

Mastering Windows Performance Counters for AVD and Cloud PCs

Dr. Benny Tritsch
Independent Performance
Data Scientist



Workplace Ninjas





Thank you Sponsors

www.workplaceninjas.us

Platinum Sponsors

robopack 



SOFTWARE
CENTRAL

Gold Sponsors

Recast



Microsoft



PATCH MY PC

inforcer 



DEVICIE

 LOGIN VSI



control 

glueck  kanja

Silver Sponsors

juriba 

 numecent
CLOUDPAGING DELIVERS

liquidware™

 FERROQUE
SYSTEMS

 Rimo3

 WEI



About Benny Tritsch

My Focus

VDI Benchmarking with EUC Score

Community Awards

MVP, Tech Insider, NEPA, VIPP, NVP

My Company

Dr. Tritsch IT Consulting

Contact

info@eucscore.com, info@drtritsch.com

My Blog

<https://drtritsch.com>

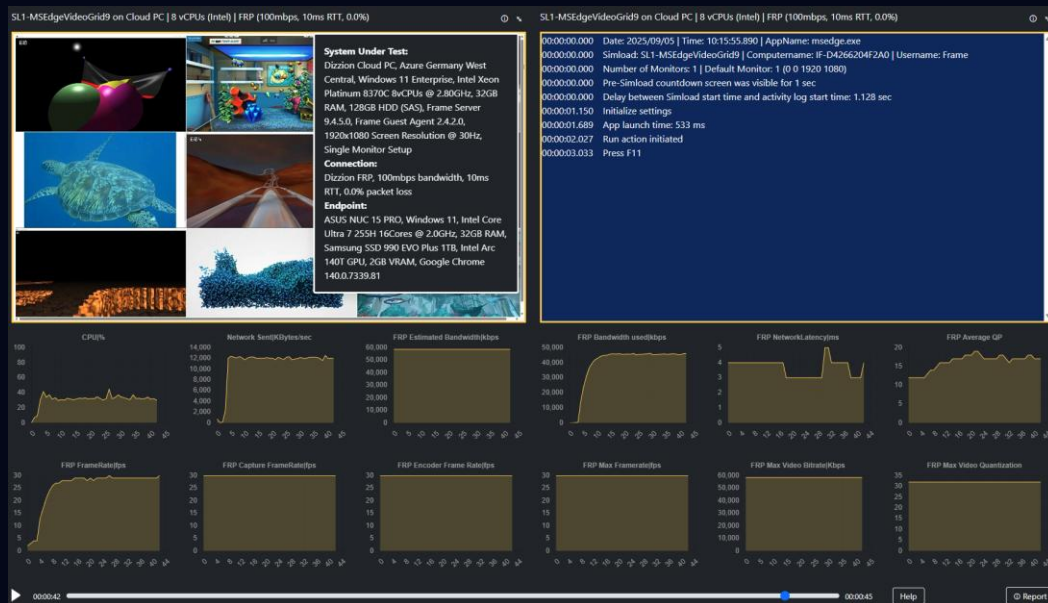
EUC Score Website

<https://eucscore.com>





Why Performance Counters Matter



EUC Score Sync Player Clips





Why Performance Counters Matter

Less control over Windows-specific telemetry data

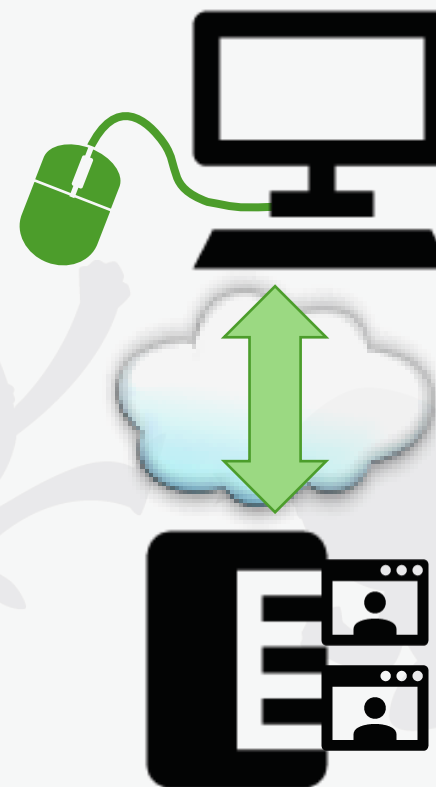
On-Prem Windows



VDI / RDSH on LAN

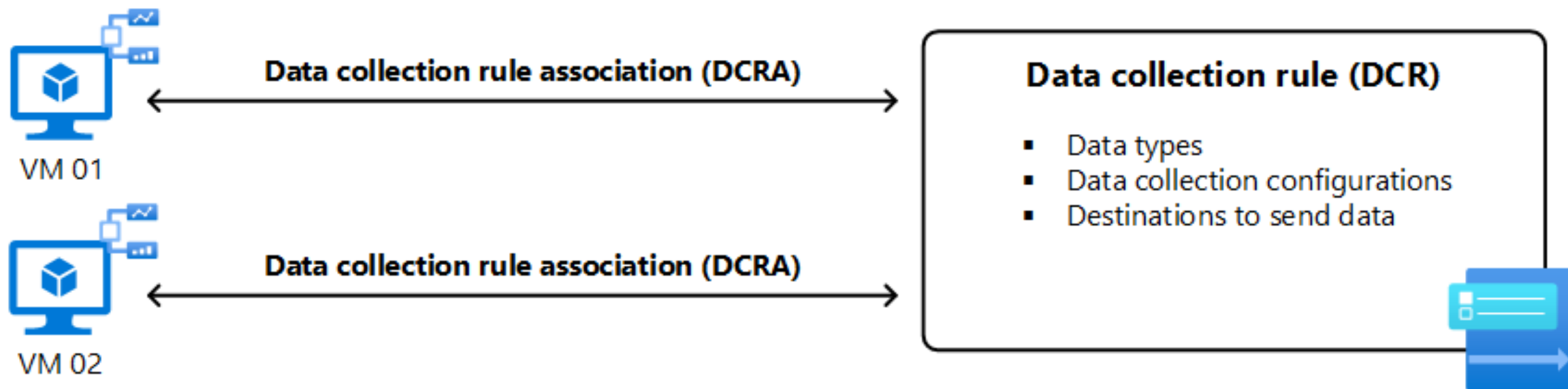


AVD / Win365





Azure Monitor Agent



<https://learn.microsoft.com/en-us/azure/azure-monitor/agents/azure-monitor-agent-overview>

Unfortunately, Azure monitoring was not designed for Windows telemetry data!!!

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

1

Settings

Help

mm-admin@ITProClou...ITPROCLOUD.DE

Home > Data collection rules > microsoft-avdi-westeuropa

Data collection rules

ITProCloud.de

Create

Manage view

Filter for any field...

We are previewing a new Browse experience. Click to switch.

Name

↑↓

hydra-itpc-dev

...

microsoft-avdi-westeuropa

...

microsoft-avdi-westeuropa | Data sources

Data collection rule

Search

+

Add

Delete

Overview

Activity log

Access control (IAM)

Tags

Settings

Locks

Configuration

Data sources

Resources

Automation

Policies (Preview)

CLI / PS

Tasks (preview)

Export template

Security

Identity

Monitoring

Alerts

Metrics

Diagnostic settings

Logs

Help

Support +

Filter by name...

