824 K	Disabled	00	
slack.exe	17296	Running	Benny	00	167,384 K	Disabled	00	
chrome.exe	5540	Running	Benny	00	94,548 K	Disabled	00	
chrome.exe	8684	Running	Benny	00	14,828 K	Disabled	00	
mmc.exe	12184	Running	Benny	00	18,144 K	Not allowed	00	
Skype.exe	17704	Running	Benny	00	159,664 K	Disabled	00	
svchost.exe	9924	Running	Benny	00	10,588 K	Disabled	00	

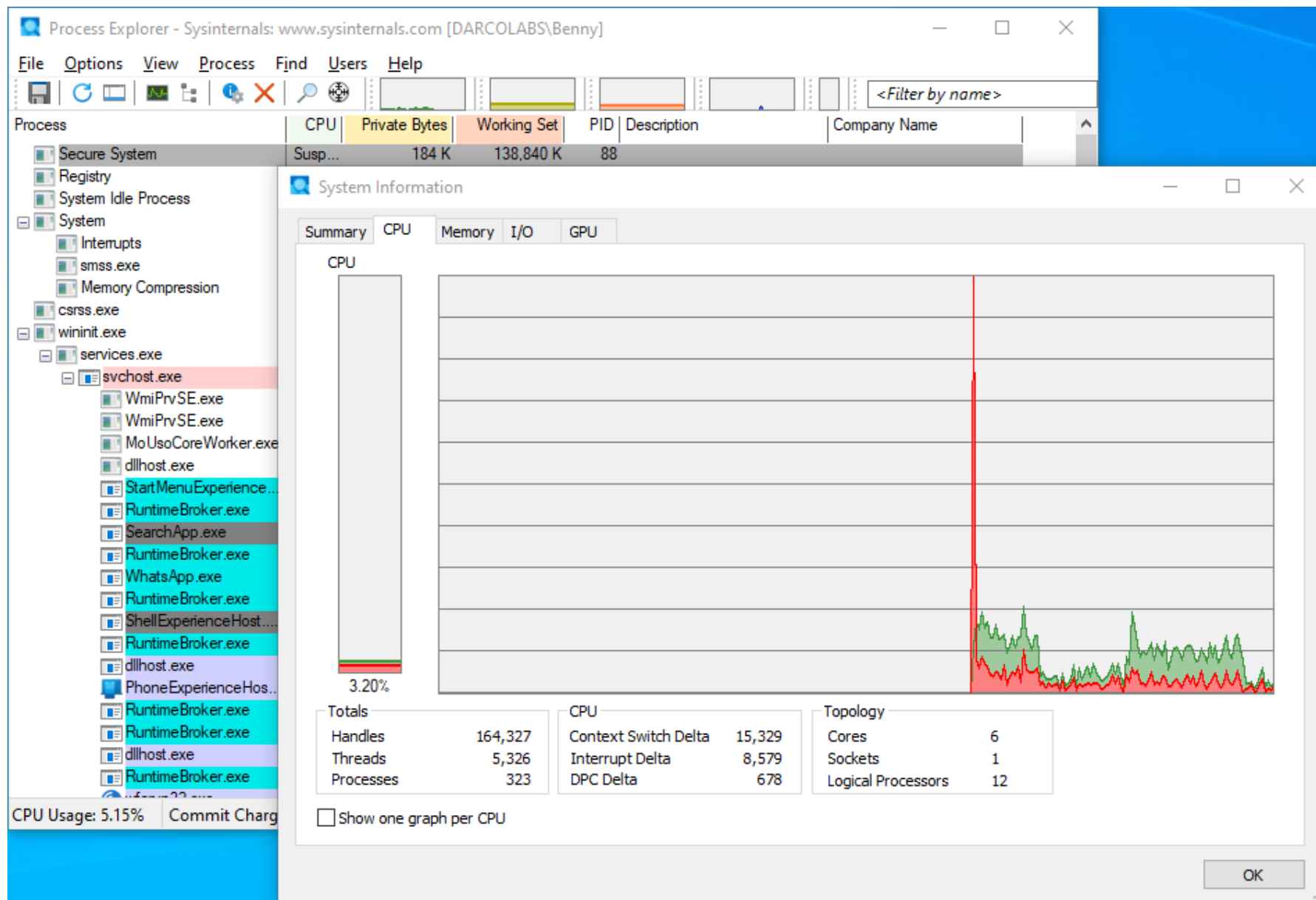
Resource Monitor



Performance Monitor



Sysinternals Process Explorer



PerfMon: Add Counters and Save Settings

The screenshot illustrates the process of adding performance counters in Windows Performance Monitor and saving the configuration. The 'Add Counters' dialog is open, showing the selection of the 'Processor' object and the '% Processor Time' counter. The 'Instances of selected object' list shows 'Total' selected. The 'Added counters' list is empty. A Notepad window displays the saved settings in E2EVC.tsv format.

Performance Monitor - Add Counters Dialog

Available counters
Select counters from computer: <Local computer> Browse...

Processor

- % C1 Time
- % C2 Time
- % C3 Time
- % DPC Time
- % Idle Time
- % Interrupt Time
- % Privileged Time
- % Processor Time

Instances of selected object:

- Total
- <All instances>
- 0
- 1
- 10
- 11
- 2
- 3

Search Add >>

☐ Show description

Added counters

Counter	Parent	Inst...
---------	--------	---------

Remove <<

E2EVC.tsv - Notepad

```
File Edit Format View Help
Reported on \\MERLIN
Date: 10/30/2023
Time: 12:45:57 PM
Time: Default
Data: Current Activity
Interval: 1.00 seconds

Computer: \\MERLIN

Object: Processor

        _Total
    % Processor Time      3.956

Object: Processor Information

        _Total
    % Processor Time      3.956

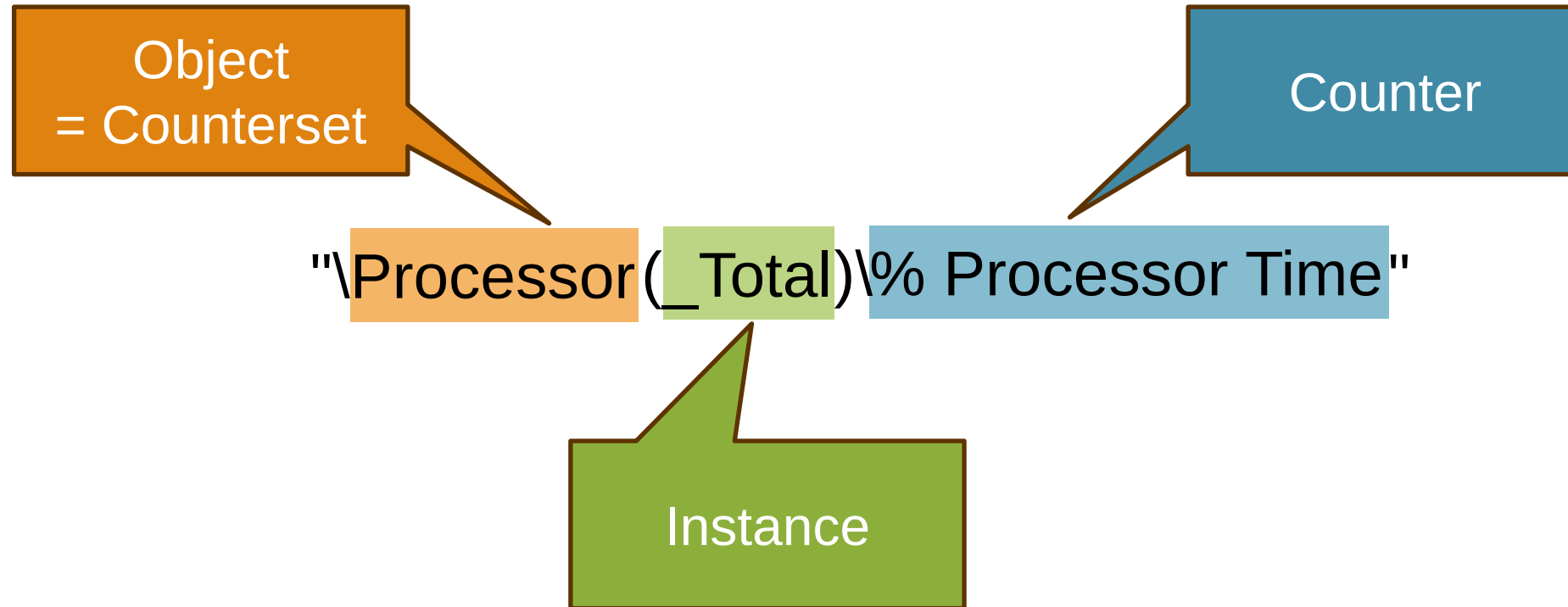
Ln 1, Col 1    100%    Windows (CRLF)    UTF-16 LE
```

