

Projektdokumentation

**-**

**System Metrics Script**

Das "System Metrics Script" überwacht und berichtet über CPU-, Speicher- und Festplattenauslastung, um potenzielle Probleme im System zu erkennen.

-

Gemacht mit:

**PowerShell | Visual Studio Code | Editor**

A screenshot of a computer

Description automatically generated with medium confidence

**Inhaltsverzeichnis**

**Betriebsdokumentation 3**

Funktion des Skripts 3

Auslöser des Skriptes 3

Aufruf des Skriptes 3

Benötigte Berechtigungen 3

Zielsystem 3

-

**Entwicklerdokumentation 4-6**

Anforderungen 4

PAP 5

Testfälle 6

-

**Testdokumentation 7**

Testprotokoll 7

Fazit 7

Kommentar 7

Quellenangaben 7

-

**Betriebsdokumentation**

**Die Funktion des Skripts:**

Das PowerShell-Skript überwacht die System Metriken wie CPU-Auslastung, Speichernutzung und Festplattenauslastung. Es legt Schwellenwerte fest und generiert einen Bericht mit aktuellen Werten. Wenn die Auslastung bestimmte Schwellenwerte überschreitet, werden Warnungen im Bericht hinzugefügt. Zusätzlich werden die Prozesse mit der höchsten Speicher- und Festplattenauslastung im Bericht aufgeführt.

Der Bericht wird in einer Textdatei gespeichert und mit dem Standardtexteditor geöffnet.

Das Skript ermöglicht die Überwachung von Systemressourcen und hilft dabei, Engpässe oder Überlastungen frühzeitig zu erkennen. Es kann für Automatisierungs- oder Überwachungssysteme verwendet werden, um regelmäßig Berichte zu generieren, die aktuellen System Metriken zu analysieren und die Prozesse mit der höchsten Auslastung zu identifizieren.

**Auslöser/Trigger des Skripts:**

Manuell in PowerShell/Visual-Studio

**Aufruf des Skripts:**

Um das Skript aufzurufen, müssen Sie die Skriptdatei mit der PowerShell ausführen

**Benötigte Berechtigungen:**

Um das Skript auszuführen, werden Administratorrechte und die Möglichkeit zum Ausführen von WMI-Abfragen benötigt.

**Zielsystem des Skripts**

**Server:** Local

**Client:** Client-System (User Laptop/PC)

**PowerShell Version:** 5.1.22621.1778

**Version:** 1.0.0

**Das Skript**

<#

.SYNOPSIS

   System Metrics Script

.DESCRIPTION

   This script generates a report of system metrics including CPU usage, disk usage, and memory usage.

.NOTES

   Author: Cyril Lutziger

   Version: 2.1

   Date: 01.07.

#>

# Define the system metrics

$cpuThreshold = 80   # CPU usage in percentage

$memoryThreshold = 80   # Memory usage in percentage

$diskThreshold = 80   # Disk usage in percentage

# Function to generate the system metrics report

function GenerateSystemMetricsReport {

    # Get system metrics

    $cpuUsage = (Get-WmiObject -Class Win32\_Processor | Measure-Object -Property LoadPercentage -Average).Average

    $memoryUsage = Get-WmiObject -Class Win32\_OperatingSystem | Select-Object -ExpandProperty FreePhysicalMemory

    $diskUsage = Get-WmiObject -Class Win32\_LogicalDisk | Where-Object { $\_.DeviceID -eq 'C:' } | Select-Object -ExpandProperty FreeSpace

    # Calculate memory and disk usage percentages

    $totalMemory = (Get-WmiObject -Class Win32\_OperatingSystem).TotalVisibleMemorySize

    $totalDiskSpace = (Get-WmiObject -Class Win32\_LogicalDisk | Where-Object { $\_.DeviceID -eq 'C:' }).Size

    $memoryUsagePercent = [math]::Round(($totalMemory - $memoryUsage) / $totalMemory \* 100, 2)

    $diskUsagePercent = [math]::Round(($totalDiskSpace - $diskUsage) / $totalDiskSpace \* 100, 2)

    # Create the report

    $report = @"

System Metrics Report

----------------------------------

CPU usage: $cpuUsage%

----------------------------------

Disk usage: $diskUsagePercent%

----------------------------------

Memory usage: $memoryUsagePercent%

"@

    # Check CPU usage

    if ($cpuUsage -ge $cpuThreshold) {

        $report += @"

