EXPERTS 2 EXPERTS Virtualization Conference
www.e2evc.com

EUC Score

Deep Dive: Collecting, analyzing and understanding Windows performance counters

E2EVC 2023, Rome

Benny Tritsch | info@drtritsch.com | @drtritsch

Benny Tritsch
Dr. Tritsch IT Consulting

EUC Score

control up

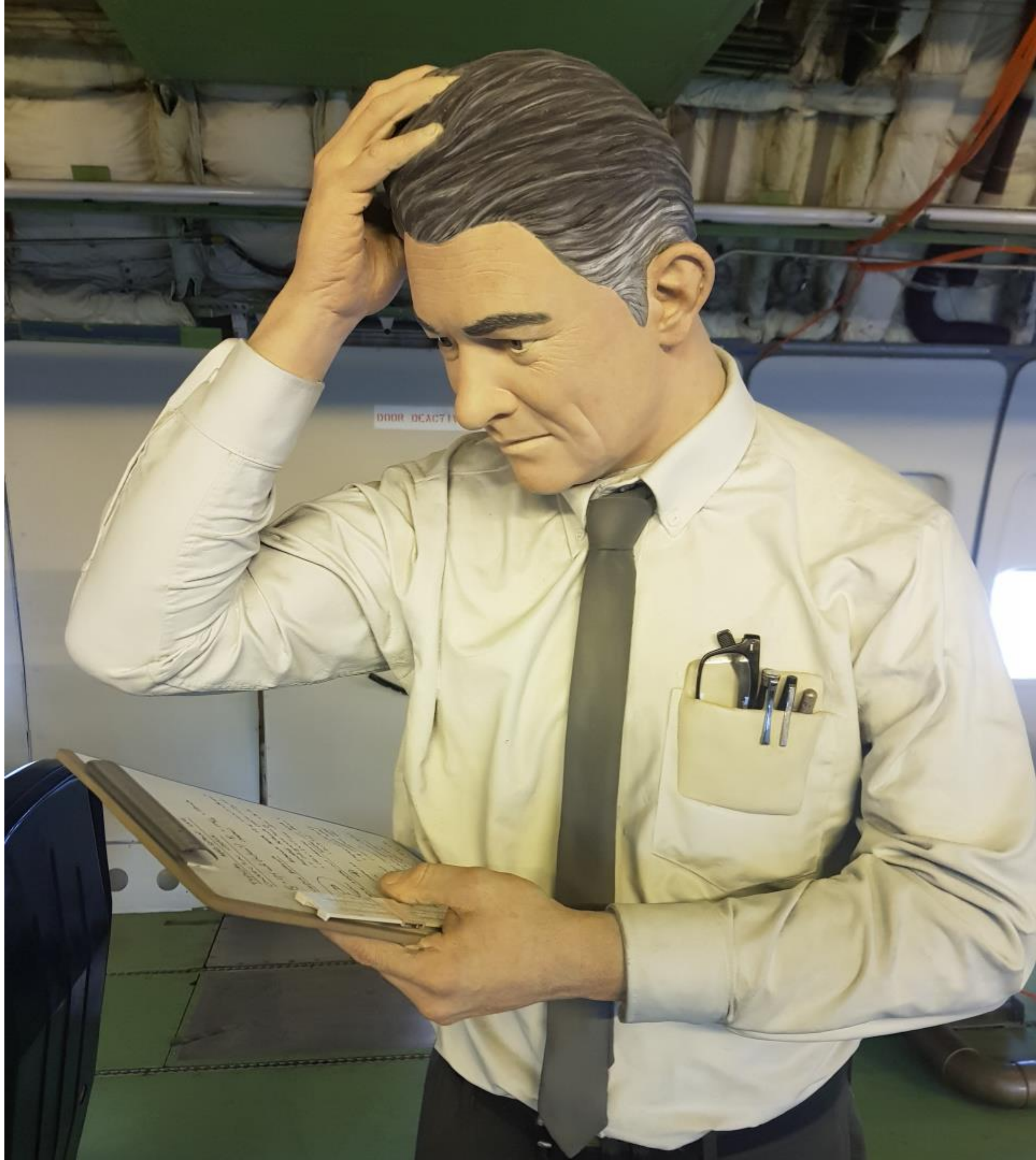
GO-EUC

VC 2 GO

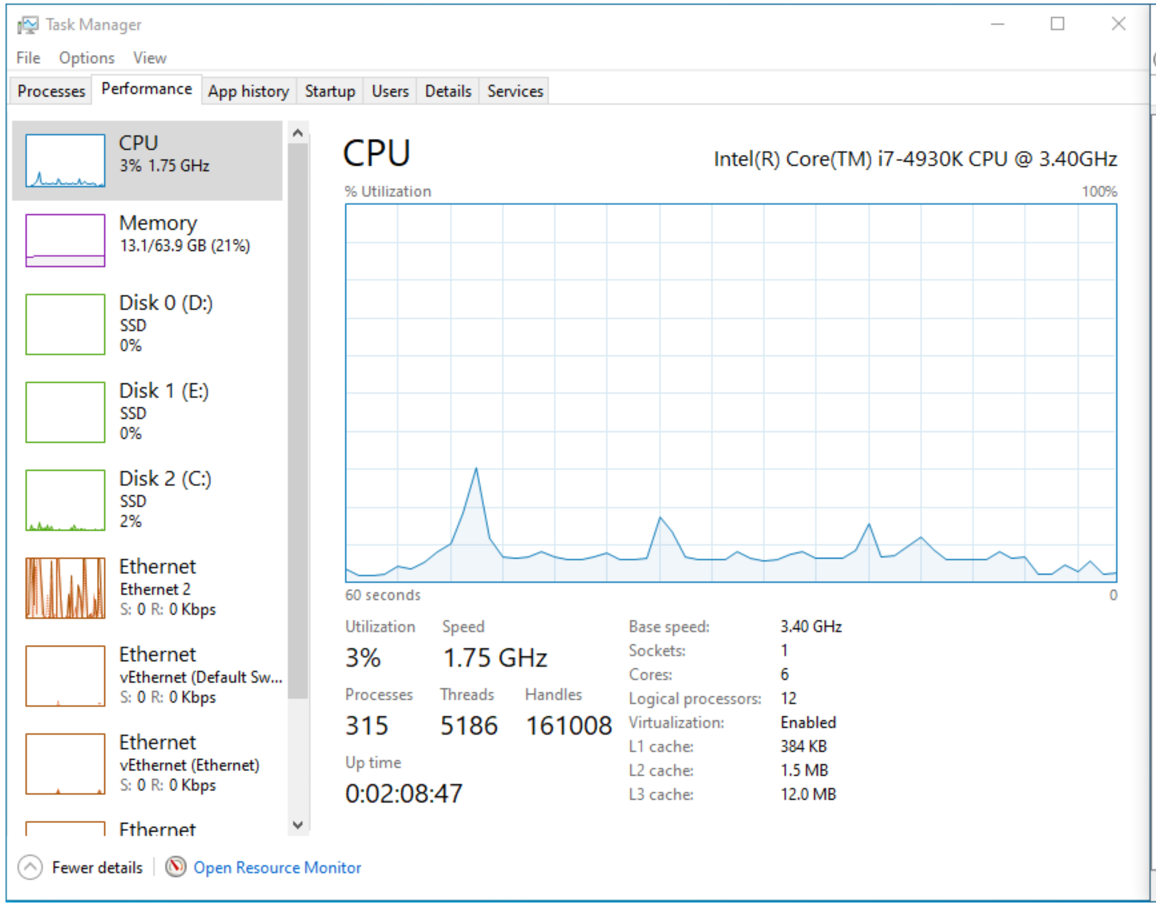
Performance Data Scientist
EUC Documentary Cameraman
MVP | CTP | vExpert EUC
NGCA | VIPP

info@drtritsch.com
@drtritsch

EXPERTS 2 EXPERTS Virtualization Conference
www.e2evc.com



Task Manager versus Performance Monitor

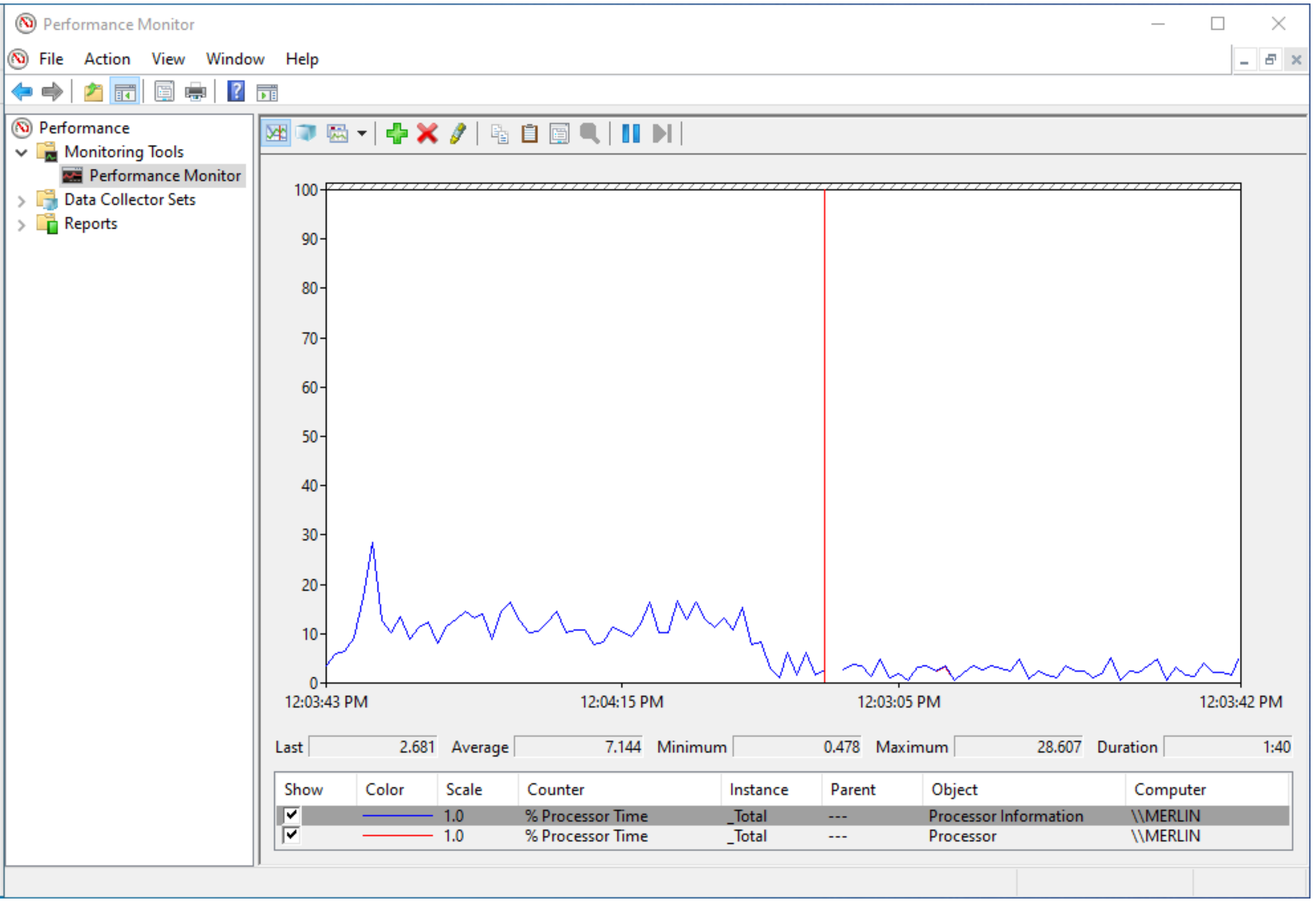

Task Manager
File Options View
Processes Performance App history Startup Users Details Services
CPU 3% 1.75 GHz
Memory 13.1/63.9 GB (21%)
Disk 0 (D:) SSD 0%
Disk 1 (E:) SSD 0%
Disk 2 (C:) SSD 2%
Ethernet Ethernet 2 S: 0 R: 0 Kbps
Ethernet vEthernet (Default Sw... S: 0 R: 0 Kbps
Ethernet vEthernet (Ethernet) S: 0 R: 0 Kbps
Ethernet
% Utilization
100%
60 seconds
0
Fewer details
Open Resource Monitor