Performance Monitor - Counter List

Counter	Instance	Value
% Processor Time	_Total	3.956
% Processor Time	_Total	3.956

Performance Counter Path Syntax

\\ComputerName\ObjectName(ObjectInstance)\ObjectCounter



A *counterset* is a grouping of performance data within a provider
A *counter* is the definition of single piece of performance data
An *instance* is an entity about which performance data is reported

Windows Management Interface

- WMI has preinstalled providers that monitor system performance on both the local system and remotely
- WMI can be used from scripts or from C/C++ applications
- The WmiPerfClass provider creates the classes derived from Win32_PerfRawData and from Win32_PerfFormattedData
- The WmiPerfInst provider supplies data dynamically to both raw and formatted classes
- **Example:** `Get-CimInstance -Query "select Name, PercentProcessorTime from Win32_PerfFormattedData_PerfOS_Processor" | Select Name, PercentProcessorTime`

CAUTION: WMI overhead can be significant!

PowerShell

Performance Counters

- `Get-Counter -ListSet "Processor"`
- `(Get-Counter -ListSet "Processor").Paths`
- `(Get-Counter -ListSet "Processor").PathsWithInstances`
- `Get-Counter -Counter "\Processor(_Total)\% Processor Time" -SampleInterval 2 -MaxSamples 3`
- `$CounterList = "\Processor(_Total)\% Processor Time", "\System\Processor Queue Length", "\Memory\Available MBytes", "\Process(_Total)\Working Set", "\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\PhysicalDisk(_Total)\Disk Transfers/sec", "\PhysicalDisk(_Total)\Current Disk Queue Length", "\System\Context Switches/sec", "\System\Processes"`
- `Get-Counter -Counter $CounterList -SampleInterval 1 -MaxSamples 20`

Dealing With Localized Counter Names

- The most severe limitation of Get-Counter are the localized counter names
- There are two API functions you can use to convert localized counter names to id numbers and vice versa
 - Get-PerformanceCounterId takes a localized performance counter name and translates it to a language-agnostic id number
 - Get-PerformanceCounterLocalName does the opposite and translates the id number to the appropriate local name

<https://powershell.one/tricks/performance/performance-counters>

<https://powershellmagazine.com/2013/07/19/querying-performance-counters-from-powershell/>

Simload Base Counters – Simloads.ini

[Telemetry]

Name1=CPU|%

Counter1=\Processor(_Total)\% Processor Time

Name2=CPU Queue Length

Counter2=\System\Processor Queue Length

Name3=Memory Available|MBytes

Counter3=\Memory\Available MBytes

Name4=Working Set|Bytes

Counter4=\Process(_Total)\Working Set

Name5=Disk Reads|Bytes/sec

Counter5=\PhysicalDisk(_Total)\Disk Read Bytes/sec

Name6=Disk Writes|Bytes/sec

Counter6=\PhysicalDisk(_Total)\Disk Write Bytes/sec

Name7=Disk IOPS

Counter7=\PhysicalDisk(_Total)\Disk Transfers/sec

Name8=Disk Avg. Queue Length

Counter8=\PhysicalDisk(_Total)\Avg. Disk Queue Length

Name9=Context Switches/sec

Counter9=\System\Context Switches/sec

Name10=Processes

Counter10=\System\Processes

Autolt Code Base

Only single-instance counters

Language-Agnostic Counters

```
;$ _GVAR_TM_aPerfCounter[1][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter1", "\\Processor(_Total)\\% Processor Time")
$_GVAR_TM_aPerfCounter[1][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter1", "::-238\\6\\(_Total)")
$_GVAR_TM_aPerfCounter[1][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name1", "CPU|%")
;$ _GVAR_TM_aPerfCounter[2][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter2", "\\System\\Processor Queue Length")
$_GVAR_TM_aPerfCounter[2][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter2", "::-2\\44\\")
$_GVAR_TM_aPerfCounter[2][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name2", "CPU Queue Length")
;$ _GVAR_TM_aPerfCounter[3][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter3", "\\Memory\\Available MBytes")
$_GVAR_TM_aPerfCounter[3][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter3", "::-4\\1382\\")
$_GVAR_TM_aPerfCounter[3][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name3", "Memory Available|MBytes")
;$ _GVAR_TM_aPerfCounter[4][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter4", "\\Process(_Total)\\Working Set")
$_GVAR_TM_aPerfCounter[4][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter4", "::-230\\180\\(_Total)")
$_GVAR_TM_aPerfCounter[4][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name4", "Working Set|Bytes")
;$ _GVAR_TM_aPerfCounter[5][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter5", "\\PhysicalDisk(_Total)\\Disk Read Bytes/sec")
$_GVAR_TM_aPerfCounter[5][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter5", "::-234\\220\\(_Total)")
$_GVAR_TM_aPerfCounter[5][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name5", "Disk Reads|Bytes/sec")
;$ _GVAR_TM_aPerfCounter[6][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter6", "\\PhysicalDisk(_Total)\\Disk Write Bytes/sec")
$_GVAR_TM_aPerfCounter[6][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter6", "::-234\\222\\(_Total)")
$_GVAR_TM_aPerfCounter[6][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name6", "Disk Writes|Bytes/sec")
;$ _GVAR_TM_aPerfCounter[7][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter7", "\\PhysicalDisk(_Total)\\Disk Transfers/sec")
$_GVAR_TM_aPerfCounter[7][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter7", "::-234\\212\\(_Total)")
$_GVAR_TM_aPerfCounter[7][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name7", "Disk IOPS")
;$ _GVAR_TM_aPerfCounter[8][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter8", "\\PhysicalDisk(_Total)\\Avg. Disk Queue Length")
$_GVAR_TM_aPerfCounter[8][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter8", "::-234\\1400\\(_Total)")
$_GVAR_TM_aPerfCounter[8][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name8", "Disk Avg. Queue Length")
;$ _GVAR_TM_aPerfCounter[9][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter9", "\\System\\Context Switches/sec")
$_GVAR_TM_aPerfCounter[9][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter9", "::-2\\146\\")
$_GVAR_TM_aPerfCounter[9][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name9", "Context Switches/sec")
;$ _GVAR_TM_aPerfCounter[10][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter10", "\\System\\Processes")
$_GVAR_TM_aPerfCounter[10][0] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Counter10", "::-2\\248\\")
$_GVAR_TM_aPerfCounter[10][1] = IniRead($ _GVAR_TM_sPerfCounterSourcePath & "Simloads.ini", "Telemetry", "Name10", "Processes")
```