CPU usage is high! Current usage: $cpuUsage%

"@

    }

    # Check disk usage

    if ($diskUsagePercent -ge $diskThreshold) {

        $report += @"

Disk usage is high! Current usage: $diskUsagePercent%

"@

        # Get processes with highest disk usage

        $topDiskProcesses = Get-Process | Sort-Object -Property DiskUsage -Descending | Select-Object -First 5

        $report += @"

Processes with highest disk usage:

$($topDiskProcesses | Format-Table -AutoSize | Out-String)

"@

    }

    # Check memory usage

    if ($memoryUsagePercent -ge $memoryThreshold) {

        $report += @"

Memory usage is high! Current usage: $memoryUsagePercent%

"@

        # Get processes with highest memory usage

        $topMemoryProcesses = Get-Process | Sort-Object -Property WorkingSet -Descending | Select-Object -First 5

        $report += @"

Processes with highest memory usage:

$($topMemoryProcesses | Format-Table -AutoSize | Out-String)

"@

    }

    return $report

}

# Generate the system metrics report

$systemMetricsReport = GenerateSystemMetricsReport

# Define the editor name

$editorName = "SystemMetricsReport-" + (Get-Date -Format "yyyy-MM-dd-HH-mm-ss")

# Save the report to a file

$reportPath = Join-Path -Path $env:TEMP -ChildPath "$editorName.txt"

$systemMetricsReport | Out-File -FilePath $reportPath -Encoding UTF8

# Open the report in an editor

notepad.exe $reportPath

**-**

**Entwicklerdokumentation**

**Anforderungen des Skripts:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Verbindlichkeit** | **Typ** | **Beschreibung** |
| 1 | Muss | Funktional | Erfassen und Darstellen von System Metriken wie CPU-Auslastung, Speichernutzung und Festplattenauslastung. |
| 2 | Muss | Funktional | Überwachen der CPU-Auslastung mit einem Schwellenwert von 80 %. |
| 3 | Muss | Funktional | Überwachen der Speichernutzung mit einem Schwellenwert von 80 %. |
| 4 | Muss | Funktional | Überwachen der Festplattenauslastung mit einem Schwellenwert von 80 %. |
| 5 | Muss | Funktional | Erstellen eines System Metriken-Berichts mit aktuellen Werten. |
| 6 | Muss | Funktional | Anzeigen von Warnungen im Bericht bei Überschreitung der Schwellenwerte. |
| 7 | Muss | Funktional | Auflisten der Prozesse mit höchster Festplatten- und Speicherauslastung im Bericht. |
| Q1 | Kann | Qualität | Ausführung in geeigneter Umgebung mit PowerShell und WMI-Unterstützung. |
| Q2 | Kann | Qualität | Verwendung von Administratorrechten und WMI-Berechtigungen für den Benutzer. |
| Q3 | Kann | Qualität | Ausführung einer Windows oder anderen Message Box, falls es über 90% des Schwellenwertes tritt. |
| 8 | Muss | Funktional | Speichern des Berichts in einer Textdatei und Öffnen mit dem Standardtexteditor. |
| Q4 | Kann | Qualität | Eingabefehler sollen abgefangen werden |
| Q5 | Kann | Qualität | Das Skript ist mit Kommentaren und einem Header versehen. |
| Q6 | Kann | Qualität | Es existiert eine Entwicklerdokumentation und eine Betriebsdokumentation. |
| R1 | Muss | Randbedingung | Das Skript muss innerhalb der vereinbarten Zeit fertiggestellt sein (inkl. Tests). |
| R2 | Muss | Randbedingung | Das Skript muss in PowerShell (mind. Version 5.1) geschrieben sein. |
| R3 | Muss | Randbedingung | Fremder Code und Fremdhilfe sind deklariert. |