D

Data source

Performance Counters

Windows Event Logs

Add data source

Azure Monitor Agent

Azure Log Analytics

AVD Insights

*Data source

Destination

Select which data source type and the data to collect for your resource(s).

Data source type *

Performance Counters

Choose Basic to enable the collection of performance counters. Choose Custom if you want more control over which performance counters are collected.

None

Basic

Custom

Configure the performance counters to collect, and how often they should be sampled:

Add

Performance counter	Sample rate (seconds)
☒ \Memory\% Committed Bytes In Use	30
☐ \Memory\Available Bytes	60
☐ \Memory\Committed Bytes	60
☐ \Memory\Cache Bytes	60
☐ \Memory\Pool Paged Bytes	60
☐ \Memory\Pool Nonpaged Bytes	60
☒ \Memory\Pages/sec	30
☒ \Memory\Page Faults/sec	30
☐ \Process(_Total)\Working Set	60
☐ \Process(_Total)\Working Set - Private	60

Showing 11 - 20 of 59 results.

<

1

2

3

4

5

>

Save

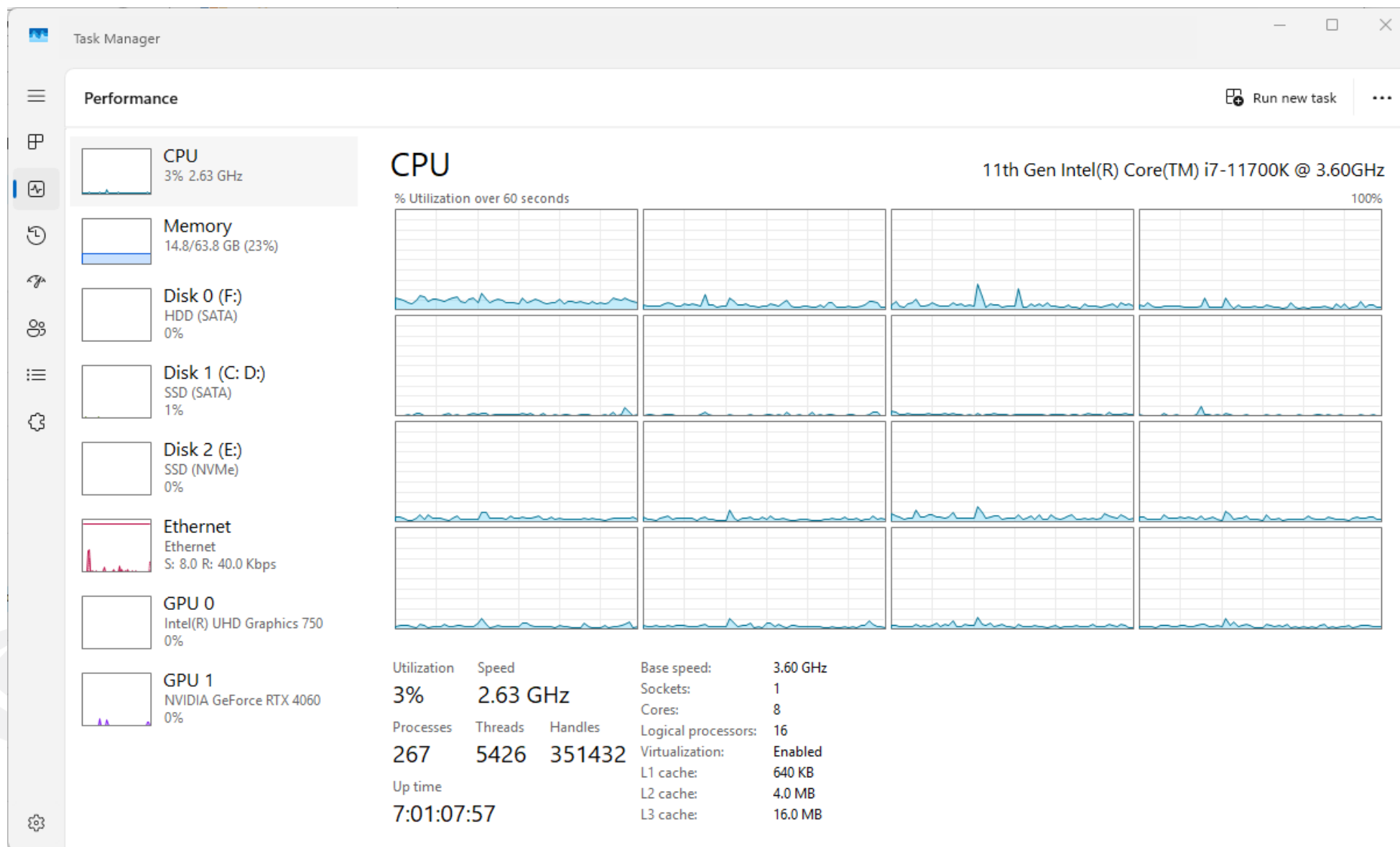
Next : Destination >

Cancel

Azure
Monitor

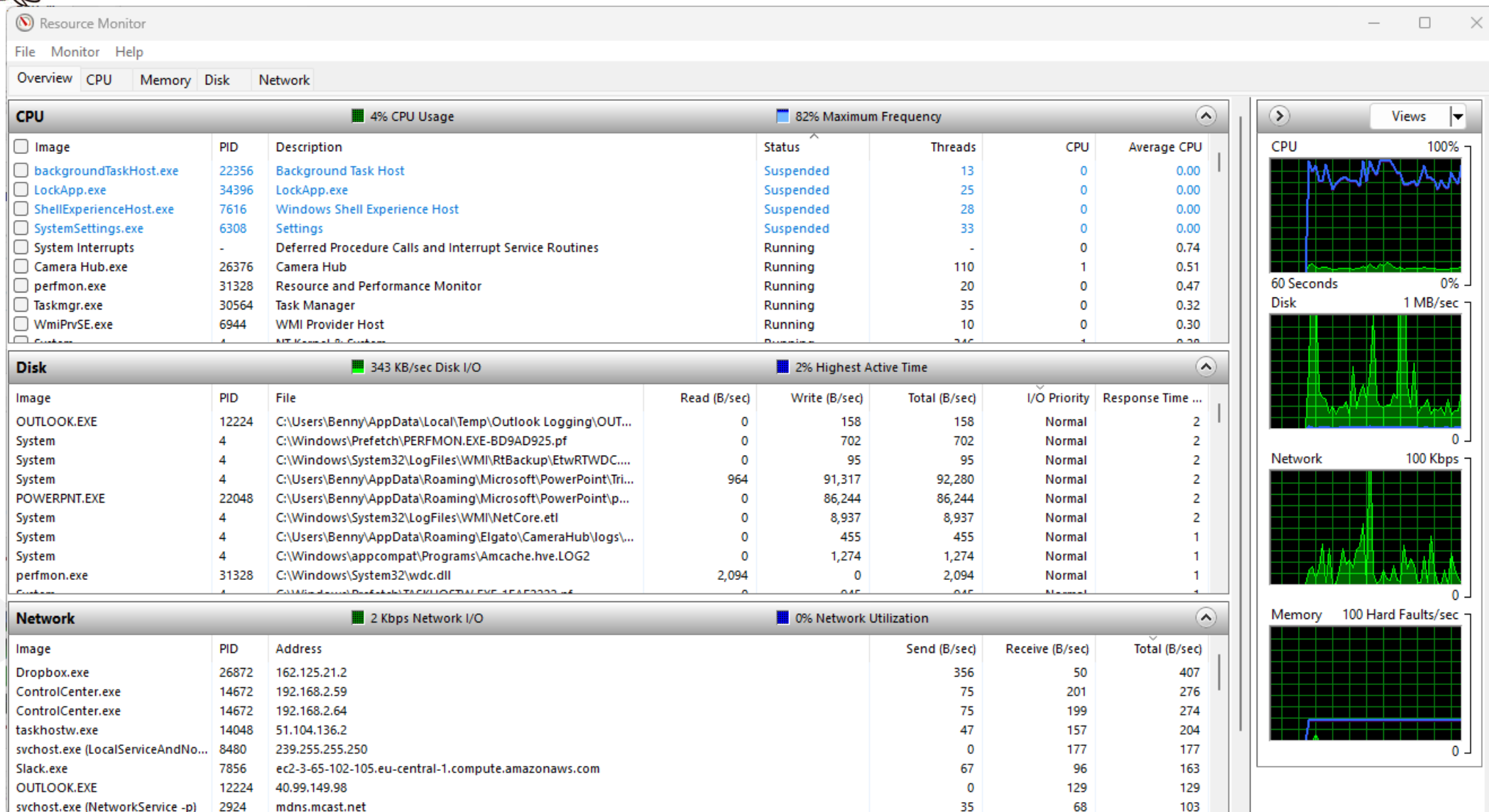


Task Manager – Performance Tab





Resource Monitor





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						
StartMenuExperienceHost.exe						
RuntimeBroker.exe						
SearchApp.exe						
RuntimeBroker.exe						
WhatsApp.exe						
RuntimeBroker.exe						
ShellExperienceHost.exe						
RuntimeBroker.exe						
dllhost.exe						
PhoneExperienceHost.exe						
RuntimeBroker.exe						
RuntimeBroker.exe						
dllhost.exe						
RuntimeBroker.exe						

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

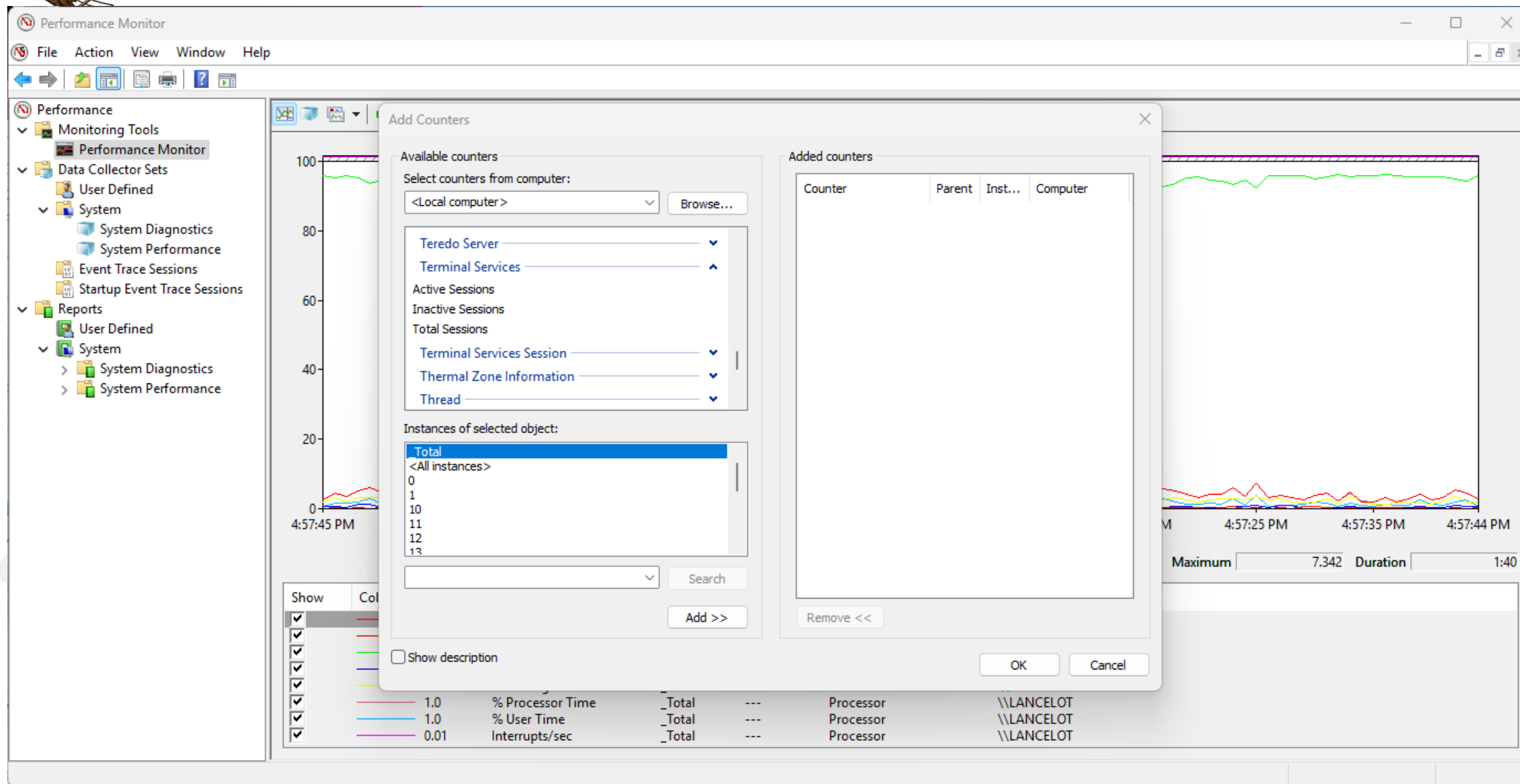
Cores	6
Sockets	1
Logical Processors	12

OK

CPU Usage: 5.15% Commit Charge



The Full View – Performance Monitor





Collecting Performance Metrics

How do performance counters work?





