

Project Title : Monitoring and Performance Tracking

Microsoft Certified Solutions Associate

Author: Bhavesh Mayekar ([linkedin.com/in/bhaveshmayekar](https://www.linkedin.com/in/bhaveshmayekar))

Course: Microsoft Certified Solutions Associate

Date: 2025-09-12

Objective

The goal of this project is to implement performance monitoring and tracking on a Windows Server environment.

By collecting operating system baselines and operational metrics, administrators can ensure:

- **Proactive capacity planning**
 - **Early detection of issues**
 - **Optimized server performance**
-

Why Performance Monitoring Matters

- **Proactive Monitoring: Detects bottlenecks before they impact users.**
 - **Baseline Comparison: Helps identify deviations from normal behavior.**
 - **Capacity Planning: Supports data-driven scaling decisions.**
 - **Troubleshooting: Provides measurable insights for diagnosing issues.**
-

Technologies Used

- **Performance Monitor (perfmon.msc)**
- **Data Collector Sets (DCS)**
- **Performance Counters → CPU, Memory, Disk, Network**

- **CSV/BLG Log Files for reporting and analysis**
- **Windows Server (2019 / 2022 / 2025)**

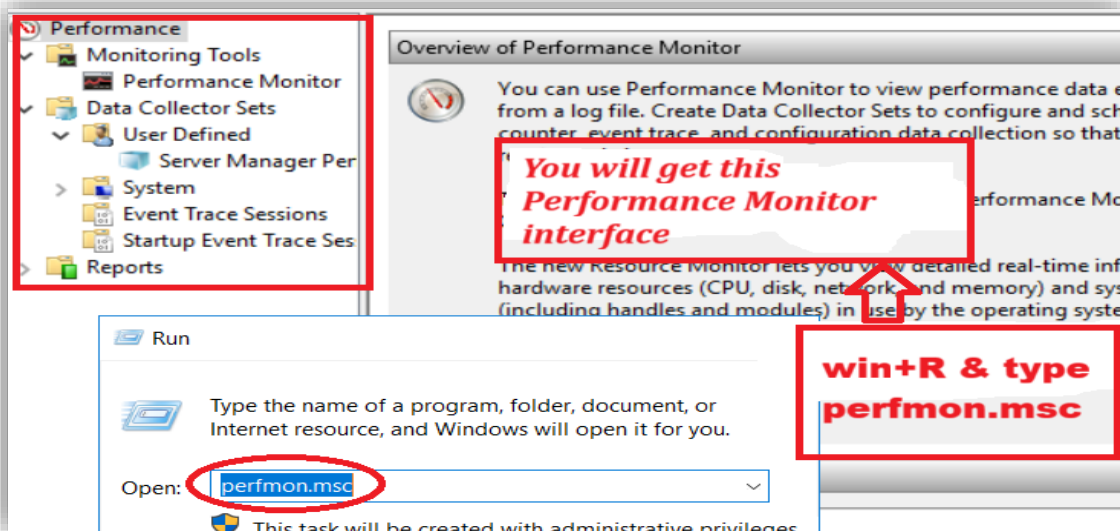
Lab Environment Setup

- **Server 1 (Target Machine):** Windows Server configured with standard roles (lab or production).
 - **Client (Admin Workstation):** Runs Performance Monitor GUI.
 - **Storage Path:** Local or network folder dedicated for storing performance logs.
-

Step-by-Step Implementation

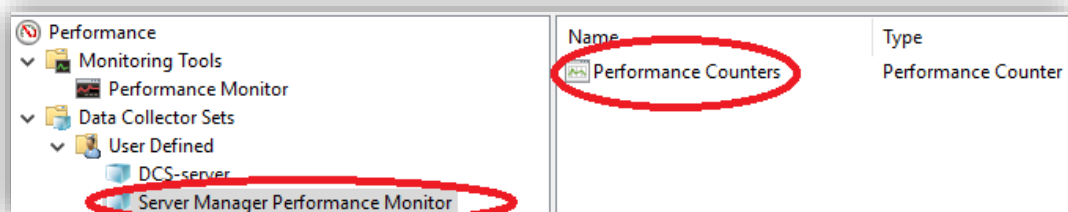
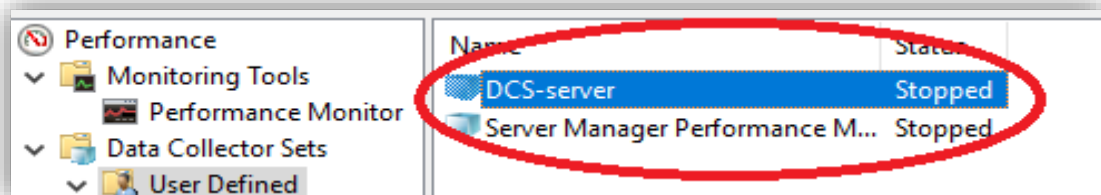
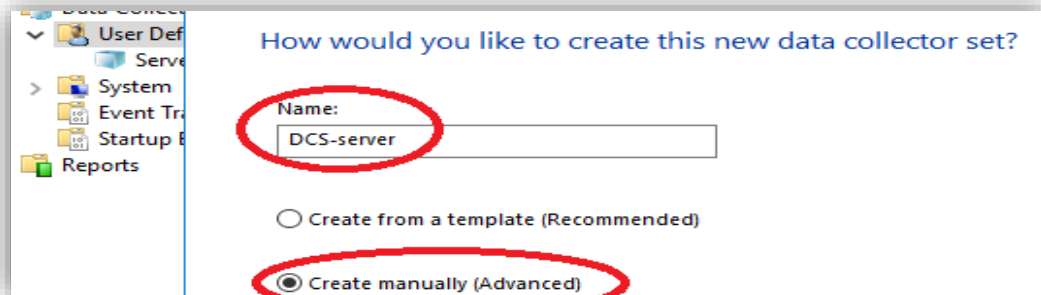
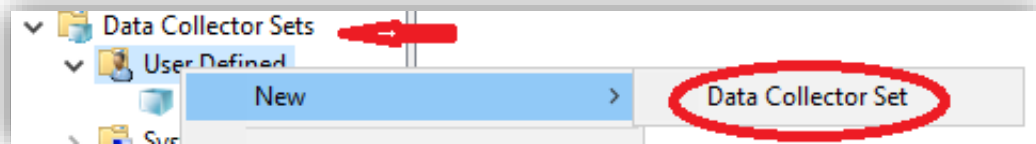
Step 1 — Launch Performance Monitor