Telemetry Collector – TelemetryDataConfig.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<TelemetryDataConfig>
  <RefreshRate>1000</RefreshRate>
  <CounterDefinitions>
    <!-- Standard Counters -->
    <Counter Name="CPU" Unit="%">
      <CategoryName>Processor</CategoryName>
      <CounterName>% Processor Time</CounterName>
      <InstanceName>_Total</InstanceName>
    </Counter>
    <Counter Name="CPU Queue Length">
      <CategoryName>System</CategoryName>
      <CounterName>Processor Queue Length</CounterName>
    </Counter>
    <Counter Name="Memory Available" Unit="MBytes">
      <CategoryName>Memory</CategoryName>
      <CounterName>Available Mbytes</CounterName>
    </Counter>
  </CounterDefinitions>
</TelemetryDataConfig>
```

C++ Code Base
Single-instance counters
plus additional metrics

TC-specific Metrics

```
<Counter Name="Network Received" Unit="KBytes/sec">
  <CategoryName>TC::network received</CategoryName>
  <InstanceName>_Total</InstanceName>
</Counter>
<Counter Name="Network Sent" Unit="KBytes/sec">
  <CategoryName>TC::network sent</CategoryName>
  <InstanceName>_Total</InstanceName>
</Counter>
<Counter Name="GPU 3D" Unit="%">
  <CategoryName>TC::GPU load</CategoryName>
  <CounterName>3D</CounterName>
  <InstanceName>_Total</InstanceName>
</Counter>
<Counter Name="GPU Video Decode" Unit="%">
  <CategoryName>TC::GPU load</CategoryName>
  <CounterName>Video Decode</CounterName>
  <InstanceName>_Total</InstanceName>
</Counter>
```

```
<Counter Name="GPU Video Processing" Unit="%">
  <CategoryName>TC::GPU load</CategoryName>
  <CounterName>Video Processing</CounterName>
  <InstanceName>_Total</InstanceName>
</Counter>
<Counter Name="GPU Memory" Unit="MBytes">
  <CategoryName>TC::GPU frame buffer</CategoryName>
  <InstanceName>_Total</InstanceName>
</Counter>
<Counter Name="Session CPU" Unit="%">
  <CategoryName>Terminal Services Session</CategoryName>
  <CounterName>% Processor Time</CounterName>
  <InstanceName>TC::current RDP session</InstanceName>
</Counter>
```

CSV Result File

```
TimeStamp|1000,CPU|%,CPU Queue Length,Available Memory|MB,Working Set|Bytes,Disk|Writes/sec,Disk|Reads/sec,IOPS,Disk Queue Length,Context Switches,Processes
2023.10.29 22:13:34.192,10,0,5993,13830582272,7,8,16,0,15232,367
2023.10.29 22:13:35.197,9,0,5995,13776322560,41,1,43,0,15390,364
2023.10.29 22:13:36.187,8,0,6013,13709582336,43,2,45,0,20534,362
2023.10.29 22:13:37.192,10,0,6008,13709668352,32,0,33,0,13920,362
2023.10.29 22:13:38.181,9,0,6019,13700558848,33,1,34,0,13950,362
2023.10.29 22:13:39.185,8,0,6017,13701595136,33,0,34,0,15660,362
2023.10.29 22:13:40.190,11,0,6019,13704974336,80,1,82,0,17593,362
2023.10.29 22:13:41.194,10,0,6064,13619404800,46,0,47,0,13589,361
2023.10.29 22:13:42.198,10,0,6063,13616513024,112,1,114,0,12734,361
2023.10.29 22:13:43.203,9,0,6067,13614186496,21,0,21,0,13642,361
2023.10.29 22:13:44.208,10,0,6066,13613527040,28,4,33,0,12419,361
2023.10.29 22:13:45.198,9,0,6068,13613506560,29,2,31,0,12968,361
2023.10.29 22:13:46.188,8,0,6067,13612544000,22,1,23,0,13148,361
2023.10.29 22:13:47.193,10,0,6069,13612376064,27,0,28,0,14813,361
2023.10.29 22:13:48.197,10,0,6070,13613633536,26,0,27,0,14568,361
2023.10.29 22:13:49.202,9,0,6069,13613637632,24,0,25,0,12665,361
2023.10.29 22:13:50.192,11,1,6068,13614481408,10,1,11,0,24493,361
2023.10.29 22:13:51.182,11,0,6071,13613842432,42,3,45,0,19295,361
2023.10.29 22:13:52.187,6,0,6069,13614387200,38,1,40,0,20770,361
2023.10.29 22:13:53.177,10,0,6068,13615935488,31,1,32,0,17654,361
2023.10.29 22:13:54.169,8,0,6068,13617676288,22,33,55,0,15473,361
```

Normalized CSV Files

1. CPU|%
2. CPU Queue Length
3. Memory Available|MBytes
4. Working Set|Bytes
5. Disk Reads|Bytes/sec
6. Disk Writes|Bytes/sec
7. Disk IOPS
8. Disk Queue Length
9. Context Switches/sec
10. Processes
11. Network Received|KBytes/sec
12. Network Sent|KBytes/sec
13. GPU 3D|%
14. GPU Video Decode|%
15. GPU Video Processing|%
16. GPU Memory|Mbytes
17. Session CPU|%
18. Normalized Total CPU
19. Total Network Received|KBytes
20. Total Network Sent|KBytes

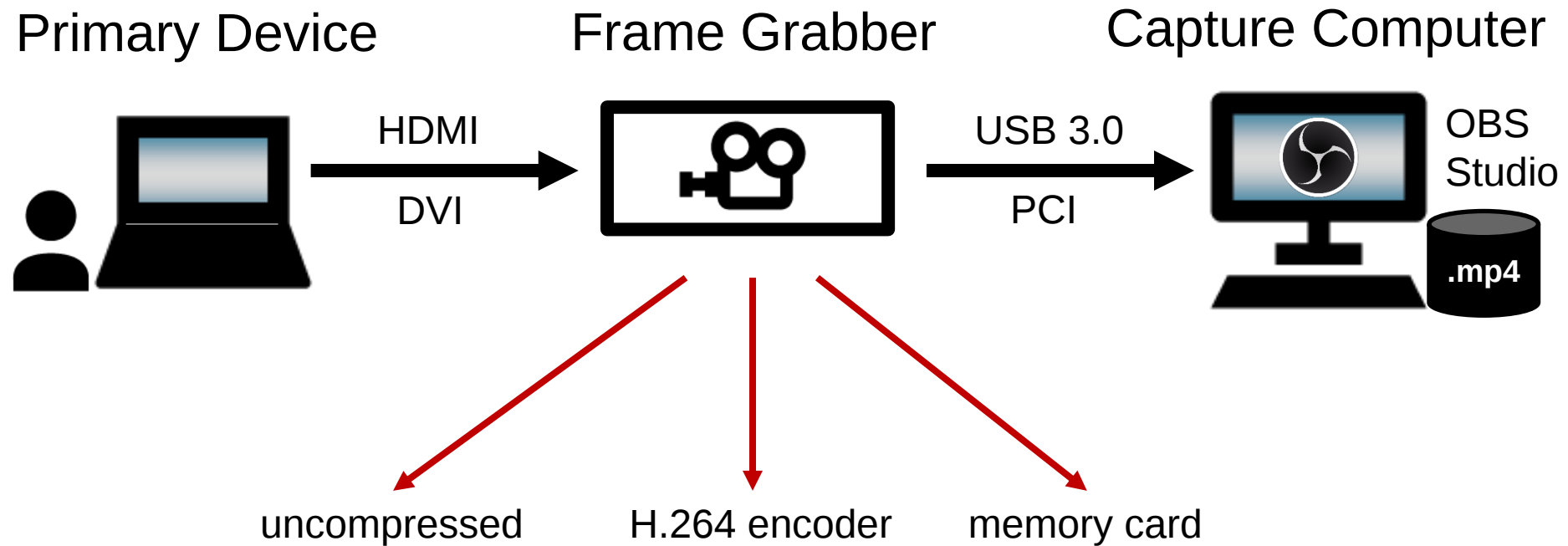
Normalized CSV File

TimeStamp|1000, CPU|%, CPU Queue Length, Memory Available|MBytes, Working Set|Bytes, Disk Reads|Bytes/sec, Disk Writes|Bytes/sec, Disk IOPS, Disk Queue Length, Context Switches/sec, Processes, Network Received|KBytes/sec, Network Sent|KBytes/sec, GPU 3D|%, GPU Video Decode|%, GPU Video Processing|%, GPU Memory|Mbytes, Session CPU|%, Total CPU, Total Network Received|KBytes, Total Network Sent|KBytes