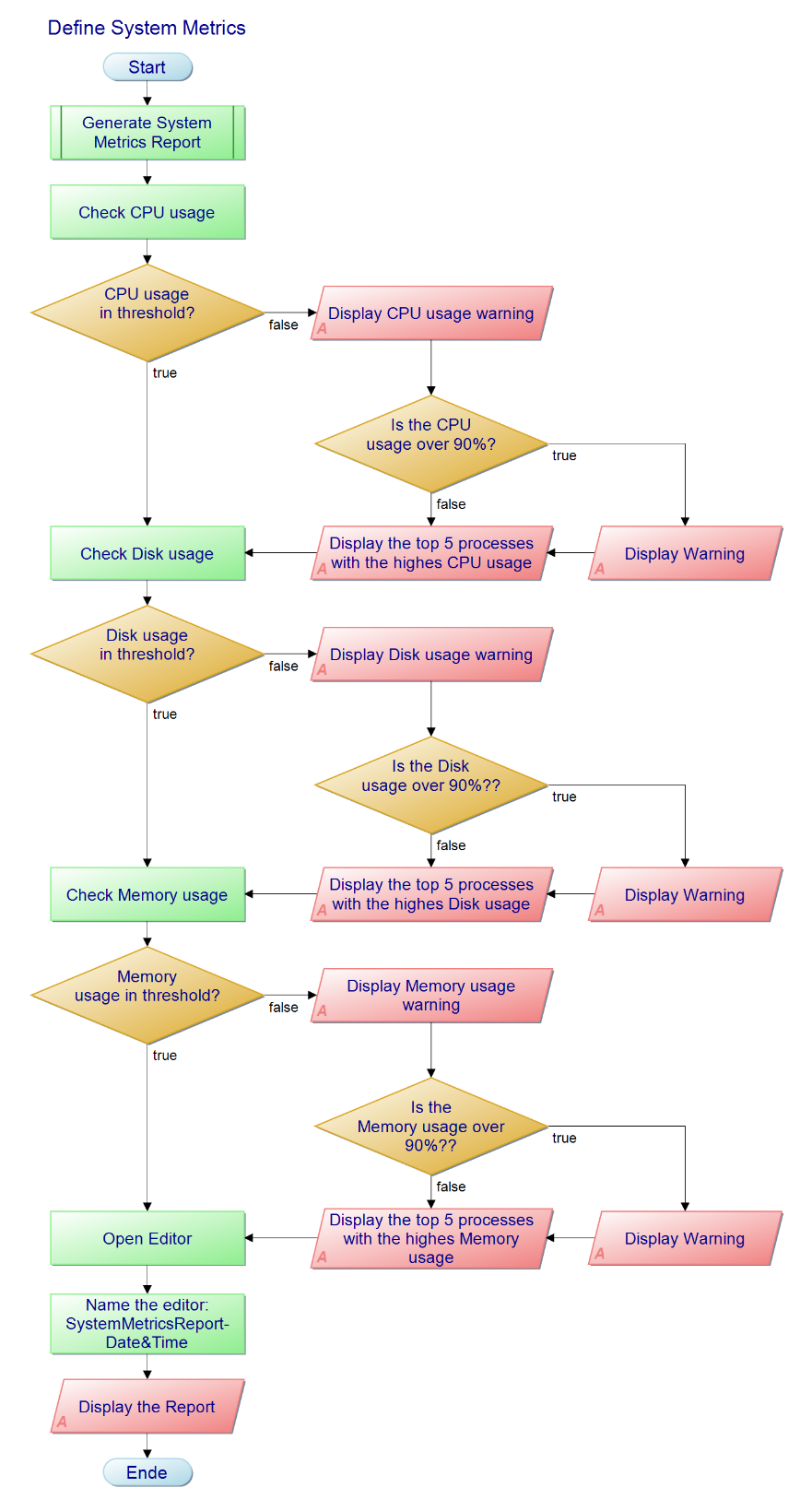
**Verbindlichkeit:**

* Muss |Angabe muss vorhanden sein
* Kann | Angabe kann vorhanden sein

**Typ:**

* Funktional | Angabe ist für die Funktionalität wichtig
* Qualität | Angabe ist für das Ästhetische hier und nicht wichtig