CPU Intel(R) Core(TM) i7-4930K CPU @ 3.40GHz

Utilization	Speed	
3%	1.75 GHz	
Processes	**Threads**	**Handles**
315	5186	161008

Up time
0:02:08:47

Base speed:	3.40 GHz
Sockets:	1
Cores:	6
Logical processors:	12
Virtualization:	Enabled
L1 cache:	384 KB
L2 cache:	1.5 MB
L3 cache:	12.0 MB



Last 2.681 Average 7.144 Minimum 0.478 Maximum 28.607 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟦	1.0	% Processor Time	_Total	---	Processor Information	\\MERLIN
✓	🟥	1.0	% Processor Time	_Total	---	Processor	\\MERLIN

Windows 10 Task Manager '% CPU' skew – A Tale of Two Metrics by Jeff Stokes
https://illuminati.services/2021/03/17/windows-10-task-manager-cpu-inaccurate-a-tale-of-two-metrics/
Task Manager's CPU numbers are all but meaningless by Aaron Margosis
https://aaron-margosis.medium.com/task-managers-cpu-numbers-are-all-but-meaningless-2d165b421e43

Windows Performance Counters

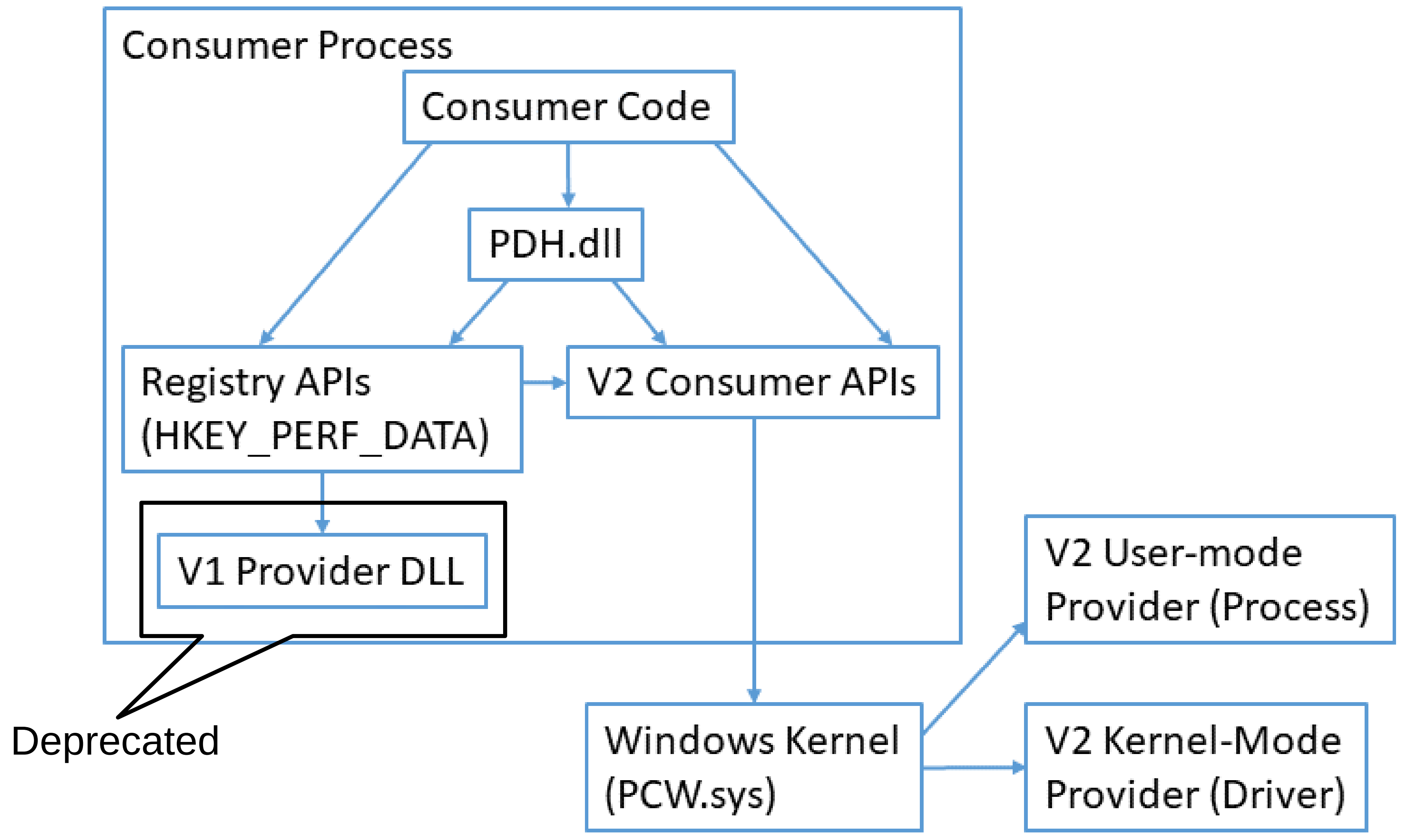
Consistent interface for collecting various kinds of system data

- 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

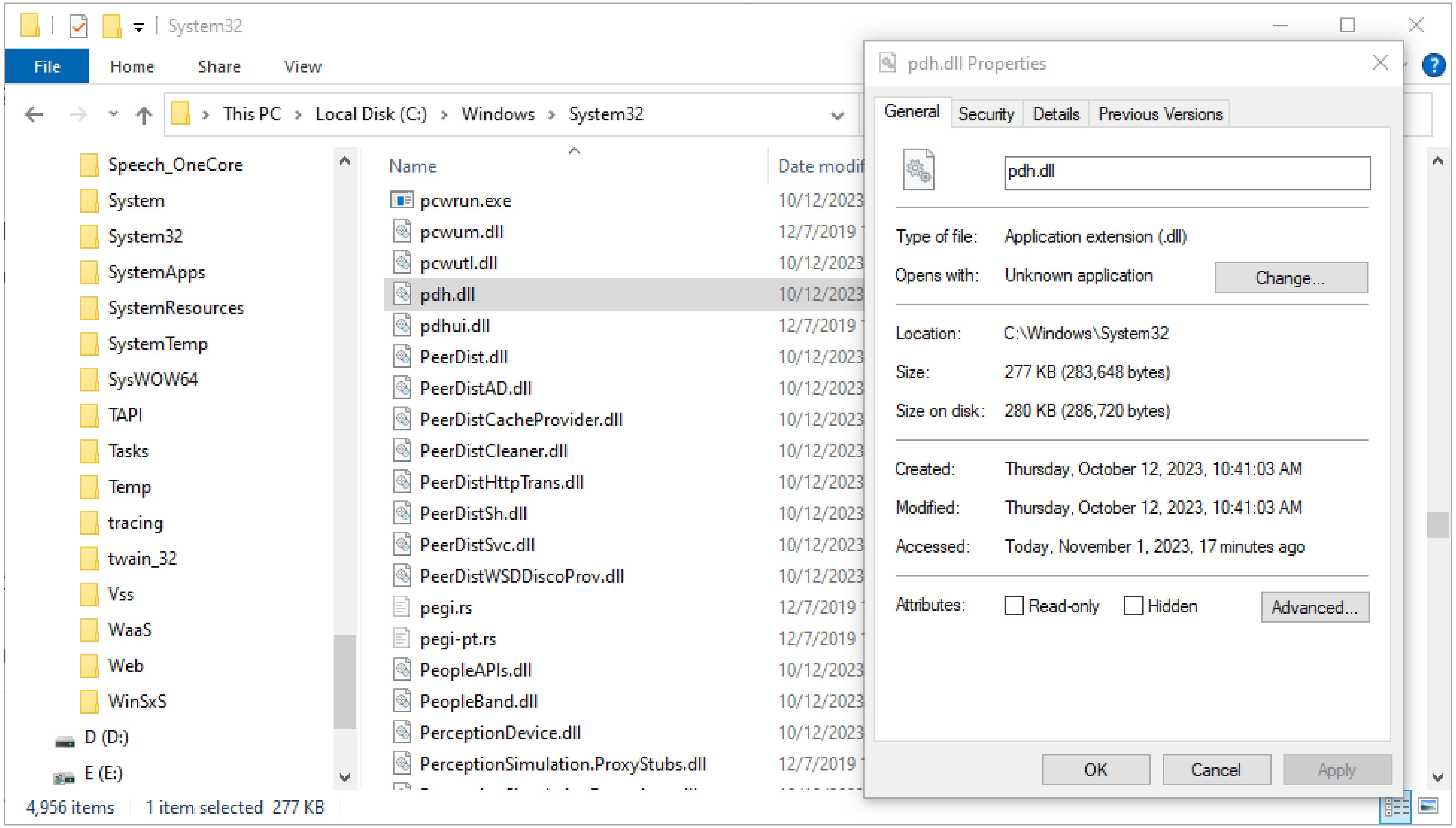
Windows Performance Counters

- **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



System32

File Home Share View

This PC › Local Disk (C:) › Windows › System32

Speech_OneCore
System
System32
SystemApps
SystemResources
SystemTemp
SysWOW64
TAPI
Tasks
Temp
tracing
twain_32
Vss
WaaS
Web
WinSxS
D (D:)
E (E:)