```
2023.11.30 14:42:27.867, 0, 0, 25049, 8.546095E+09, 0, 0, 0, 0, 293, 3, 1, 1, 0, 0, 480, 0, 0, 3, 1
2023.11.30 14:42:28.930, 9.452645, 0, 25031, 8.54843E+09, 77401.96, 427572.4, 35.89294, 0, 16746.77, 292, 15, 15, 3, 0, 0, 489, 7.348182, 4.31040620354092, 18, 16
2023.11.30 14:42:29.948, 12.63744, 0, 25004, 8.600961E+09, 112572.1, 1012775, 50.07052, 0, 44503.12, 293, 9, 52, 22, 0, 0, 579, 10.37199, 10.0730786470981, 27, 68
2023.11.30 14:42:30.931, 23.30761, 0, 25004, 8.600756E+09, 33619.6, 247426.8, 16.41682, 0, 59352.43, 293, 12, 559, 95, 0, 0, 586, 15.45638, 20.7013484722269, 39, 627
2023.11.30 14:42:31.933, 20.83463, 0, 24984, 8.616894E+09, 0, 0, 0, 0, 68134.76, 293, 12, 296, 100, 0, 0, 694, 13.68174, 30.2019392038403, 51, 923
2023.11.30 14:42:32.950, 25.52207, 0, 24984, 8.62072E+09, 32069.18, 710522.2, 7.829204, 0, 80827.47, 293, 18, 937, 96, 0, 0, 694, 16.13209, 41.8400029463517, 69, 1860
2023.11.30 14:42:33.953, 24.508, 0, 24980, 8.622621E+09, 82247.13, 287811.1, 19.06979, 0, 78769.84, 293, 13, 490, 98, 0, 0, 695, 18.23511, 53.0156505417115, 82, 2350
2023.11.30 14:42:34.963, 21.04968, 0, 24991, 8.603976E+09, 16169.81, 379059.5, 29.6183, 0, 74899.3, 293, 12, 465, 99, 0, 0, 695, 15.37268, 62.6143048249508, 94, 2815
2023.11.30 14:42:35.975, 23.94374, 0, 24986, 8.614699E+09, 0, 0, 0, 0, 72524.41, 293, 19, 1253, 96, 0, 0, 695, 15.8974, 73.5326505131389, 113, 4068
2023.11.30 14:42:36.984, 25.70733, 0, 24982, 8.615223E+09, 16322.78, 151020.2, 9.966332, 0, 73982.41, 293, 22, 1356, 93, 0, 0, 695, 17.83603, 85.2551931669437, 135, 5424