**Das PAP (Programmablaufplan):**



**Testfälle des Skripts:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TC-№** | **Ausgangslage** | **Eingabe** | **Erwartete Ausgabe** |
| 1 | Das System hat eine CPU-Auslastung von 70% und eine Speichernutzung von 60%. | - | System Metrics Report  ----------------------------------  CPU usage: 70%  ----------------------------------  Disk usage: 0%  ----------------------------------  Memory usage: 60% |
| 2 | Das System hat eine CPU-Auslastung von 90% und eine Speichernutzung von 80%. | - | System Metrics Report  ----------------------------------  CPU usage: 90%  ----------------------------------  Disk usage: 0%  ----------------------------------  Memory usage: 80%  Memory usage is high! Current usage: 80%  Processes with highest Memory usage:  Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName  ------- ------ ----- ----- ------ -- -- ----------- |
| 3 | Das System hat eine Festplattenauslastung von 70%. | - | System Metrics Report  ----------------------------------  CPU usage: 0%  ----------------------------------  Disk usage: 70%  ----------------------------------  Memory usage: 0% |
| 4 | Das System hat eine Festplattenauslastung von 90%. | - | System Metrics Report  ----------------------------------  CPU usage: 0%  ----------------------------------  Disk usage: 90%  ----------------------------------  Memory usage: 0%  Disk usage is high! Current usage: 90%  Processes with highest Disk usage:  Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName  ------- ------ ----- ----- ------ -- -- ----------- |
| 5 | Das System hat eine CPU-Auslastung von 80%, eine Speichernutzung von 70% und eine Festplattenauslastung von 60%. | - | System Metrics Report  ----------------------------------  CPU usage: 80%  ----------------------------------  Disk usage: 60%  ----------------------------------  Memory usage: 70%  CPU usage is high! Current usage: 80%  Memory usage is high! Current usage: 70%  Processes with highest CPU usage:  Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName  ------- ------ ----- ----- ------ -- -- ----------- |

**Testdokumentation**

**Testprotokoll**:

|  |  |  |  |
| --- | --- | --- | --- |
| **TC-№** | **Datum** | **Resultat** | **Tester** |
| 1 | 12.06.23 | Fehlgeschlagen | Cyril Lutziger |
| 1.2 | 12.06.23 | Erfolgreich | Cyril Lutziger |
| 1.3 | 12.06.23 | Erfolgreich | Cyril Lutziger |
| 2.1 | 12.06.23 | Erfolgreich | Cyril Lutziger |
| 2.2 | 12.06.23 | Erfolgreich | Cyril Lutziger |
| 3.1 | 19.06.23 | Erfolgreich | Marek von Rogall / Cyril Lutziger |
| 3.2 | 19.06.23 | Erfolgreich | Lorenzo Lai / Cyril Lutziger |
| 3.3 | 19.06.23 | Erfolgreich | Cyril Lutziger |
| 4.1 | 19.06.23 | Erfolgreich | Cyril Lutziger |
| 4.2 | 19.06.23 | Erfolgreich | Cyril Lutziger |
| 4.3 | 26.06.23 | Erfolgreich | Cyril Lutziger |
| 4.4 | 26.06.23 | Erfolgreich | Lorenzo Lai / Cyril Lutziger |
| 5.1 | 26.06.23 | Erfolgreich | Marek von Rogall / Cyril Lutziger |
| 5.2 | 26.06.23 | Erfolgreich | Lorenzo Lai / Cyril Lutziger |

**Fazit:**

Das Skript erfüllt seine Aufgabe effektiv und zuverlässig, indem es die Systemressourcen überwacht. Es ist gut strukturiert und leicht verständlich, mit klaren Kommentaren. Die erfassten System Metriken sind korrekt und die Schwellenwerte angemessen definiert. Bei Überschreitung der Schwellenwerte werden angemessene Maßnahmen ergriffen. Der Bericht enthält alle relevanten Informationen, wird erfolgreich gespeichert und ordentlich benannt. Bisher wurden keine Fehler oder Probleme festgestellt. Zusammenfassend ist das Skript eine nützliche und zuverlässige Lösung zur Überwachung der Systemleistung und Ressourcennutzung.

**Kommentar zu dem Projekt:**

Das Projekt mit PowerShell hat mir im Ganzen Sehr gut gefallen, jedoch sind wir ein wenig zu schnell nach meiner Meinung eingestiegen. Ich würde aber definitiv so ein Projekt nochmals machen, da es auch sehr viel Spass gemacht hatte auch wenn es manchmal verwirrend war mit PowerShell zu arbeiten.

**Quellen welche mir beim Erstellen des Skripts geholfen haben:**

[TextFile](https://blog.finxter.com/how-to-edit-a-text-file-in-windows-powershell/)

[MessageBox](https://adamtheautomator.com/powershell-message-box/)

[ChatGPT](https://chat.openai.com/)

[CPU Memory Disk usage](https://www.anyviewer.com/how-to/how-to-remotely-get-computer-cpu-and-memory-usage-2578.html)

**Link zum GitHub Repository:**

[GitHub](https://github.com/Cikle/M122-PowerShell/tree/main)

Inhalt:

Skript, Lernbericht, Word Dokument, Portfolioeintrag