Name	Date modif
pcwrun.exe	10/12/2023
pcwum.dll	12/7/2019
pcwutl.dll	10/12/2023
pdh.dll	10/12/2023
pdhui.dll	12/7/2019
PeerDist.dll	10/12/2023
PeerDistAD.dll	10/12/2023
PeerDistCacheProvider.dll	10/12/2023
PeerDistCleaner.dll	10/12/2023
PeerDistHttpTrans.dll	10/12/2023
PeerDistSh.dll	10/12/2023
PeerDistSvc.dll	10/12/2023
PeerDistWSDDiscoProv.dll	10/12/2023
pegi.rs	12/7/2019
pegi-pt.rs	12/7/2019
PeopleAPIs.dll	10/12/2023
PeopleBand.dll	10/12/2023
PerceptionDevice.dll	10/12/2023
PerceptionSimulation.ProxyStubs.dll	12/7/2019

4,956 items 1 item selected 277 KB

pdh.dll Properties

General Security Details Previous Versions

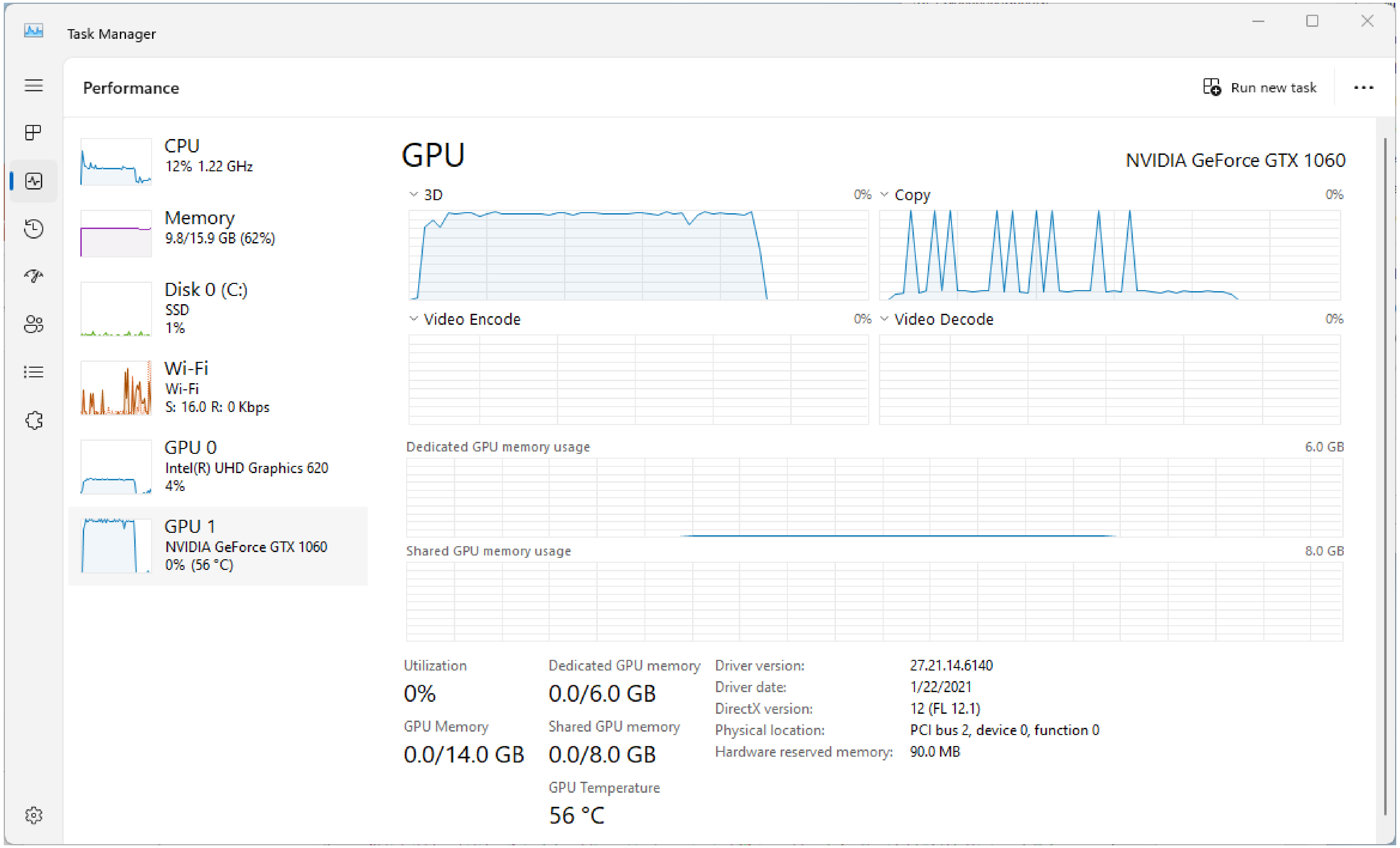
pdh.dll

Type of file:	Application extension (.dll)
Opens with:	Unknown application
Location:	C:\Windows\System32
Size:	277 KB (283,648 bytes)
Size on disk:	280 KB (286,720 bytes)
Created:	Thursday, October 12, 2023, 10:41:03 AM
Modified:	Thursday, October 12, 2023, 10:41:03 AM
Accessed:	Today, November 1, 2023, 17 minutes ago
Attributes:	☐ Read-only ☐ Hidden

HKEY_PERFORMANCE_DATA & PDH API

- This key provides runtime information into performance data provided by either the NT kernel itself, or running system drivers, programs and services that provide performance data
- This key is not stored in any hive and **not displayed in the Registry Editor**, but it is visible through the registry functions in the Windows API, or in a simplified view via the **Performance tab of the Task Manager**
- For details about the API, check out https://learn.microsoft.com/en-us/windows/win32/api/pdh/

Task Manager – Performance Tab



Task Manager

Performance

Run new task

CPU
12% 1.22 GHz

Memory
9.8/15.9 GB (62%)

Disk 0 (C:)
SSD
1%

Wi-Fi
Wi-Fi
S: 16.0 R: 0 Kbps

GPU 0
Intel(R) UHD Graphics 620
4%

GPU 1
NVIDIA GeForce GTX 1060
0% (56 °C)