2023.11.30 14:42:38.003, 19.58298, 0, 24986, 8.615268E+09, 15928.67, 578434.4, 103.062, 0, 74970.03, 293, 21, 1316, 94, 0, 0, 695, 13.11076, 94.1850315711243, 156, 6740
2023.11.30 14:42:39.000, 24.07301, 0, 24986, 8.616145E+09, 0, 0, 0, 0, 73834.09, 293, 20, 1434, 93, 0, 0, 695, 17.28562, 105.162323761258, 176, 8174
2023.11.30 14:42:39.998, 25.14586, 0, 24983, 8.617308E+09, 0, 280738, 11.0278, 0, 69740.45, 293, 18, 1331, 95, 0, 0, 695, 17.45483, 116.628836083803, 194, 9505
2023.11.30 14:42:41.002, 24.61835, 0, 24984, 8.618148E+09, 16300.58, 510286.5, 9.943838, 0, 69854.39, 293, 18, 1300, 93, 0, 0, 695, 16.59252, 127.854803121281, 212, 10805
2023.11.30 14:42:42.007, 21.01777, 0, 24983, 8.618775E+09, 16131.75, 482990.9, 13.7874, 0, 71900.82, 293, 18, 1294, 97, 0, 0, 695, 14.51063, 137.438906470865, 230, 12099
```

Recording Screen Videos with Frame Grabbers

Screen Recording: How it works...



Typical resolutions: Full HD (1080p) or 4k @ 30fps or 60fps

High-End Screen Capture

- **Epiphan AV.io 4K** (€580) or AV.io HD (€450) <http://www.epiphan.com>
- Epiphan DVI2PCIe (€1,400) or DVI2USB 3.0 (€800)
- **Elgato 4K X** (€250) <https://www.elgato.com/us/en/p/game-capture-4k-x>
- Elgato Game Capture 4K60 S+ (€400), including built-in HEVC/H.264 encoder
- Elgato Game Capture 4K60 Pro MK.2, 4K60FPS HDR Capture, PCIe Card
- Hauppauge HD PVR Pro 60 4K (€170), 1080p 60fps downscaler and built-in HEVC/H.264 encoder
- Hauppauge HD PVR 2 Gaming Edition Plus, Hardware H.264 Encoder