Performance Counter Terminology

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



Performance Counter Terminology

- **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, Relog.exe, WMI, and PowerShell
 - Monitoring agents provided by Microsoft or 3rd party vendors
 - EUC Score: Simload Base Counters, Telemetry Collector, Data Miner



Performance Counter Path Syntax

\\ComputerName\ObjectName(ObjectInstance)\ObjectCounter

Object
= Counterset

Counter

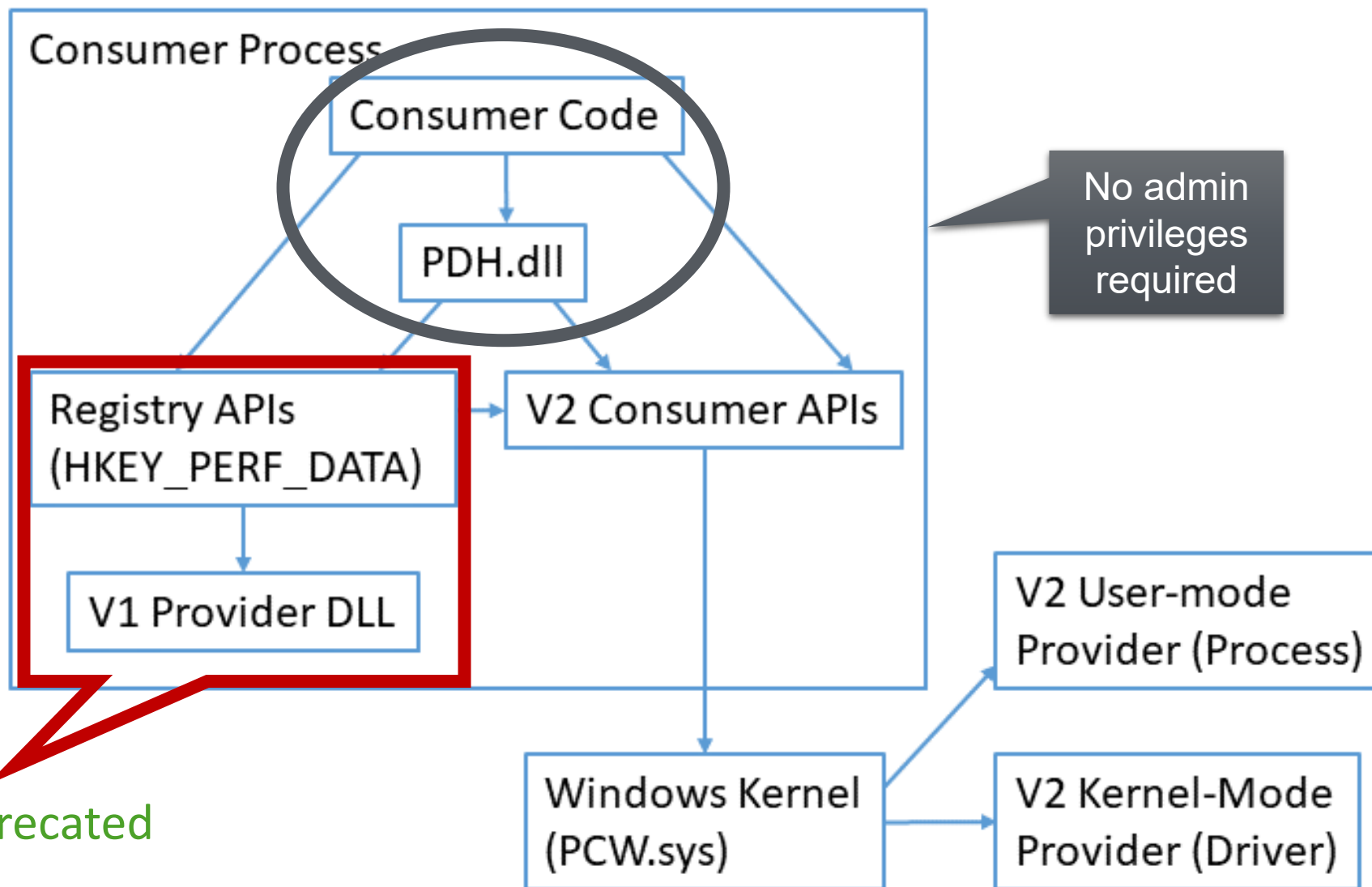
"\Processor (_Total) \% Processor Time "

Instance

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

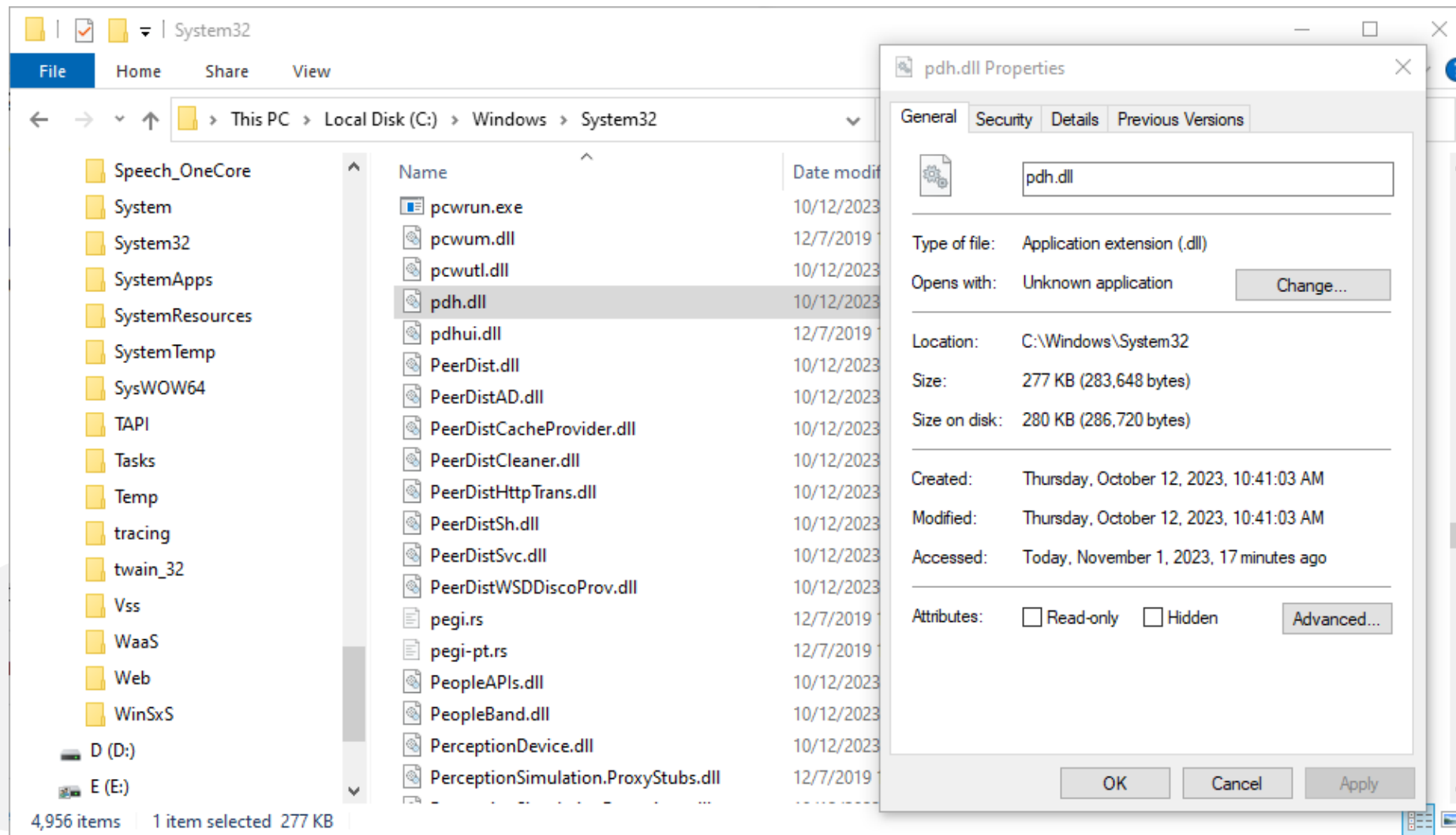


Performance API Architecture





Performance Data Helper DLL PDH.dll



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



RDSH/AVD/Win365 Perf Counters

\Processor(_Total)\% Processor Time
\System\Processor Queue Length
\System\Context Switches/sec
\System\Processes
\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)\Avg. Disk Queue Length