GPU

NVIDIA GeForce GTX 1060

3D 0%

Copy 0%

Video Encode 0%

Video Decode 0%

Dedicated GPU memory usage 6.0 GB

Shared GPU memory usage 8.0 GB

Utilization
0%

Dedicated GPU memory
0.0/6.0 GB

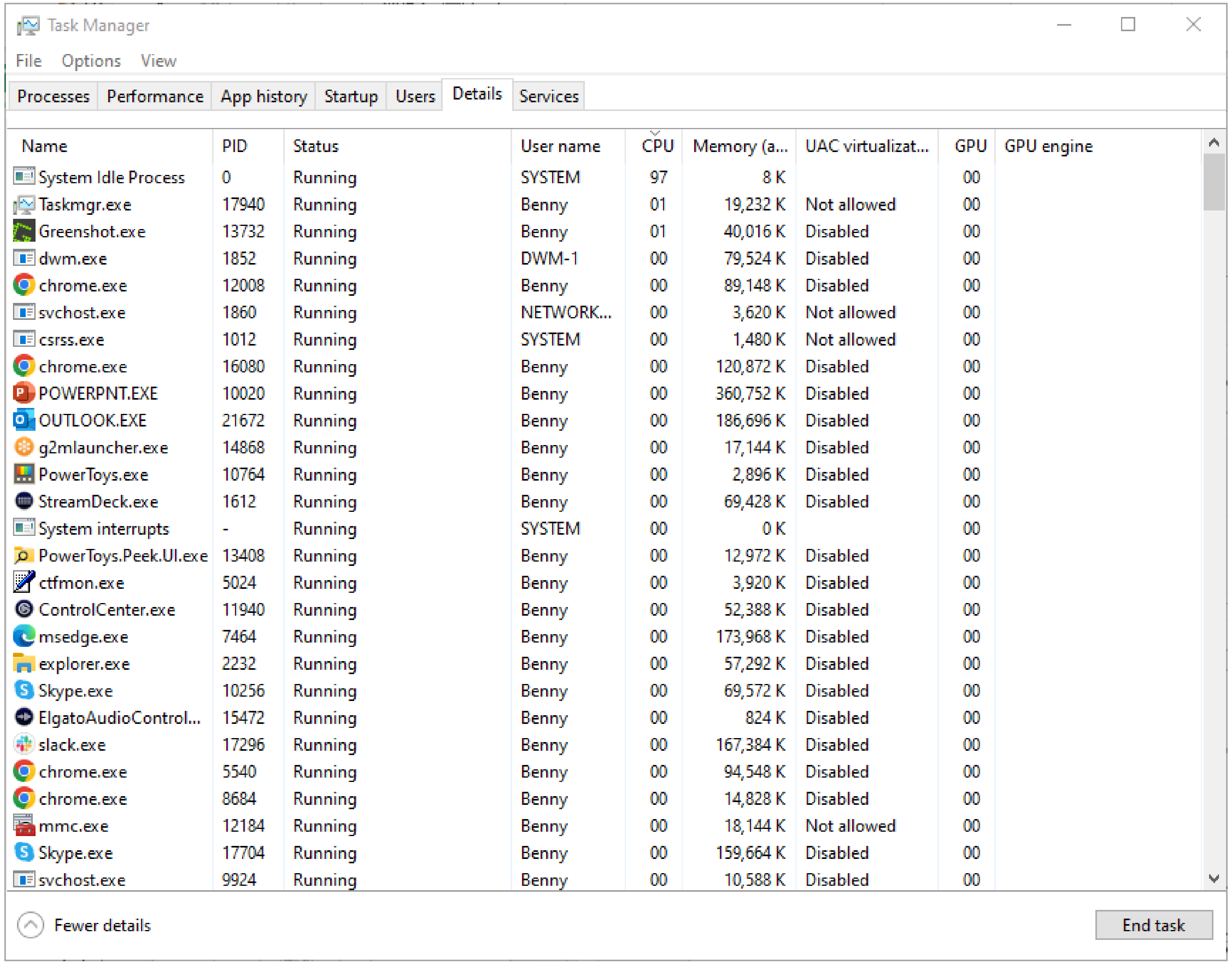
GPU Memory
0.0/14.0 GB

Shared GPU memory
0.0/8.0 GB

GPU Temperature
56 °C

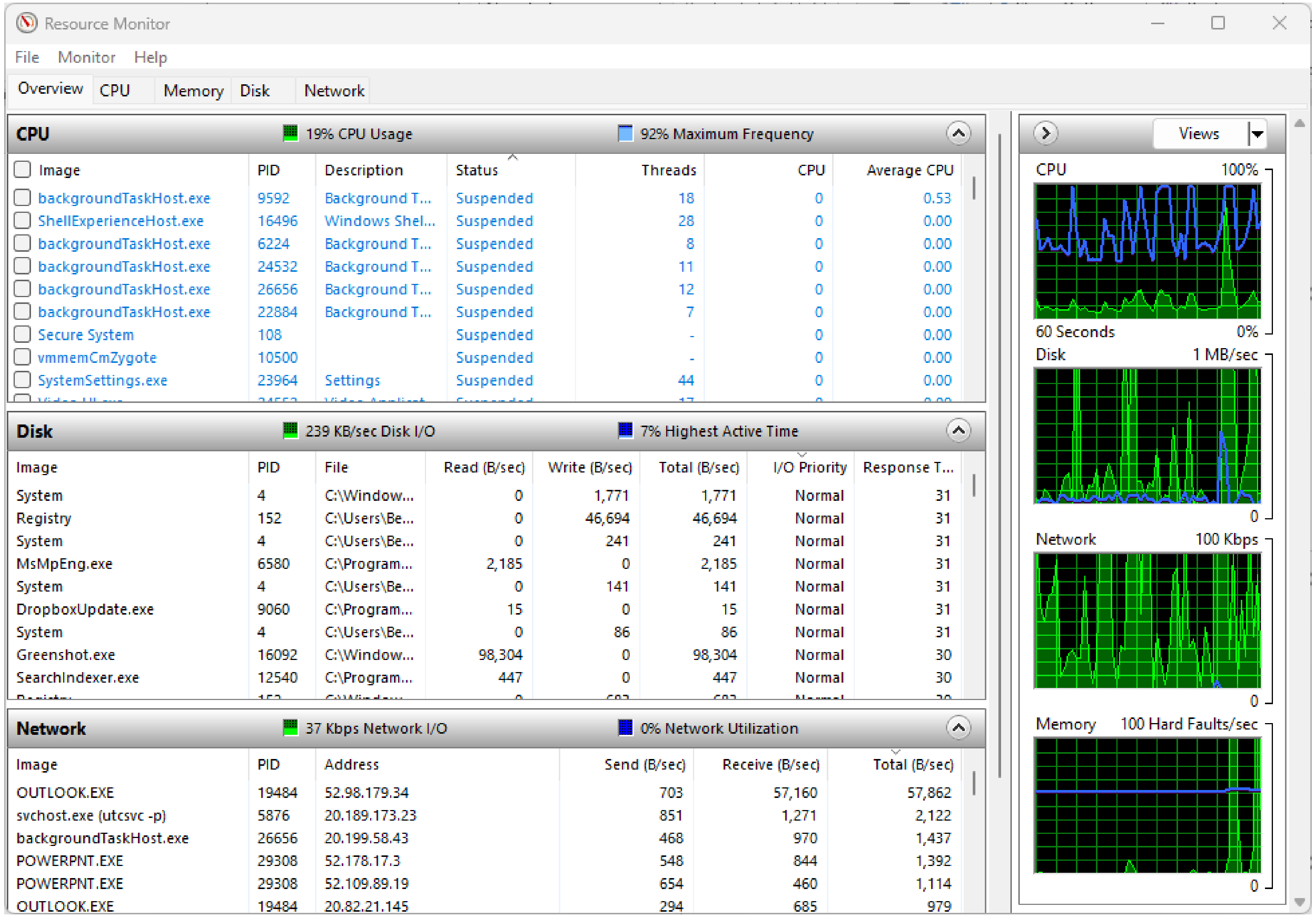
Driver version:	27.21.14.6140
Driver date:	1/22/2021
DirectX version:	12 (FL 12.1)
Physical location:	PCI bus 2, device 0, function 0
Hardware reserved memory:	90.0 MB

Task Manager – Detail Tab


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

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



CPU 🟩 19% CPU Usage 🟦 92% Maximum Frequency

Image	PID	Description	Status	Threads	CPU	Average CPU
backgroundTaskHost.exe	9592	Background T...	Suspended	18	0	0.53
ShellExperienceHost.exe	16496	Windows Shel...	Suspended	28	0	0.00
backgroundTaskHost.exe	6224	Background T...	Suspended	8	0	0.00
backgroundTaskHost.exe	24532	Background T...	Suspended	11	0	0.00
backgroundTaskHost.exe	26656	Background T...	Suspended	12	0	0.00
backgroundTaskHost.exe	22884	Background T...	Suspended	7	0	0.00
Secure System	108		Suspended	-	0	0.00
vmmemCmZygote	10500		Suspended	-	0	0.00
SystemSettings.exe	23964	Settings	Suspended	44	0	0.00

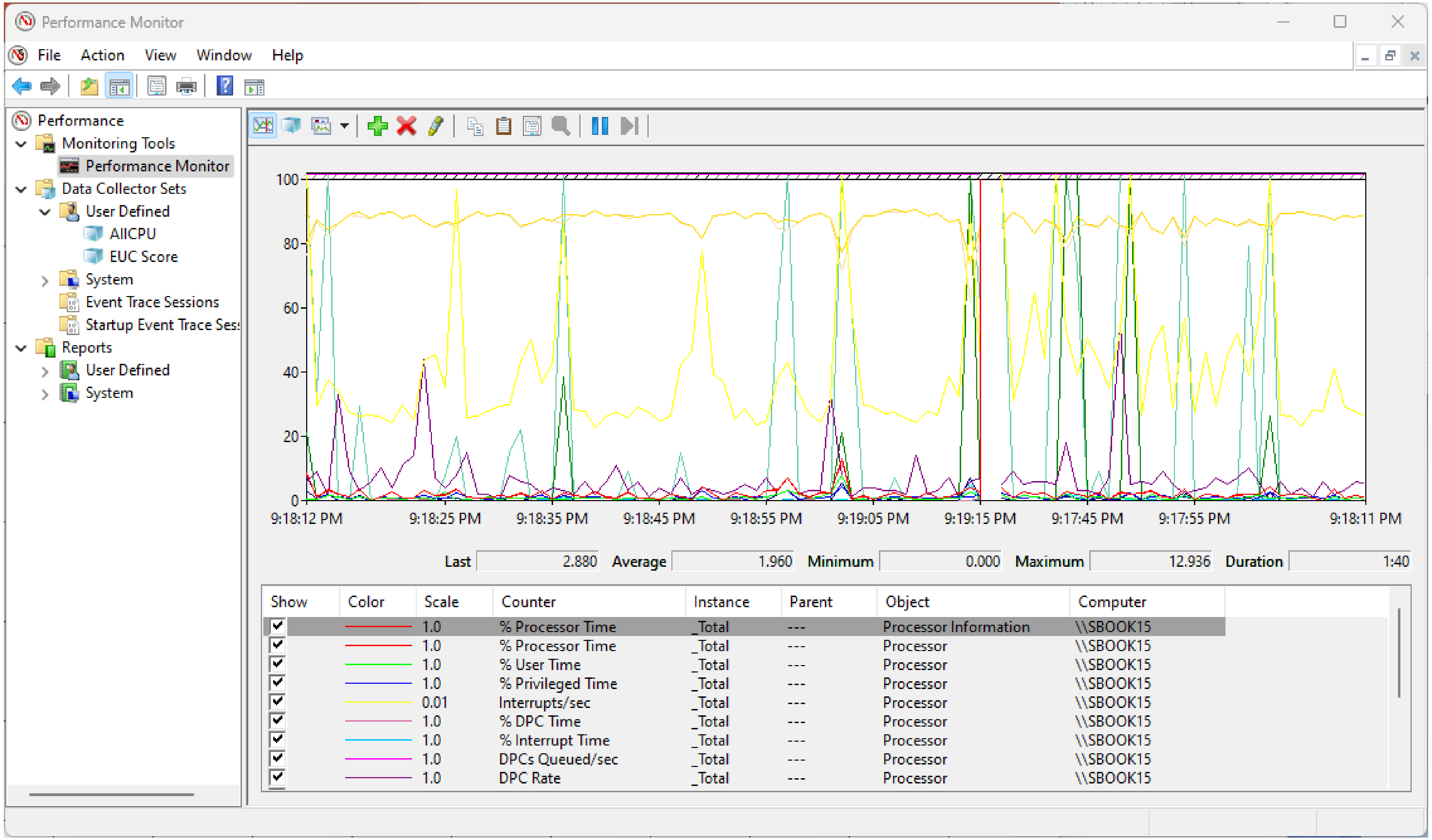
Disk 🟩 239 KB/sec Disk I/O 🟦 7% Highest Active Time

Image	PID	File	Read (B/sec)	Write (B/sec)	Total (B/sec)	I/O Priority	Response T...
System	4	C:\Window...	0	1,771	1,771	Normal	31
Registry	152	C:\Users\Be...	0	46,694	46,694	Normal	31
System	4	C:\Users\Be...	0	241	241	Normal	31
MsMpEng.exe	6580	C:\Program...	2,185	0	2,185	Normal	31
System	4	C:\Users\Be...	0	141	141	Normal	31
DropboxUpdate.exe	9060	C:\Program...	15	0	15	Normal	31
System	4	C:\Users\Be...	0	86	86	Normal	31
Greenshot.exe	16092	C:\Window...	98,304	0	98,304	Normal	30
SearchIndexer.exe	12540	C:\Program...	447	0	447	Normal	30

Network 🟩 37 Kbps Network I/O 🟦 0% Network Utilization

Image	PID	Address	Send (B/sec)	Receive (B/sec)	Total (B/sec)
OUTLOOK.EXE	19484	52.98.179.34	703	57,160	57,862
svchost.exe (utcsvc -p)	5876	20.189.173.23	851	1,271	2,122
backgroundTaskHost.exe	26656	20.199.58.43	468	970	1,437
POWERPNT.EXE	29308	52.178.17.3	548	844	1,392
POWERPNT.EXE	29308	52.109.89.19	654	460	1,114
OUTLOOK.EXE	19484	20.82.21.145	294	685	979