- Blackmagic Design Atem Mini (€350)
- Blackmagic Intensity Pro 4K PCIe Card (€200)
- Blackmagic Design DeckLink Quad HDMI Recorder 4 x 4K (€520)

Low-Cost Frame Grabbers

- Elgato Game Capture HD60 S or HD60 S+ (€140)
- AVerMedia Live Gamer Portable 2 Plus, 4K Pass-Through, 4K Full HD 1080p60 USB Game Capture, Stand-alone Mode using Memory Card (€130)
- TerraTec 130653 PVR Grabster Extreme HD Converter (€120)
- ClearClick HD Capture Box (€120)
- HDMI Video Capture Device USB 3.0 1080P 60 FPS Video & Audio Grabber
- MYPIN 1080P HD 60fps USB 3.0 Type C HDMI Game Video Capture Card
- MiraBox USB 3.0 HDMI Game Capture Karte 1080P 60fps Portable HD Video Recorder

<https://eucscore.com/equipment.html>

Elgato 4K X

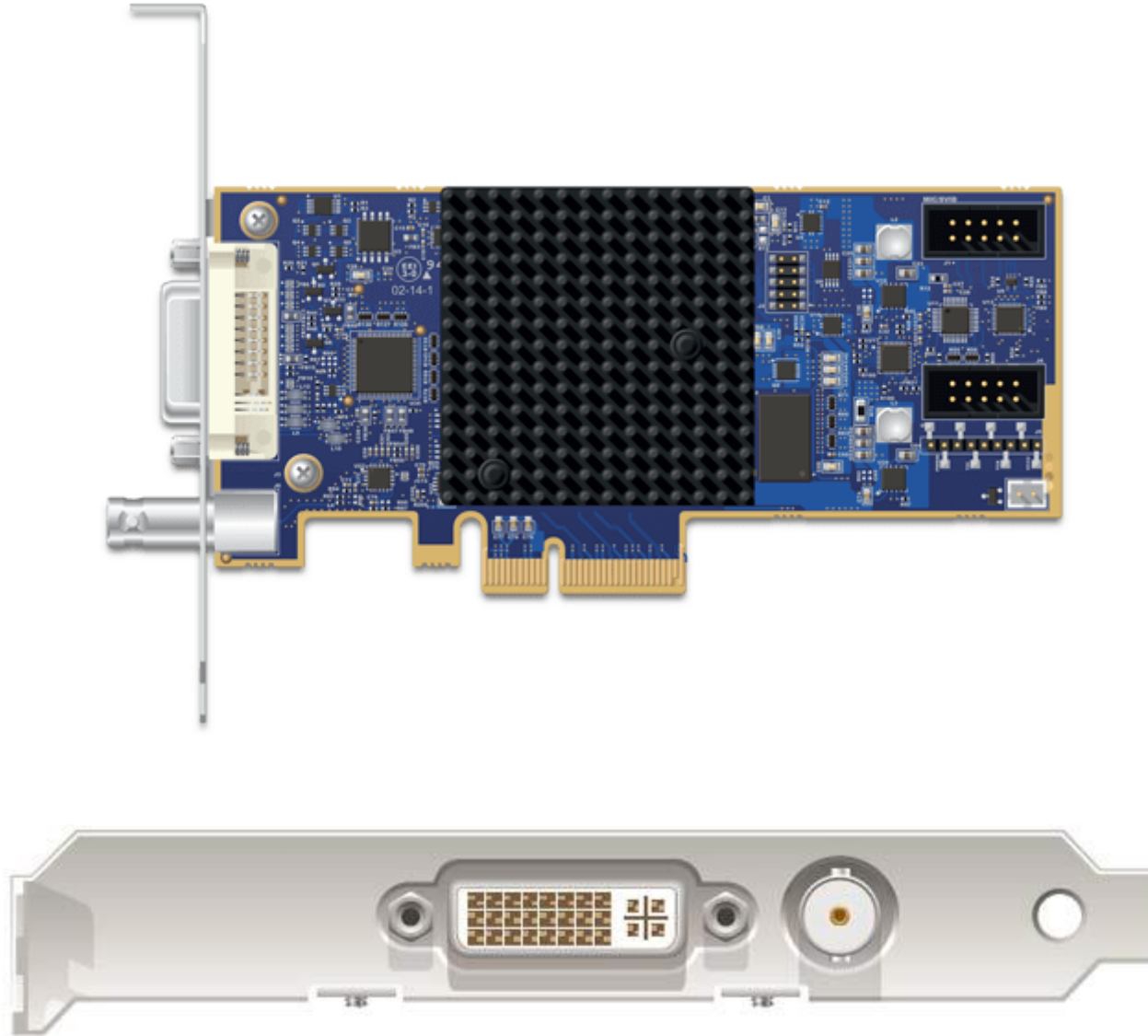


Epiphan AV.io

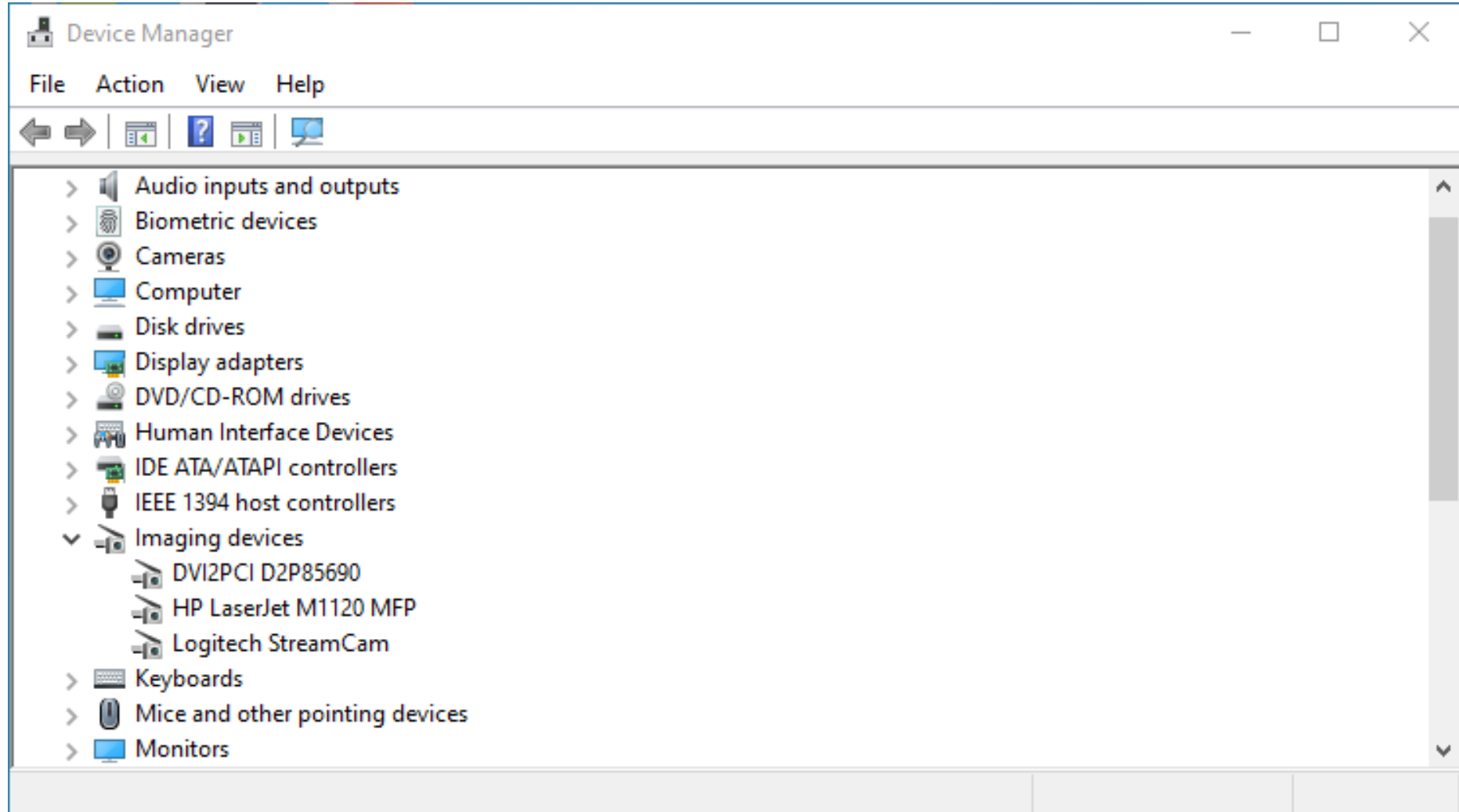


Uncompressed video capture

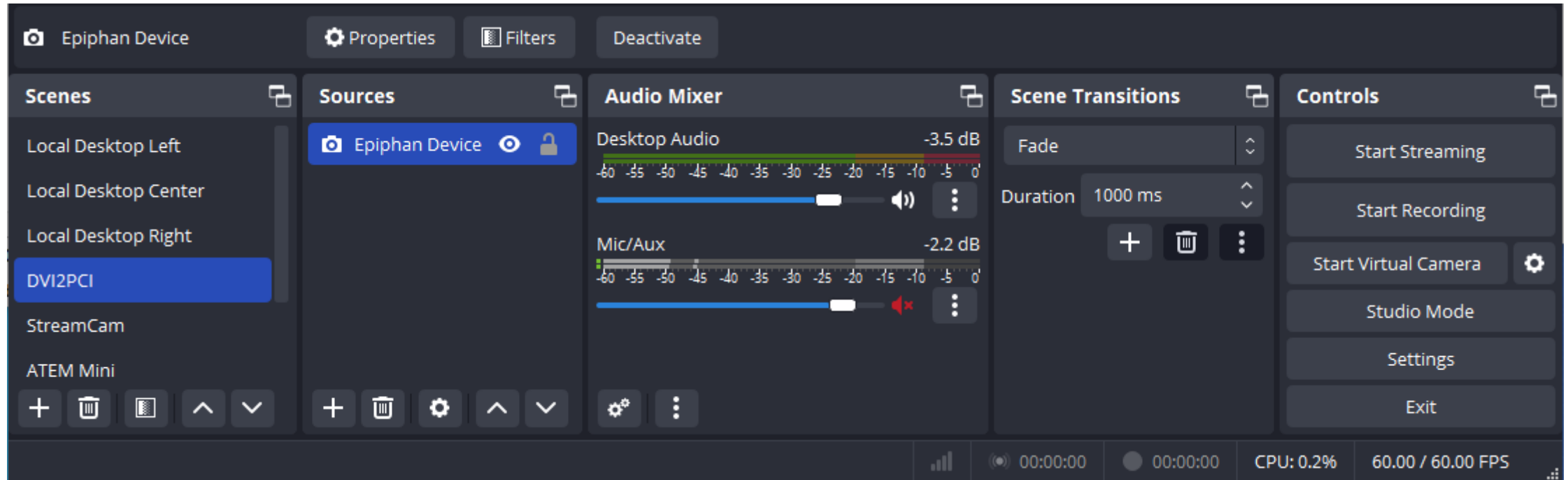
Epiphan DVI2PCI Duo



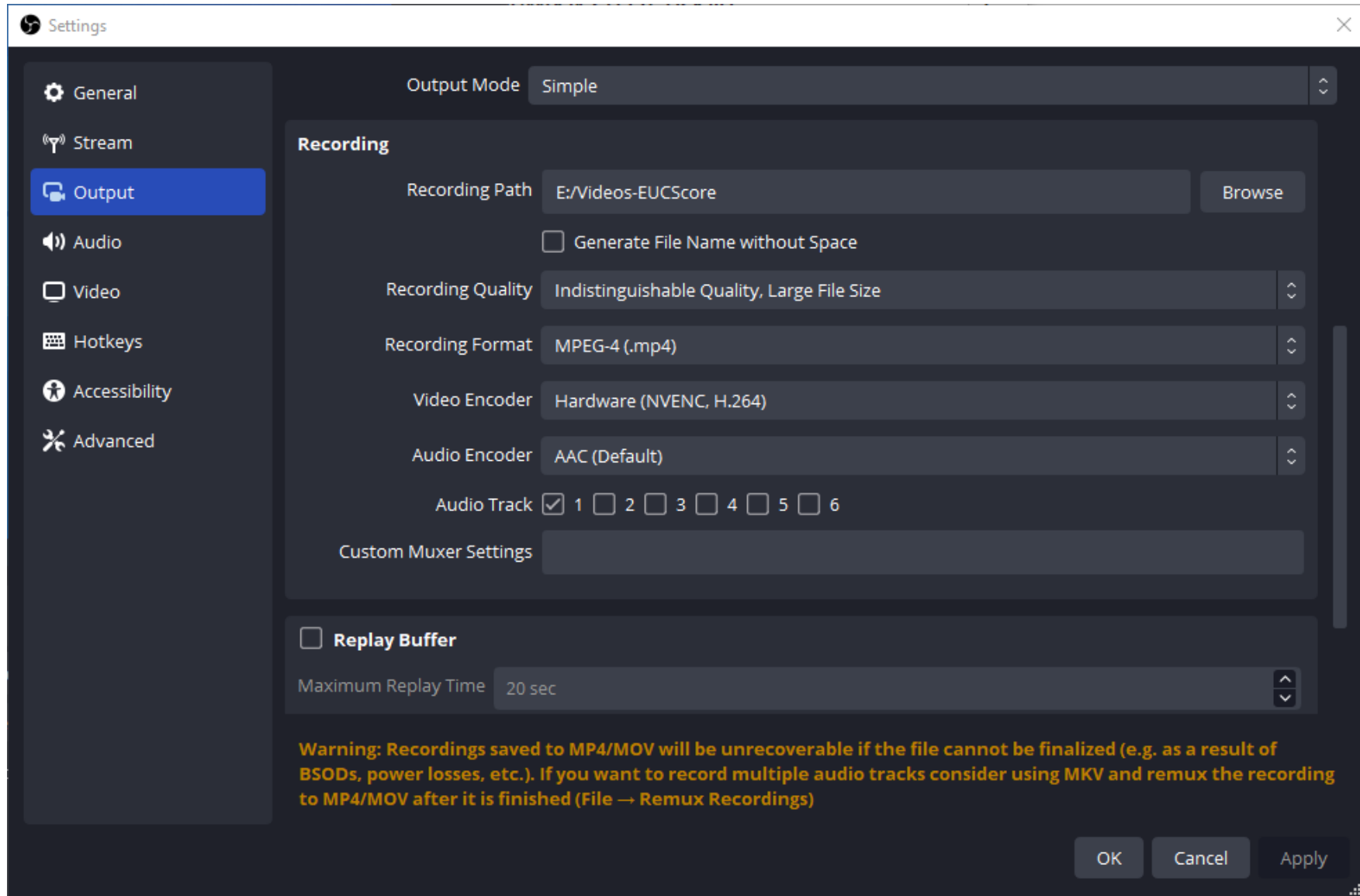
Epiphan DVI2PCI



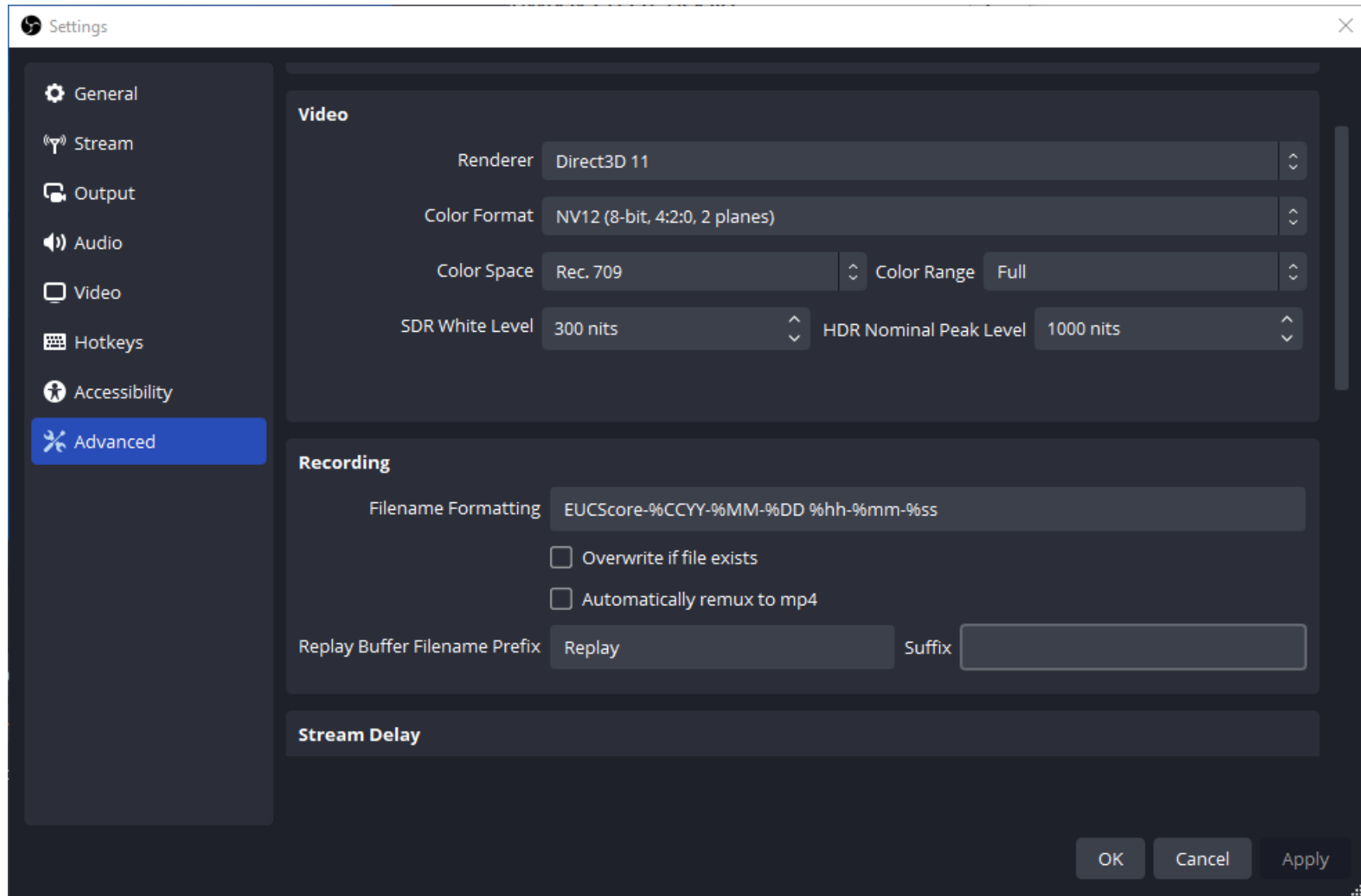
OBS Studio



OBS Studio



OBS Studio



Alternative Video Recorder: FFmpeg.org

 FFmpeg

About

News

Download

Documentation

Community

- Code of Conduct
- Mailing Lists
- IRC
- Forums
- Bug Reports
- Wiki

Developers

- Source Code
- Contribute
- FATE
- Code Coverage
- Funding through SPI

More

- Donate 
- Hire Developers
- Contact
- Security
- Legal

FFmpeg

A complete, cross-platform solution to record, convert and stream audio and video.

 **Download**

Converting video and audio has never been so easy.