Core Telemetry / Base Counters

\Terminal Services\Active Sessions
\Terminal Services\Total Sessions
\Processor(_Total)\Interrupts/sec
\Processor(_Total)\% Interrupt Time
\System\File Read Operations/sec
\System\File Write Operations/sec
\Memory\Free System Page Table Entries
\Memory\Page Faults/sec
\Memory\Pages/sec
\Memory\Pool Nonpaged Bytes
\Memory\Pool Paged Bytes



RDSH/AVD/Win365 Perf Counters

\Terminal Services Session(RDP-Tcp x)\% Processor Time

\Terminal Services Session(RDP-Tcp x)\Working Set

\User Input Delay per Session(Max)\Max Input Delay

\Network Interface(*)\Output Queue Length

\Network Interface(*)\Current Bandwidth

\Network Interface(*)\Packets Received Unicast/sec

\Network Interface(*)\Packets Received Non-Unicast/sec

\Network Interface(*)\Packets Sent Unicast/sec

\Network Interface(*)\Packets Sent Non-Unicast/sec

\Network Interface(*)\Bytes Received/sec

\Network Interface(*)\Bytes Sent/sec

\Network Interface(*)\Bytes Total/sec

No _Total Instance



Citrix HDX Perf Counters (Win365)

\ICA Session(Console (UserName))\EDT bandwidth available (bps)
\ICA Session(Console (UserName))\Input Session Bandwidth
\ICA Session(Console (UserName))\Output Session Bandwidth
\ICA Session(Console (UserName))\Input Session Compression
\ICA Session(Console (UserName))\Output Session Compression
\ICA Session(Console (UserName))\Latency - Last Recorded
\ICA Session(Console (UserName))\Latency - Session Average
\ICA Session(Console (UserName))\EDT RTT
\ICA Graphics(<n>)\BitRate - Current
\ICA Graphics(<n>)\Frame Rate - Current FPS
\ICA Graphics(<n>)\Frame Rate - Input FPS
\ICA Graphics(<n>)\Frame Rate - Target FPS
\ICA Graphics(<n>)\Thinwire Average Input FPS
\ICA Graphics(<n>)\Thinwire Average Output FPS
\ICA Graphics(<n>)\Thinwire Bandwidth Estimate (bps)
\ICA Graphics(<n>)\Thinwire Latency Estimate (ms)



Omnissa Blast Perf Counters (Win365)

\Horizon Blast Session Counters(Session ID: n; (main))\Estimated Bandwidth (Uplink)
\Horizon Blast Session Counters(Session ID: n; (main))\Instantaneous Transmitted Bytes over TCP
\Horizon Blast Session Counters(Session ID: n; (main))\Instantaneous Transmitted Bytes over UDP
\Horizon Blast Session Counters(Session ID: n; (main))\Instantaneous Received Bytes over TCP
\Horizon Blast Session Counters(Session ID: n; (main))\Instantaneous Received Bytes over UDP
\Horizon Blast Session Counters(Session ID: n; (main))\Packet Loss (Uplink)
\Horizon Blast Session Counters(Session ID: n; (main))\Jitter (Uplink)
\Horizon Blast Session Counters(Session ID: n; (main))\RTT
\Horizon Blast Session Counters(Session ID: n; (main))\Automatic Reconnect Count
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Frames per second
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Dirty frames per second
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\FBC Rate
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Poll Rate
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Outbound Bandwidth (Kbps)
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Inbound Bandwidth (Kbps)
\Horizon Blast Imaging Counters(Session ID: n; Channel: Imaging; (main))\Out Queueing time (us)



Collecting Performance Metrics

Command Line, Scripting and more...





Command Line Tools

- **Typeperf** writes performance data to the command window or to a log file
<https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/typeperf>
- **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

These tools are very powerful, but they're not easy to use

```
C:\Windows\system32\cmd.exe

C:\Users\benny.DARCOLABS>typeperf -q processor
\processor(*)\% Processor Time
\processor(*)\% User Time
\processor(*)\% Privileged Time
\processor(*)\Interrupts/sec
\processor(*)\% DPC Time
\processor(*)\% Interrupt Time
\processor(*)\DPCs Queued/sec
\processor(*)\DPC Rate
\processor(*)\% Idle Time
\processor(*)\% C1 Time
\processor(*)\% C2 Time
\processor(*)\% C3 Time
\processor(*)\C1 Transitions/sec
\processor(*)\C2 Transitions/sec
\processor(*)\C3 Transitions/sec

The command completed successfully.

C:\Users\benny.DARCOLABS>typeperf "\Processor(_Total)\% Processor Time"

"(PDH-CSV 4.0)","\\MERLIN\Processor(_Total)\% Processor Time"
"08/22/2024 09:13:53.884","1.019657"
"08/22/2024 09:13:54.900","1.991526"
"08/22/2024 09:13:55.912","1.445304"
"08/22/2024 09:13:56.918","1.629610"
"08/22/2024 09:13:57.928","5.375511"
"08/22/2024 09:13:58.942","0.992495"
"08/22/2024 09:13:59.956","3.530964"
"08/22/2024 09:14:00.957","3.460074"
"08/22/2024 09:14:01.969","1.571992"
"08/22/2024 09:14:02.977","2.472840"
"08/22/2024 09:14:03.989","0.286005"

The command completed successfully.

C:\Users\benny.DARCOLABS>
```



TypePerf Examples

- `typeperf.exe -q > "C:\Temp\AllCounters.txt"`
- `typeperf -q processor`
- `typeperf -qx "Terminal Services Session"`
- `typeperf "\processor(_Total)\% Processor Time"`
- `typeperf "\processor(_Total)\% Processor Time" "\System\Processor Queue Length" -sc 5`
- `typeperf "\processor(_Total)\% Processor Time" -sc 5 -f CSV -o C:\Temp\CPU.csv`

```
Administrator: Command Prompt

Parameters:
  <counter [counter ...]>      Performance counters to monitor.

Options:
  -?                          Displays context sensitive help.
  -f <CSV|TSV|BIN|SQL>        Output file format. Default is CSV.
  -cf <filename>              File containing performance counters to monitor, one per line.
  -si <[[hh:]mm:]ss>          Time between samples. Default is 1 second.
  -o <filename>               Path of output file or SQL database. Default is STDOUT.
  -q [object]                 List installed counters (no instances). To list counters for one object, include the
                              object name, such as Processor.
  -qx [object]                List installed counters with instances. To list counters for one object, include the
                              object name, such as Processor.
  -sc <samples>               Number of samples to collect. Default is to sample until CTRL+C.
  -config <filename>          Settings file containing command options.
  -s <computer_name>          Server to monitor if no server is specified in the counter path.
  -y                          Answer yes to all questions without prompting.

Note:
  Counter is the full name of a performance counter in
  "\\<Computer>\<Object>(<Instance>)\<Counter>" format,
  such as "\\Server1\Processor(0)\% User Time".

Examples:
  typeperf "\Processor(_Total)\% Processor Time"
  typeperf -cf counters.txt -si 5 -sc 50 -f TSV -o domain2.tsv
  typeperf -qx PhysicalDisk -o counters.txt

C:\Windows\System32>
```



PowerShell Get-Counter Command

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`
- <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.diagnostics/get-counter>



Windows Management Interface

- WMI has preinstalled providers that monitor system performance on both the local system and remotely
- WMI can be used from PowerShell 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
- PowerShell example: `Get-CimInstance -Query "select Name, PercentProcessorTime from Win32_PerfFormattedData_PerfOS_Processor" | Select Name, PercentProcessorTime`
- Good read: <https://xkln.net/blog/analyzing-cpu-usage-with-powershell-wmi-and-excel/>



Collecting Performance Metrics

Here comes some geeky stuff...