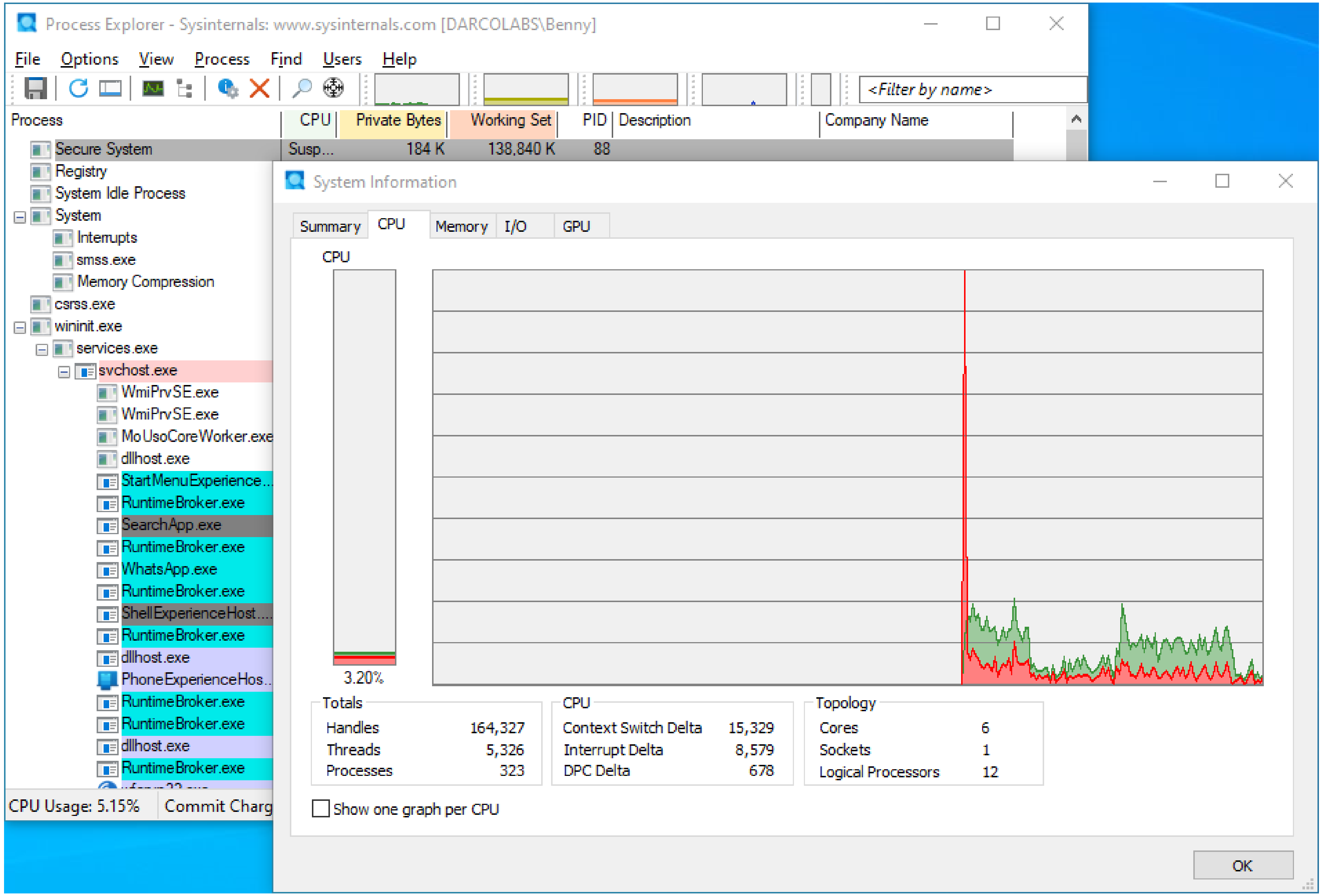
Performance Monitor



Last 2.880 Average 1.960 Minimum 0.000 Maximum 12.936 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Processor Time	_Total	---	Processor Information	\\SBOOK15
✓	🟥	1.0	% Processor Time	_Total	---	Processor	\\SBOOK15
✓	🟩	1.0	% User Time	_Total	---	Processor	\\SBOOK15
✓	🟦	1.0	% Privileged Time	_Total	---	Processor	\\SBOOK15
✓	🟨	0.01	Interrupts/sec	_Total	---	Processor	\\SBOOK15
✓	🩷	1.0	% DPC Time	_Total	---	Processor	\\SBOOK15
✓	🩵	1.0	% Interrupt Time	_Total	---	Processor	\\SBOOK15
✓	🟪	1.0	DPCs Queued/sec	_Total	---	Processor	\\SBOOK15
✓	🟪	1.0	DPC Rate	_Total	---	Processor	\\SBOOK15

Sysinternals Process Explorer



Process Explorer - Sysinternals: www.sysinternals.com [DARCOLABS\Benny]

File Options View Process Find Users Help

<Filter by name>

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Secure System	Susp...	184 K	138,840 K	88		
Registry						
System Idle Process						
System						
Interrupts						
smss.exe						
Memory Compression						
csrss.exe						
wininit.exe						
services.exe						
svchost.exe						
WmiPrvSE.exe						
WmiPrvSE.exe						
MoUsoCoreWorker.exe						
dllhost.exe						
StartMenuExperience...						
RuntimeBroker.exe						
SearchApp.exe						
RuntimeBroker.exe						
WhatsApp.exe						
RuntimeBroker.exe						
ShellExperienceHost....						
RuntimeBroker.exe						
dllhost.exe						
PhoneExperienceHos...						
RuntimeBroker.exe						
RuntimeBroker.exe						
dllhost.exe						
RuntimeBroker.exe						

CPU Usage: 5.15% Commit Charg

System Information

Summary CPU Memory I/O GPU

CPU

3.20%

Totals	
Handles	164,327
Threads	5,326
Processes	323

CPU	
Context Switch Delta	15,329
Interrupt Delta	8,579
DPC Delta	678

Topology	
Cores	6
Sockets	1
Logical Processors	12

☐ Show one graph per CPU

OK

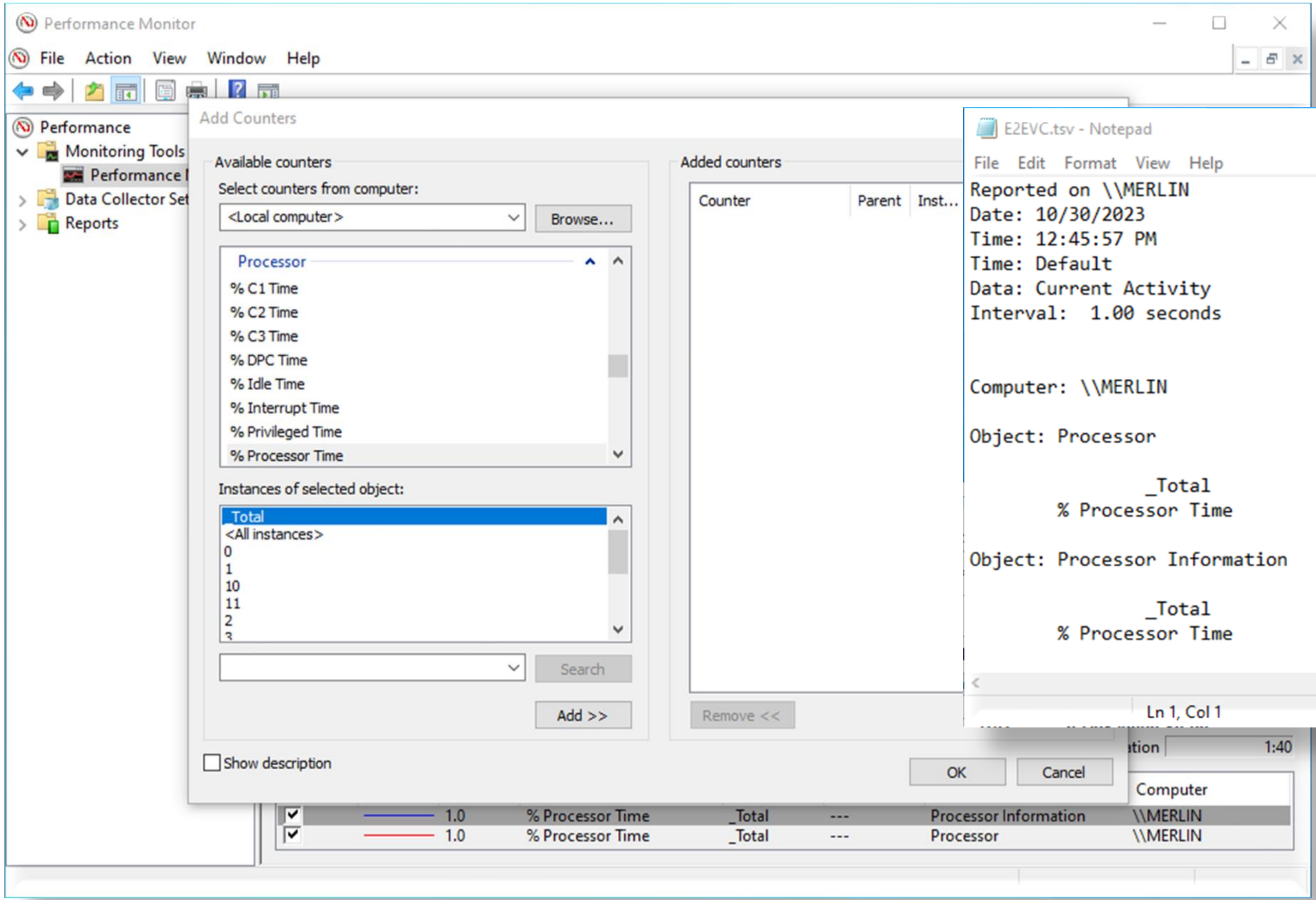
Command-Line Consumers

- **Typeperf** writes performance data to the command window or to a log file
- **Logman** creates and manages Event Trace Session and Performance logs and supports many functions of Performance Monitor from the command line
- **Relog** extracts performance counters from performance counter logs into other formats, such as text-TSV (for tab-delimited text), text-CSV (for comma-delimited text), binary-BIN (BLG), or SQL