```
$ ffmpeg -i input.mp4 output.avi
```

```
ffmpeg -t 45 -f gdigrab -framerate 60 -i desktop c:\tmp\recording.mp4
```

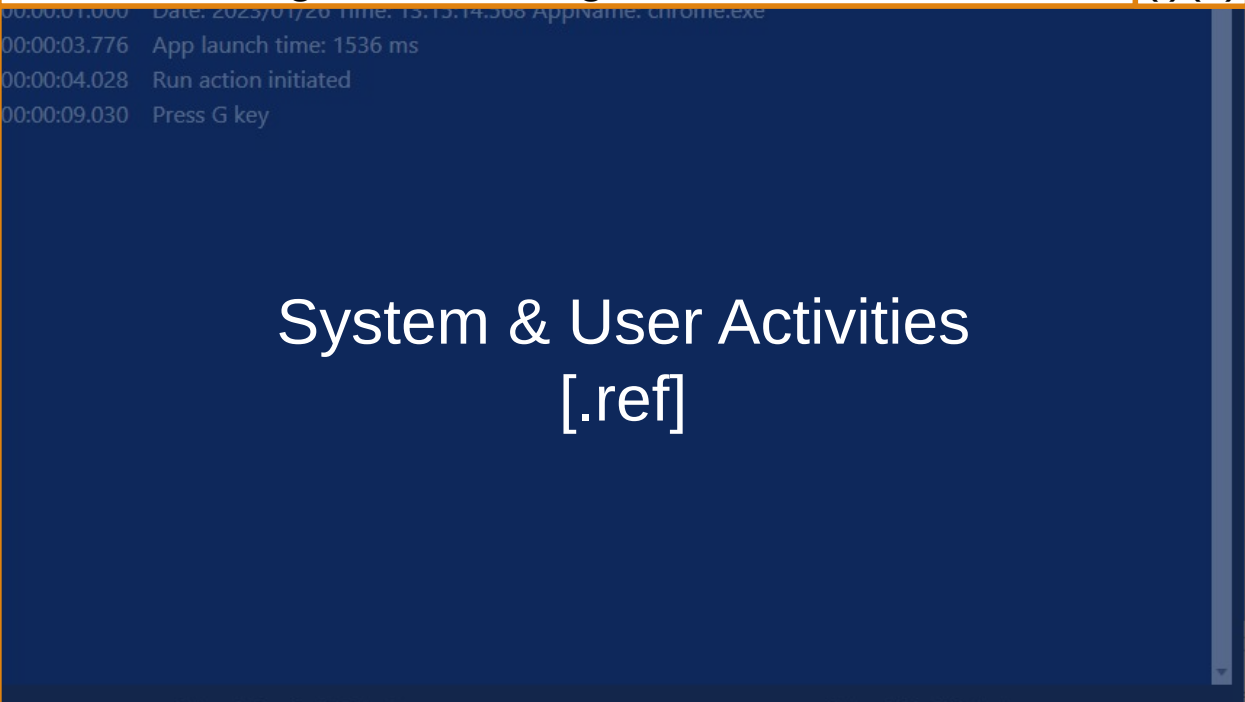
Discover more

Visualizing Data Using HTML5

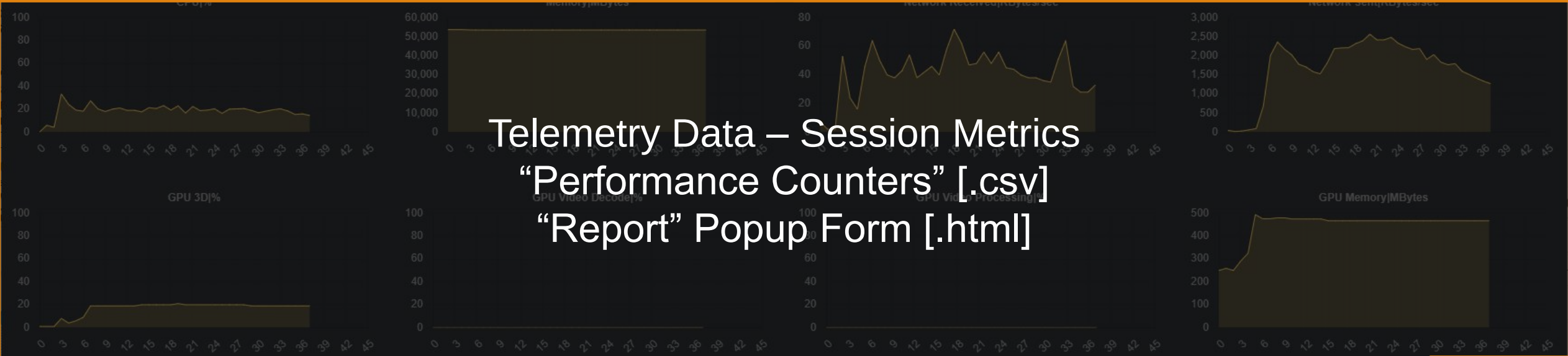
Left Media Tile
“Screen Recording”
[.mp4]

Description

System & User Activities
[.ref]



Telemetry Data – Session Metrics
“Performance Counters” [.csv]
“Report” Popup Form [.html]



Left Media Tile “Pacemaker” Video [.mp4]

Right Media Tile Comparison Video [.mp4]

Color-Coded Telemetry Data – 2 x Session Metrics “Performance Counters” Overlay [.csv] “Report” Popup Form [.html]

Left Media Tile
“Pacemaker” Video
[.mp4]

Right Media Tile
Comparison Video
[.mp4]

Left System & User Activities
[.ref]

Right System & User Activities
[.ref]

HTML Templates

- Template-VidRef-8Charts.html → 1 video + 1 activity + 8 charts
- Template-VidRef-12Charts.html → 1 video + 1 activity + 12 charts
- Template-VidVid-8Charts.html → 2 side-by-side videos + 8 charts
- Template-VidVid-12Charts.html → 2 side-by-side videos + 12 charts
- Template-VidVid-RefRef.html → 2 side-by-side videos + 2 activities
- Template-Report.html → overlay of the lower half of the page on button click
- Template-LinkList.html → list of links to Sync Player clips in a test sequence
- Template-Index.html → all links to Sync Player clips in a run collection

HTML Templates

<!-- Style Sheets -->

<link href="../../_libs3/bootstrap.min.css" rel="stylesheet">

<link href="../../_libs3/custom.css" rel="stylesheet">

<!-- Load Bootstrap, Chart and JQuery JS files -->

<script src="../../_libs3/bootstrap.bundle.min.js"></script>

<script src="../../_libs3/chart.min.js"></script>

<script src="../../_libs3/jquery.min.js"></script>

New! Never-Ending Support for Bootstrap →



Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

`$ npm i bootstrap@5.3.3` 

 [Read the docs](#)

Currently **v5.3.3** · [Download](#) · [All releases](#)

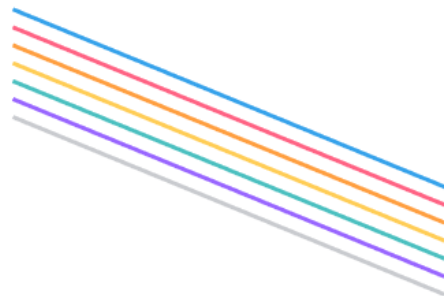


Implement Auth0 in any app in just 5 minutes. Start building today.

ads via Carbon

Simple yet flexible JavaScript charting library for the modern web

GitHub

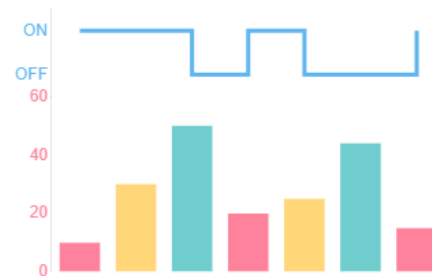


New in 4.0 Colors plugin

[Default palette](#) of Chart.js brand colors is available as a built-in time-saving zero-configuration plugin.

New in 4.0 Tree-shaking

JavaScript bundle size can be reduced by dozens of kilobytes by registering only **necessary components**.