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/>



Localized Perf Counter Names

Registry Editor

File Edit View Favorites Help

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\CurrentLanguage

Name	Type	Data
(Default)	REG_SZ	(value not set)
Counter	REG_MULTI_SZ	1 1847 2 System 4 Memory 6 % Processor Time 10 File Read Operations/sec 12 File Write Operations/sec 14 File Co...
Help	REG_MULTI_SZ	3 The System performance object consists of counters that apply to more than one instance of a component proce...

Registrierungs-Editor

Datei Bearbeiten Ansicht Favoriten Hilfe

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009

Name	Typ	Daten
(Standard)	REG_SZ	(Wert nicht festgelegt)
Counter	REG_MULTI_SZ	1 1847 2 System 4 Memory 6 % Processor Time 10 File Read Operations/sec 12 File Write Operations/sec 14 File Co...
Help	REG_MULTI_SZ	3 The System performance object consists of counters that apply to more than one instance of a component proc...

Registrierungs-Editor

Datei Bearbeiten Ansicht Favoriten Hilfe

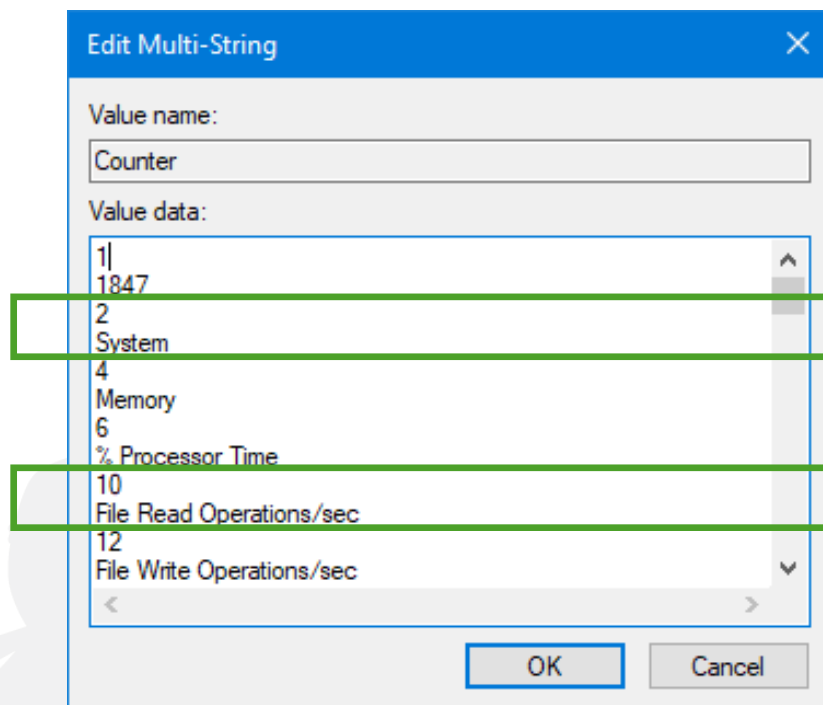
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\CurrentLanguage

Name	Typ	Daten
(Standard)	REG_SZ	(Wert nicht festgelegt)
Counter	REG_MULTI_SZ	1 1847 2 System 4 Arbeitsspeicher 6 Prozessorzeit (%) 10 Lesevorgänge/s 12 Schreibvorgänge/s 14 Dateisteuervorg...
Help	REG_MULTI_SZ	3 Das System-Leistungsindikatorenobjekt besteht aus Leistungsindikatoren, die für mehrere Instanzen eines Komp...

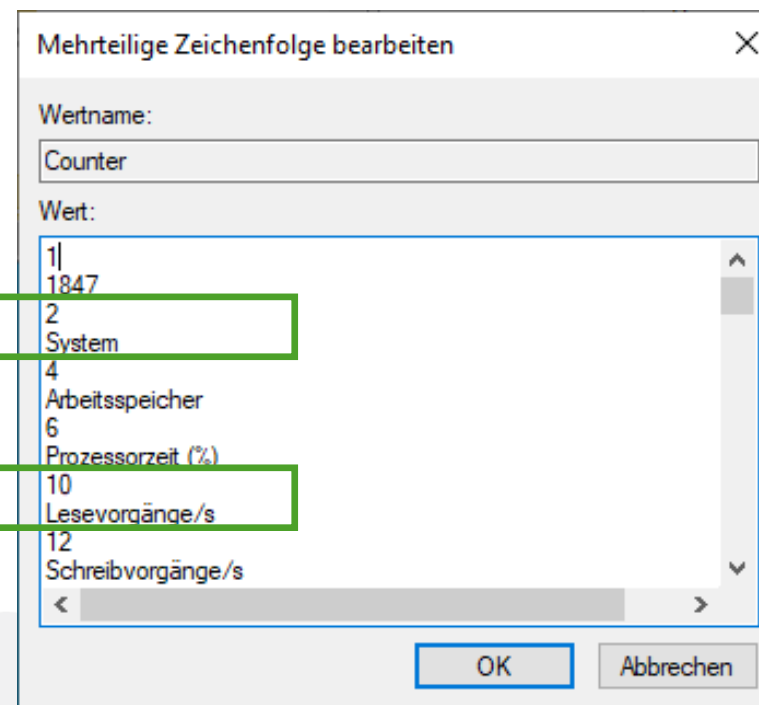


Localized Perf Counter Names

Perflib\CurrentLanguage\Counter
on a system with English (United States) language



Perflib\CurrentLanguage\Counter
on a system with German language



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\CurrentLanguage



Get-PerformanceCounterId

```
function Get-PerformanceCounterId
{
    param(
        [Parameter(Mandatory)]
        [string] $Name,
        $ComputerName = $env:COMPUTERNAME
    )

    $code = '[DllImport("pdh.dll", SetLastError=true, CharSet=CharSet.Unicode)]public static extern
    UInt32 PdhLookupPerfIndexByName(string szMachineName, string szNameBuffer, ref uint
    dwNameIndex);'
    $type = Add-Type -MemberDefinition $code -PassThru -Name PerfCounter2 -Namespace Utility

    [UInt32]$Index = 0
    if ($type::PdhLookupPerfIndexByName($ComputerName, $Name, [Ref]$Index) -eq 0)
    {
        $index
    }
    else
    {
        throw "Cannot find '$Name' on '$ComputerName'."
    }
}
```