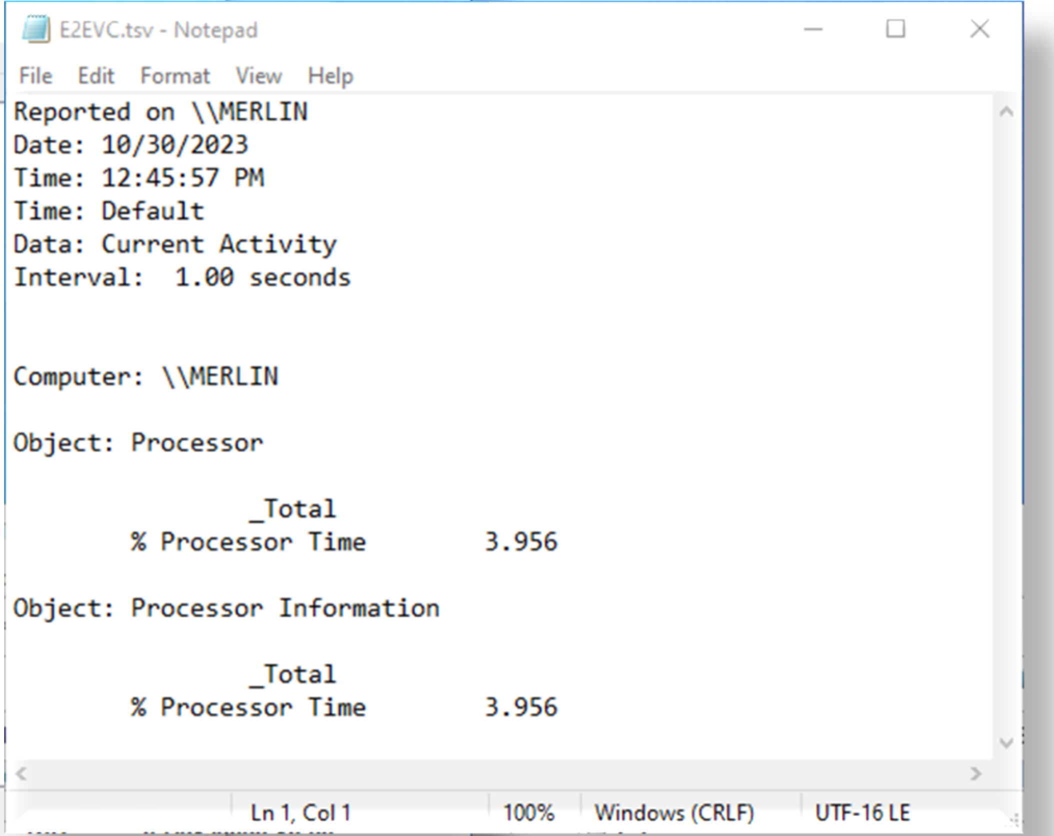
Performance Data Provider Tools

- **CtrPP** is a command-line build tool from the Windows SDK that validates and compiles a Performance Counters V2 provider manifest. This tool generates the .h headers and .rc resource scripts needed to build a V2 provider
- **LodCtr** is the command-line tool used to install a provider onto a system
- **UnlodCtr** is the command-line tool used to uninstall a provider from a system

PerfMon: Add Counters and Save Settings



✓		1.0	% Processor Time	_Total	---	Processor Information	\\MERLIN
✓		1.0	% Processor Time	_Total	---	Processor	\\MERLIN



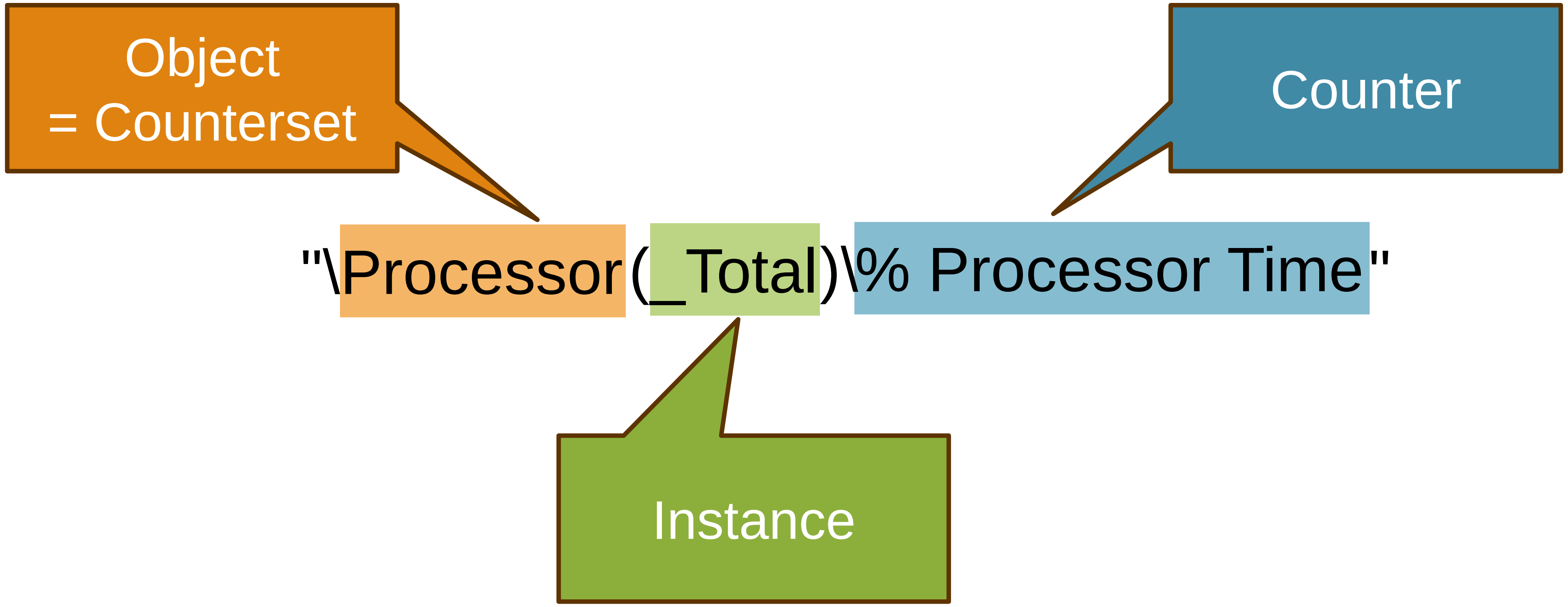
```
Reported on \\MERLIN
Date: 10/30/2023
Time: 12:45:57 PM
Time: Default
Data: Current Activity
Interval:  1.00 seconds


Computer: \\MERLIN

Object: Processor

                _Total
        % Processor Time        3.956

Object: Processor Information

                _Total
        % Processor Time        3.956
```

Some Important EUC Counters

Counter	Instance	Object
Available MBytes		Memory
Free System Page Table Entries		Memory
Page Faults/sec		Memory
Pages/sec		Memory
Pool Nonpaged Bytes		Memory
Pool Paged Bytes		Memory
Bytes Total/sec	*	Network Adapter
Avg. Disk Queue Length	_Total	PhysicalDisk
Working Set	_Total	Process
% Processor Time	_Total	Processor
% Interrupt Time	_Total	Processor
Interrupts/sec	_Total	Processor
Context Switches/sec		System
Processes		System
Processor Queue Length		System
Active Sessions		Terminal Services
[A range of counters]	*	RemoteFX Network
[A range of counters]	*	RemoteFX Graphics
Max Input Delay	Max	User Input Delay per Session

Performance Counter Path Syntax

\\ComputerName\ObjectName(ObjectInstance)\ObjectCounter



IMPORTANT: Believe it or not, but counter names are localized, so above example works only on English systems!

Performance Logs and Alerts (PLA)

- Provides application programmers the ability to generate alert notifications based on performance counter thresholds
  - Create new Data Collector Set
  - Create manually (Advanced)
  - Create data logs - performance counter, event trace data, system configuration information
- In Performance Monitor
  - Add performance counters and sample interval
  - Go to existing data collector set and change log format to create CSV files in the PerfLog folder
  - In case Binary format was selected, the BLG file can be converted to CSV by the relog command located in WINDOWS\System32
    - relog -f csv "C:\location\blg\file.blg" -o "C:\location\output\file.csv"

PerfMon Data Collector Sets

- The Logman command can start and stop a PerfMon Data Collector Set
  - Logman start "EUC Score"
  - Logman stop "EUC Score"
- Logman query shows all scheduled tasks created by Performance Monitor
- Logman import "EUC Score" -xml c:\windows\perf_log.xml
- In Data Collector Set properties, use Stop Conditions and Schedule to trigger by Task Scheduler. In Task Scheduler, the scheduled task is visible under the "Task Scheduler Library" > "Microsoft" > "Windows" > "PLA" folder.
- HINT: In case Task Scheduler is not running, start it with "net start task scheduler"
- HINT: Schtasks.exe enables an administrator to create, delete, query, change, run and end scheduled tasks on a local or remote system.

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

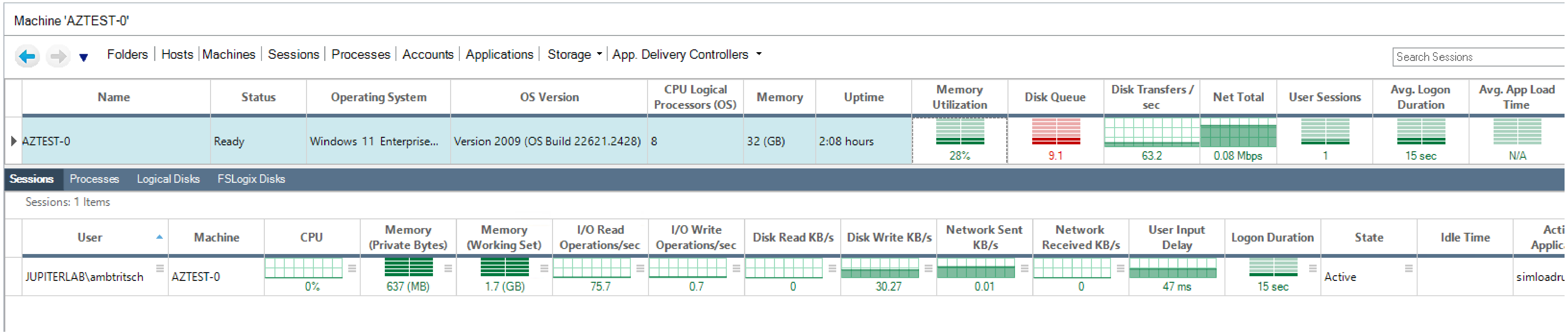
Dealing With Localized Counter Names

- 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

3rd Party Monitoring Solutions

For example, ControlUp Management Console visualizing performance data collected by the ControlUp Real-Time Agent (a consumer)



Machine 'AZTEST-0'

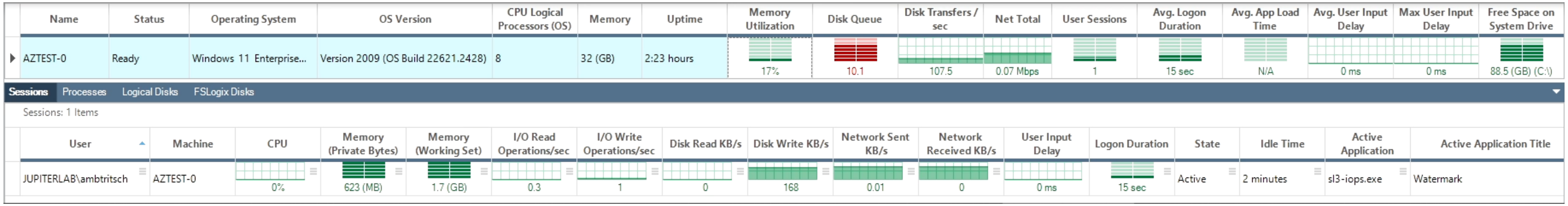
Folders | Hosts | Machines | Sessions | Processes | Accounts | Applications | Storage | App. Delivery Controllers
Search Sessions