New in 3.5 Scale stacking

Layout boxes can be stacked and weighted in groups.

New in 3.4 Subtitle plugin

A secondary title `pluin` with all the same options





Lightweight Footprint
Only 30kB minified and gzipped. Can also be included as an AMD module



CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation



Cross-Browser
[Chrome](#), [Edge](#), [Firefox](#), [IE](#), [Safari](#), [Android](#), [iOS](#), and more



Download jQuery
v3.7.1

The 1.x and 2.x branches no longer receive patches.

[View Source on GitHub →](#)
[How jQuery Works →](#)

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)
- [Browse or Submit jQuery Bugs](#)

A Brief Look

DOM Traversal and Manipulation

Get the `<button>` element with the class 'continue' and change its HTML to 'Next Step...'

```
1 | $( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the `#banner-message` element that is hidden with `display:none` in its CSS when any button in `#button-container` is clicked.

Custom JS Files

```
<!-- Load custom JS files -->
```

```
<script src="../../_libs3/charts.js"></script>
```

```
<script src="../../_libs3/player.js"></script>
```

- Charts.js defines the standard values of the charts
- Player.js implements an HTML5 video player
- Placeholders in the HTML templates
- PowerShell scripts to import data and replace placeholders

PowerShell Build Scripts

1. Create-SimloadFolders.ps1 → create folder structure from raw data files
2. Create-IniSingle.ps1 → Create an .ini used by consecutive build scripts
3. Optional: Convert-CsvFileByIni.ps1 → Replace decimal separator “,” to “.”
4. Create-CsvNormalized.ps1 → Create normalized .csv files
5. Create-CsvSingle.ps1 → Create .csv files according to Telemetry settings
6. Create-HtmlSingle.ps1 → Create single-view Sync Player clips
7. Create-IniDouble.ps1 → Create an .ini used by consecutive build scripts
8. Create-HtmlDouble.ps1 → Create side-by-side Sync Player clips
9. Create-Index.ps1 → Create a start page with links to all Sync Player clips

Sync Player

<https://eucscore.com/docs/syncplayer.html>

Call to Action

If you want to learn more about
EUC Score, send me an email

info@eucscore.com



<https://eucscore.com>

NOTE: The EUC Score toolset is free for
community benchmarking tests when the
results are made freely available to the public



Thank You

Benny Tritsch | info@eucscore.com | [@drtritsch](https://twitter.com/drtritsch)