Localized Perf Counter Names

```
PS F:\> Get-Counter -Counter "\System\File Read Operations/sec"
```

Timestamp	CounterSamples
-----	-----
7/30/2020 2:24:42 PM	\\host1\system\file read operations/sec : 359.52894517603

```
PS F:\> Get-Counter -Counter "\2\10"
```

Timestamp	CounterSamples
-----	-----
7/30/2020 2:24:51 PM	\\host1\2\10 : 341.618616976008

- Unfortunately, many “localized” counter IDs do not work with Get-Counter
- And the decimal separator – period (.) or a comma (,) – depends on the locale...
- ...as does the 1000 separator (expect big fun with CSV files)



Language-Agnostic Counters

Simload.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

```
Name1=CPU|%
Counter1=:238\6\(_Total)
Name2=CPU Queue Length
Counter2=:2\44\
Name3=Memory Available|MBytes
Counter3=:4\1382\
Name4=Working Set|Bytes
Counter4=:230\180\(_Total)
Name5=Disk Reads|Bytes/sec
Counter5=:234\220\(_Total)
Name6=Disk Writes|Bytes/sec
Counter6=:234\222\(_Total)
Name7=Disk IOPS
Counter7=:234\212\(_Total)
Name8=Disk Avg. Queue Length
Counter8=:234\1400\(_Total)
Name9=Context Switches/sec
Counter9=:2\146\
Name10=Processes
Counter10=:2\248\
```

Get-Counter / TypePerf

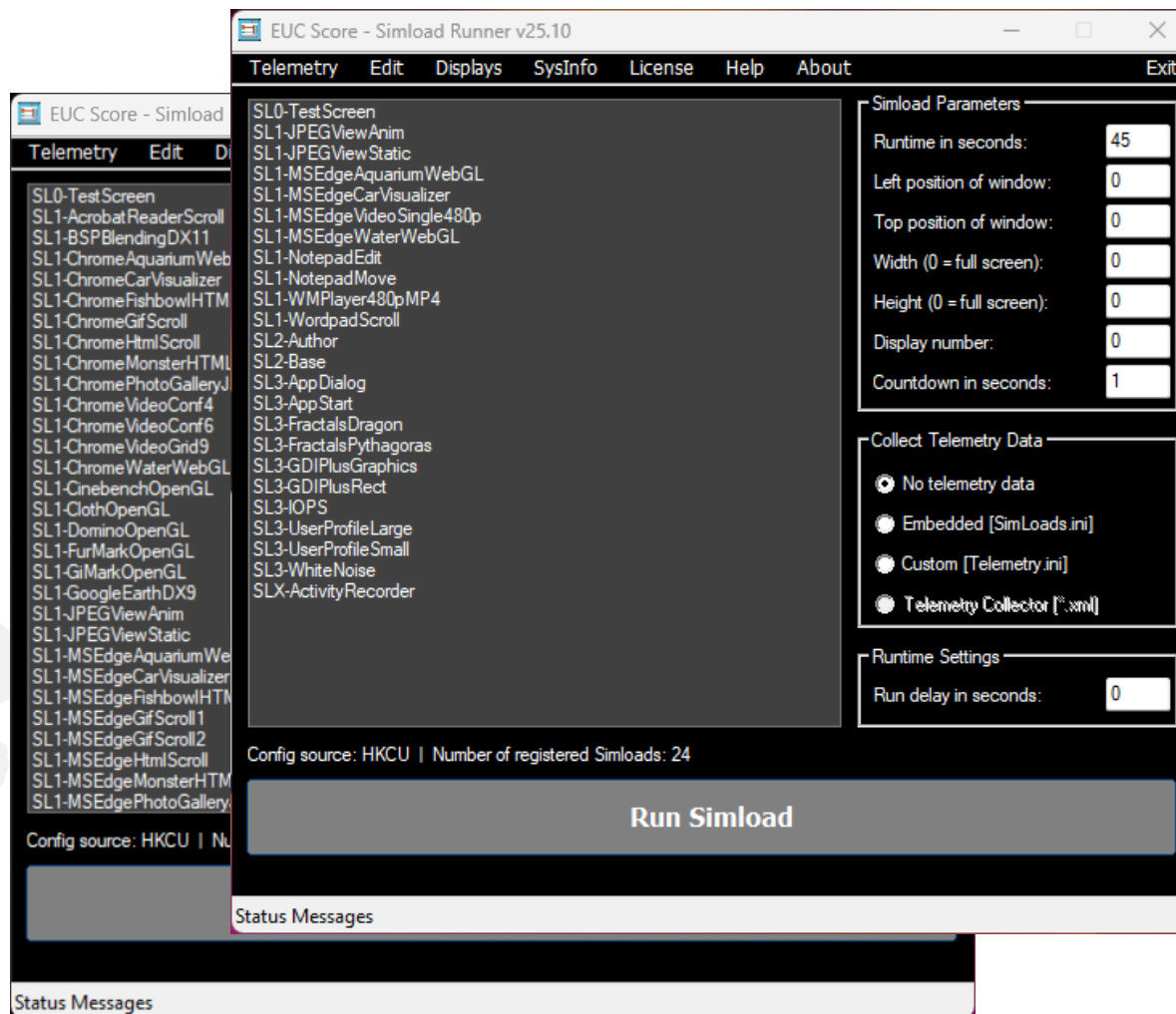
```
CPU|%
-Counter "\238(_Total)\6"
CPU Queue Length
-Counter "\2\44"
Memory Available
-Counter "\4\1382"
Working Set
-Counter "\230(_Total)\180"
Disk Reads
-Counter "\234(_Total)\220"
Disk Writes
-Counter "\234(_Total)\222"
Disk IOPS
-Counter "\234(_Total)\212"
Disk Avg. Queue Length
-Counter "\234(_Total)\1400"
Context Switches/sec
-Counter "\2\146"
Processes
-Counter "\2\248"
```