Name	Status	Operating System	OS Version	CPU Logical Processors (OS)	Memory	Uptime	Memory Utilization	Disk Queue	Disk Transfers / sec	Net Total	User Sessions	Avg. Logon Duration	Avg. App Load Time
AZTEST-0	Ready	Windows 11 Enterprise...	Version 2009 (OS Build 22621.2428)	8	32 (GB)	2:08 hours	28%	9.1	63.2	0.08 Mbps	1	15 sec	N/A

Sessions Processes Logical Disks FSLogix Disks

Sessions: 1 Items

User	Machine	CPU	Memory (Private Bytes)	Memory (Working Set)	I/O Read Operations/sec	I/O Write Operations/sec	Disk Read KB/s	Disk Write KB/s	Network Sent KB/s	Network Received KB/s	User Input Delay	Logon Duration	State	Idle Time	Acti Applic
JUPITERLAB\ambtritsch	AZTEST-0	0%	637 (MB)	1.7 (GB)	75.7	0.7	0	30.27	0.01	0	47 ms	15 sec	Active		simloadru

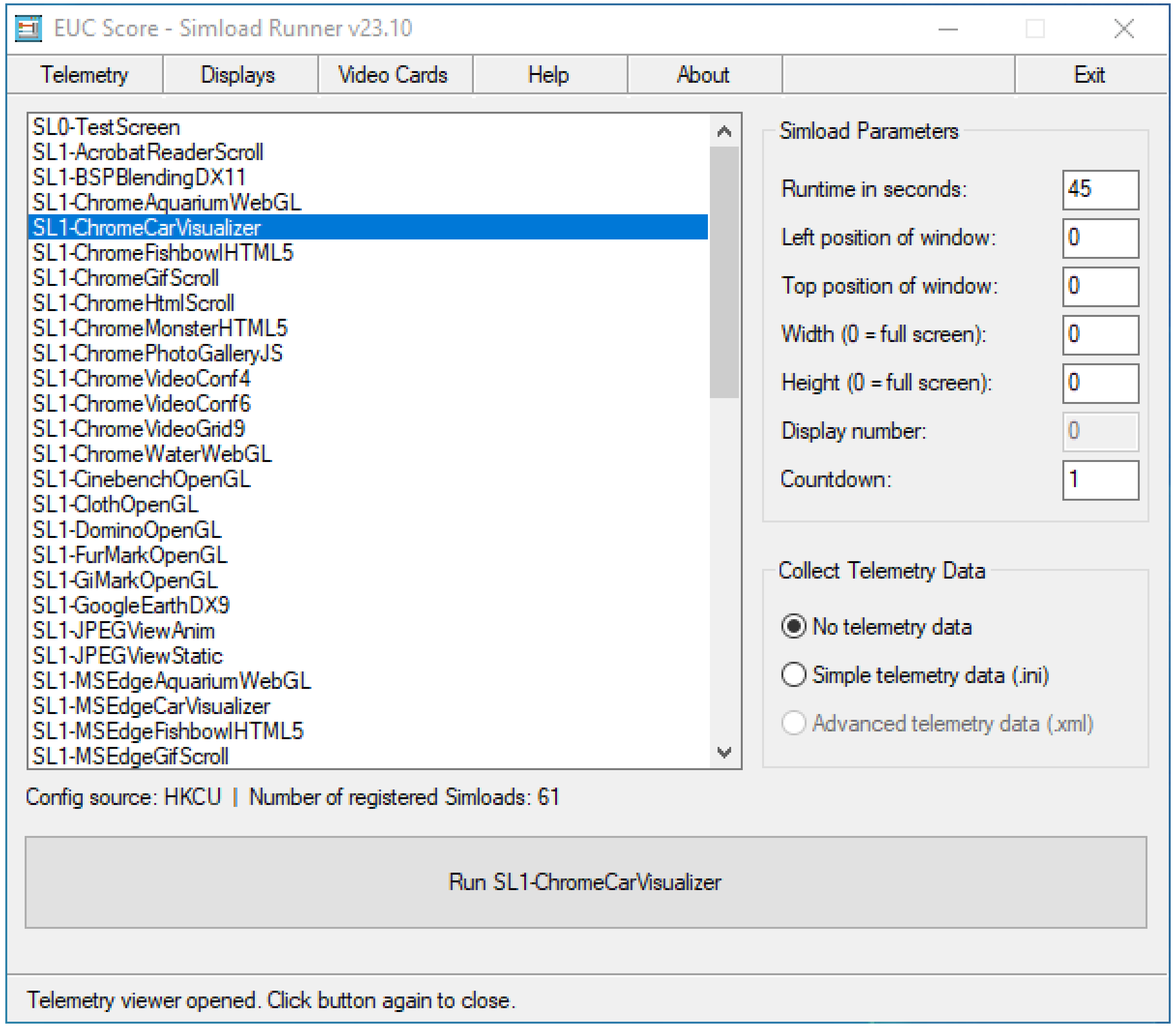


Name	Status	Operating System	OS Version	CPU Logical Processors (OS)	Memory	Uptime	Memory Utilization	Disk Queue	Disk Transfers / sec	Net Total	User Sessions	Avg. Logon Duration	Avg. App Load Time	Avg. User Input Delay	Max User Input Delay	Free Space on System Drive
AZTEST-0	Ready	Windows 11 Enterprise...	Version 2009 (OS Build 22621.2428)	8	32 (GB)	2:23 hours	17%	10.1	107.5	0.07 Mbps	1	15 sec	N/A	0 ms	0 ms	88.5 (GB) (C:\)

Sessions: 1 Items

User	Machine	CPU	Memory (Private Bytes)	Memory (Working Set)	I/O Read Operations/sec	I/O Write Operations/sec	Disk Read KB/s	Disk Write KB/s	Network Sent KB/s	Network Received KB/s	User Input Delay	Logon Duration	State	Idle Time	Active Application	Active Application Title
JUPITERLAB\ambtritsch	AZTEST-0	0%	623 (MB)	1.7 (GB)	0.3	1	0	168	0.01	0	0 ms	15 sec	Active	2 minutes	sl3-iops.exe	Watermark

EUC Score Telemetry Collectors



SL0-TestScreen
SL1-AcrobatReaderScroll
SL1-BSPBlendingDX11
SL1-ChromeAquariumWebGL
SL1-ChromeCarVisualizer
SL1-ChromeFishbowlHTML5
SL1-ChromeGifScroll
SL1-ChromeHtmlScroll
SL1-ChromeMonsterHTML5
SL1-ChromePhotoGalleryJS
SL1-ChromeVideoConf4
SL1-ChromeVideoConf6
SL1-ChromeVideoGrid9
SL1-ChromeWaterWebGL
SL1-CinebenchOpenGL
SL1-ClothOpenGL
SL1-DominoOpenGL
SL1-FurMarkOpenGL
SL1-GiMarkOpenGL
SL1-GoogleEarthDX9
SL1-JPEGViewAnim
SL1-JPEGViewStatic
SL1-MSEdgeAquariumWebGL
SL1-MSEdgeCarVisualizer
SL1-MSEdgeFishbowlHTML5
SL1-MSEdgeGifScroll

Simload Parameters

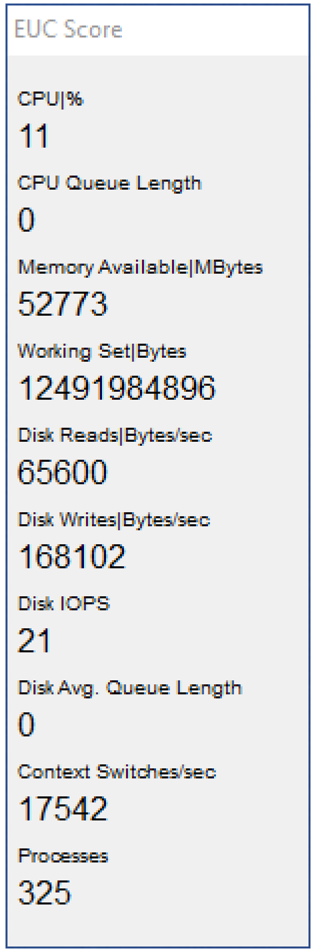
Runtime in seconds:	45
Left position of window:	0
Top position of window:	0
Width (0 = full screen):	0
Height (0 = full screen):	0
Display number:	0
Countdown:	1

Collect Telemetry Data

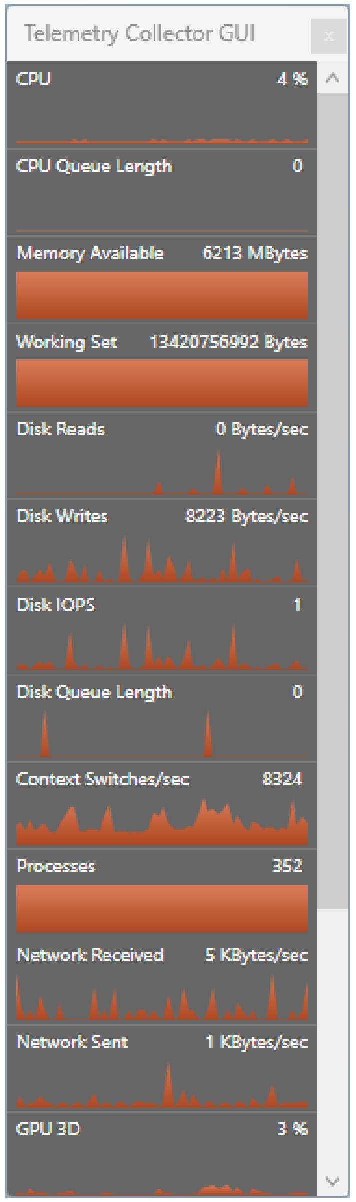
[✓ No telemetry data]
Simple telemetry data (.ini)
Advanced telemetry data (.xml)

Config source: HKCU | Number of registered Simloads: 61

Telemetry viewer opened. Click button again to close.



CPU|%	11
CPU Queue Length	0
Memory Available|MBytes	52773
Working Set|Bytes	12491984896
Disk Reads|Bytes/sec	65600
Disk Writes|Bytes/sec	168102
Disk IOPS	21
Disk Avg. Queue Length	0
Context Switches/sec	17542
Processes	325



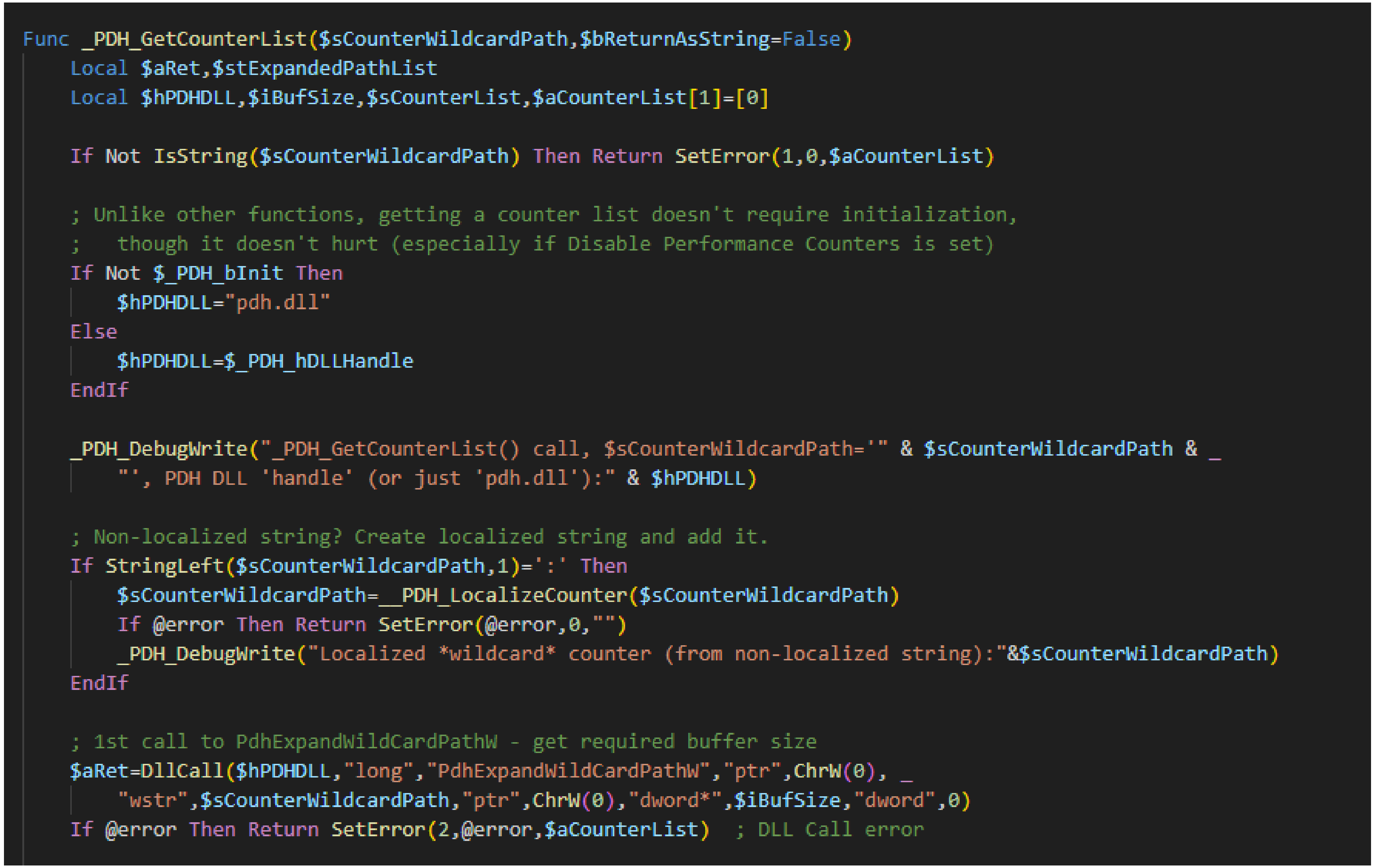
CPU	4 %
CPU Queue Length	0
Memory Available	6213 MBytes
Working Set	13420756992 Bytes
Disk Reads	0 Bytes/sec
Disk Writes	8223 Bytes/sec
Disk IOPS	1
Disk Queue Length	0
Context Switches/sec	8324
Processes	352
Network Received	5 KBytes/sec
Network Sent	1 KBytes/sec
GPU 3D	3 %

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
```

AutoIt Code Base
Only single-instance counters

AutoIt – _PDH_PerformanceCounters



```
Func _PDH_GetCounterList($sCounterWildcardPath,$bReturnAsString=False)
    Local $aRet,$stExpandedPathList
    Local $hPDHDLL,$iBufSize,$sCounterList,$aCounterList[1]=[0]

    If Not IsString($sCounterWildcardPath) Then Return SetError(1,0,$aCounterList)

    ; Unlike other functions, getting a counter list doesn't require initialization,
    ;   though it doesn't hurt (especially if Disable Performance Counters is set)
    If Not $_PDH_bInit Then
        $hPDHDLL="pdh.dll"
    Else
        $hPDHDLL=$_PDH_hDLLHandle
    EndIf

    _PDH_DebugWrite("_PDH_GetCounterList() call, $sCounterWildcardPath='" & $sCounterWildcardPath & _
        "', PDH DLL 'handle' (or just 'pdh.dll'):" & $hPDHDLL)

    ; Non-localized string? Create localized string and add it.
    If StringLeft($sCounterWildcardPath,1)=':' Then
        $sCounterWildcardPath=__PDH_LocalizeCounter($sCounterWildcardPath)
        If @error Then Return SetError(@error,0,"")
        _PDH_DebugWrite("Localized *wildcard* counter (from non-localized string):"&$sCounterWildcardPath)
    EndIf

    ; 1st call to PdhExpandWildCardPathW - get required buffer size
    $aRet=DllCall($hPDHDLL,"long","PdhExpandWildCardPathW","ptr",ChrW(0), _
        "wstr",$sCounterWildcardPath,"ptr",ChrW(0),"dword*",$iBufSize,"dword",0)
    If @error Then Return SetError(2,@error,$aCounterList)  ; DLL Call error
```

CSV Result Files

```
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
```

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

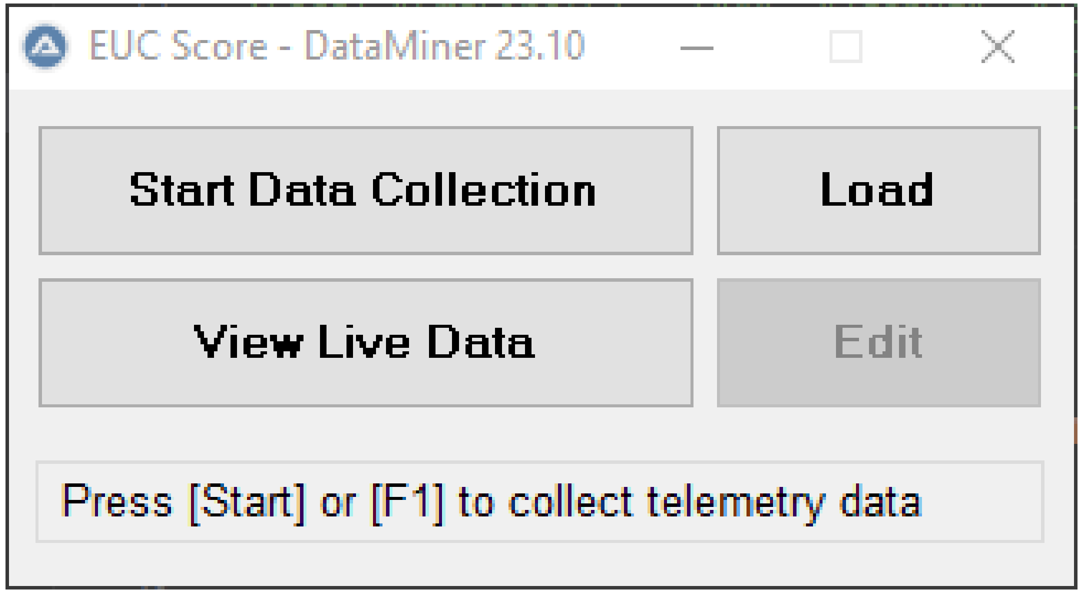
TC-specific Metrics

```
<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>
```

What’s Next (Part 1)

EUC Score Data Miner

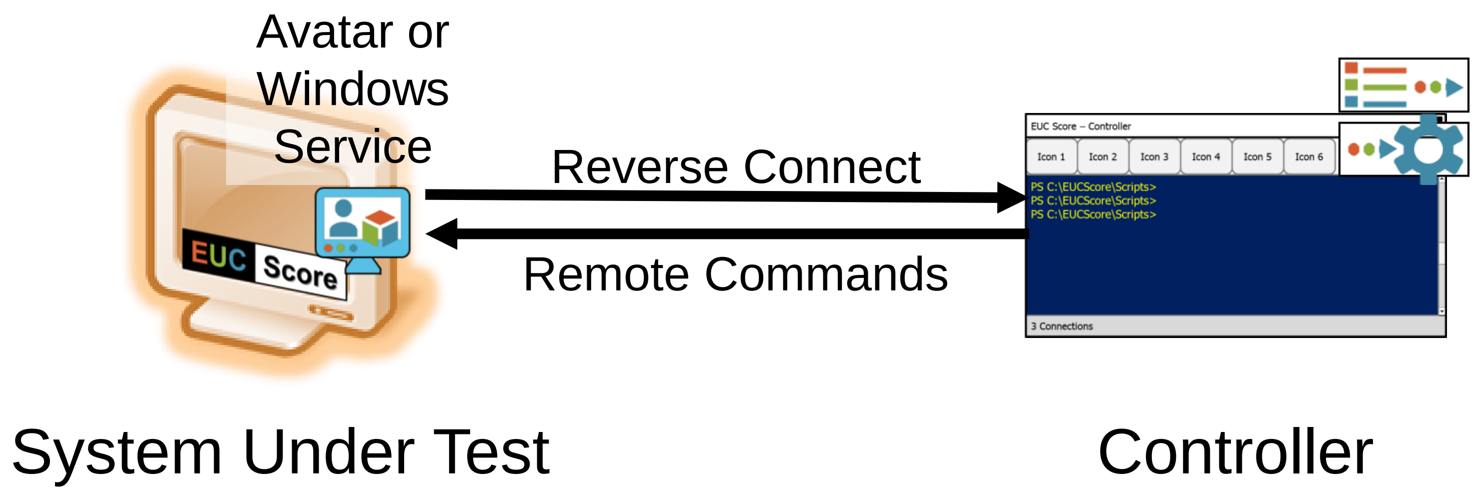
- Stand-alone performance counter consumer
- Configurable by INI file
- CSV output file
- 1 second sample intervals
- Pre-launch countdown



Press [Start] or [F1] to collect telemetry data

What’s Next (Part 2)

- EUC Score Avatar or EUC Score Windows Service reverse connect to an EUC Score Controller
- The EUC Score Controller can send PowerShell commands to launch Simloads or to collect performance data



```
PS C:\EUCScore\Scripts>
PS C:\EUCScore\Scripts>
PS C:\EUCScore\Scripts>
```

Perf Counters look scary at night…

…not so much when viewed under light

Call to Action

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

info@eucscore.com

EUC Score

https://eucscore.com

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

Thank You

Benny Tritsch | info@eucscore.com | @drtritsch