EUC Score Simload Base Counters

Free Download: EUC Score
Base Package

<https://eucscore.com/freeware>



EUC Score

CPU|%

6

CPU Queue Length

0

Memory Available|MBytes

16441

Working Set|Bytes

12791861248

Disk Reads|Bytes/sec

0

Disk Writes|Bytes/sec

8172

Disk IOPS

0

Disk Avg. Queue Length

0

Context Switches/sec

3934

Processes

333



CSV Result Files

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
2024.08.08 10:30:36.730	0	1	48828	1.721609E+10	0	0	0	0	0	350
2024.08.08 10:30:37.802	16.0371	0	48808	1.72669E+10	1589410	623872.5	96.00307	0	41422.38	351
2024.08.08 10:30:38.823	16.24802	0	48806	1.728885E+10	2385041	0	19.6916	0	55233.54	352
2024.08.08 10:30:39.823	12.55208	0	48804	1.728939E+10	1083131	354221.5	30.81559	0	24862.56	352
2024.08.08 10:30:40.834	14.79185	0	48810	1.728841E+10	0	158669.5	26.8188	0	25856.44	352
2024.08.08 10:30:41.850	13.87708	0	48808	1.729144E+10	1009577	204725.7	46.0234	0	32820.31	352
2024.08.08 10:30:42.861	8.800934	1	48804	1.728834E+10	0	89528.84	6.960476	0	29901.39	352
2024.08.08 10:30:43.859	13.36615	0	48805	1.728833E+10	0	159098.4	26.89124	0	34092.06	352
2024.08.08 10:30:44.886	17.26958	0	48796	1.729805E+10	0	124779.1	24.56684	0	44967.18	352
2024.08.08 10:30:45.888	15.71634	0	48800	1.729505E+10	0	162253	24.73013	0	41759.28	352
2024.08.08 10:30:46.913	14.22636	0	48800	1.729274E+10	0	622944	21.60948	0	34674.31	352
2024.08.08 10:30:47.895	9.426932	0	48798	1.729646E+10	0	421994.3	42.42533	0	32910.01	352
2024.08.08 10:30:48.913	9.435274	0	48801	1.729297E+10	0	181588.1	34.47602	0	32947.26	352
2024.08.08 10:30:49.922	12.37798	0	48800	1.732194E+10	0	232905.4	36.62416	0	32353.19	353
2024.08.08 10:30:50.965	14.22228	0	48790	1.732325E+10	94723.96	146031	19.42027	0	32691.35	353
2024.08.08 10:30:51.940	13.66384	0	48792	1.732352E+10	0	182360	30.35652	0	34137.27	353
2024.08.08 10:30:52.945	12.97976	0	48784	1.733555E+10	0	216170.4	31.86481	0	33511.35	353
2024.08.08 10:30:53.964	8.67326	0	48781	1.733713E+10	0	81109.59	14.8483	0	29580.76	353
2024.08.08 10:30:54.988	9.81427	0	48784	1.733138E+10	0	63610.1	5.824227	0	32242.77	353
2024.08.08 10:30:55.981	7.289457	0	48784	1.733362E+10	0	140021.6	16.08899	0	31924.67	353



Collecting Performance Metrics

What's still missing? Going deeper...





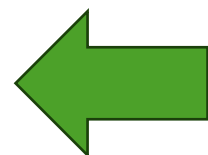
The Perf Counters We Don't Have

- GPU Performance Counters
 - 3D (rendering)
 - Video Decode
 - Video Encode (NVIDIA) / Video Processing (AMD, INTEL)
 - Dedicated Memory (Frame Buffer)
- Network Performance Counters
 - \Network Interface(_Total)\Bytes Received/sec
 - \Network Interface(_Total)\Bytes Sent/sec
- User Session ID – Auto-Detect



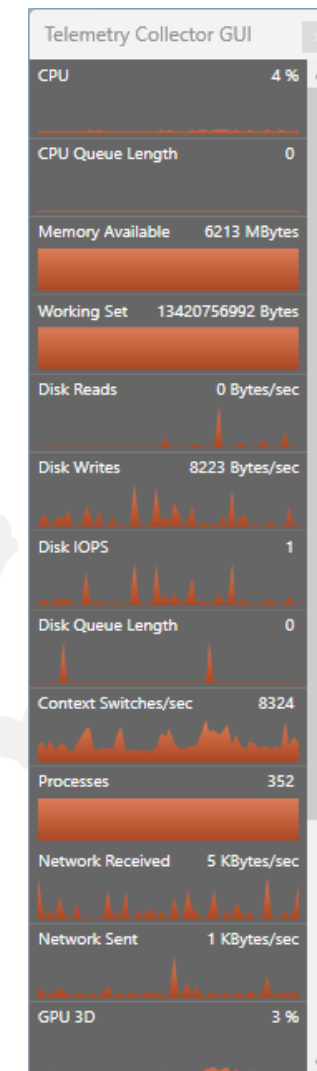
Beyond Standard Perf Counters

EUC Score	
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



Core Telemetry / Base Counters

Avatar Telemetry Collector





Telemetry Collector

```
<?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



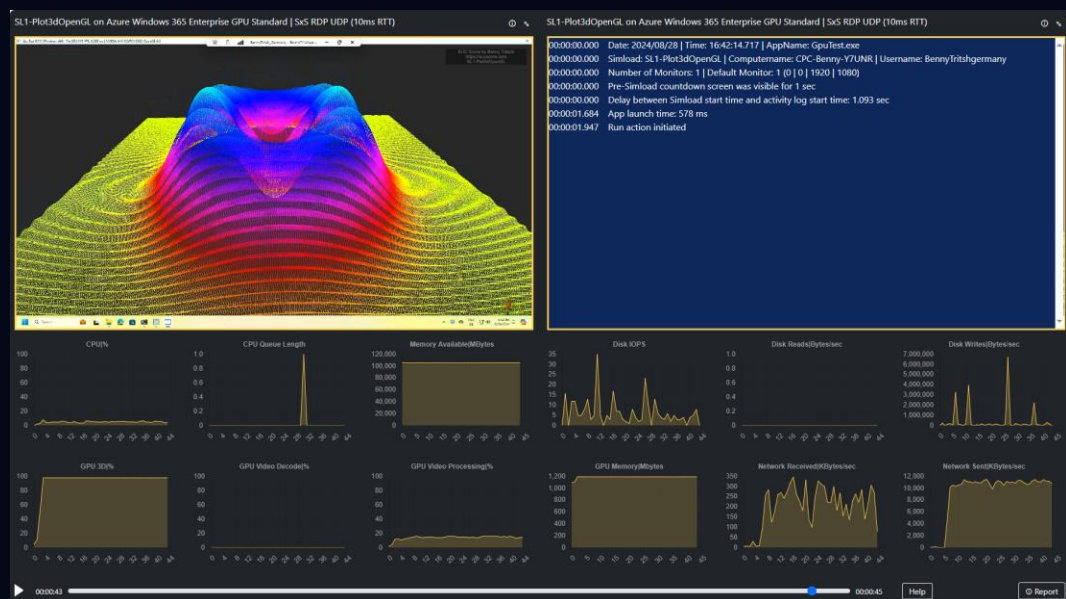
TC-specific Metrics

```
<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>
```



Why Performance Counters Matter



EUC Score Sync Player Clips



Call to Action

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

info@eucscore.com



<https://eucscore.com>

<https://eucscore.com/results>

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





EUC Score Links

<https://eucscore.com>



Home Page

<https://eucscore.com/freeware>



Freeware Download

- Blog articles: <https://drtritsch.com>
- Toolset documentation: <https://docs.eucscore.com>
- Test Methodology: <https://eucscore.com/methodology.html>
- Simload Gallery: <https://eucscore.com/gallery.html>
- Test Results (Sync Player): <https://eucscore.com/results>
- Terminology (Glossary): <https://eucscore.com/terminology.html>
- Lab Equipment: <https://eucscore.com/equipment.html>

Dr. Benny Tritsch | Independent Performance Data Scientist
info@eucscore.com | <https://drtritsch.com> | [linkedin.com/in/drtritsch](https://www.linkedin.com/in/drtritsch)



Thank You!



Mastering Windows Performance Counters for AVD and Cloud PCs

Dr. Benny Tritsch
Independent Performance
Data Scientist



Workplace Ninjas