1. Press **Win + R**, type **perfmon.msc**, Enter.
2. In the left pane expand **Performance Monitor** → **Data Collector Sets** → **User Defined**.
3. Right-click **User Defined** to create a new set.



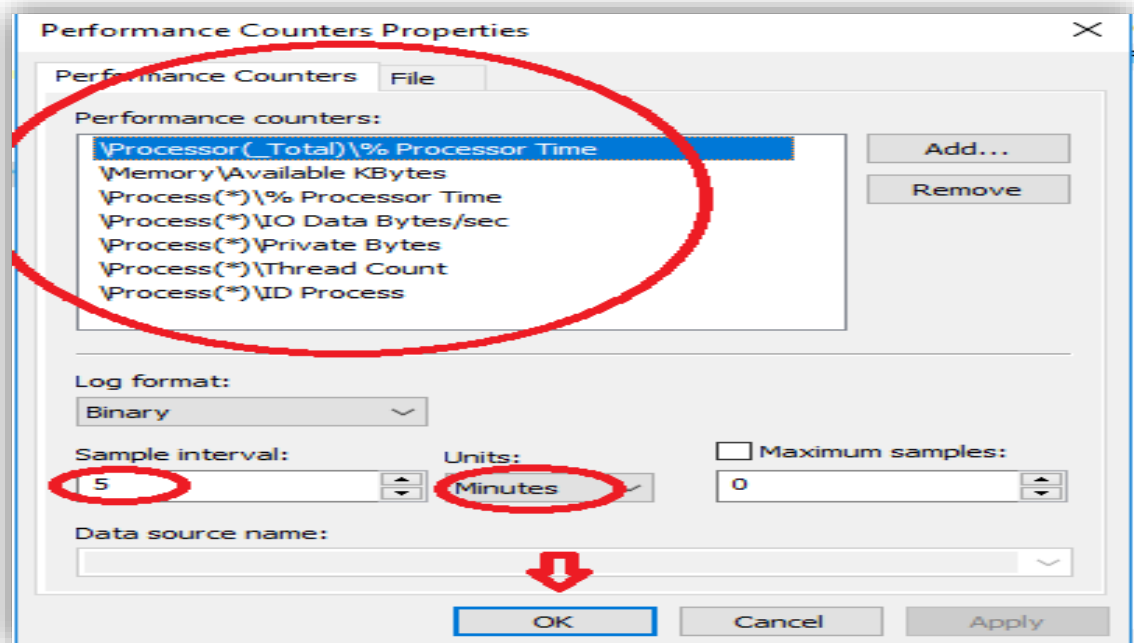
Step 2 — Create a new Data Collector Set

1. Right-click **User Defined** → **New** → **Data Collector Set**.
2. **Name** the set (use a clear name, e.g. DCS_Server1_Baseline).
3. Select **Create manually (Advanced)** → **Next**.
4. Check **Performance counter** (you can also add Event Trace or Performance Alert if needed) → **Next**.



Step 3 — Add Performance Counters

1. Click **Add...** to open the counters dialog.
2. Add these recommended counters (use **_Total** instance if you want overall server metrics):
 - **Processor** → **_Total** → **% Processor Time**
 - **Memory** → **Available MBytes**
 - **PhysicalDisk** → **_Total** → **% Disk Time**
 - **PhysicalDisk** → **_Total** → **Avg. Disk Queue Length**
 - **Network Interface** → **<your NIC>** → **Bytes Total/sec**
3. Optional (role-specific): **Process** → **% Processor Time (sqlservr)**, **LogicalDisk** → **% Free Space**, **Paging File** → **% Usage**.
4. Set **Sample interval** (seconds) at the bottom of the dialog: **15–60 seconds** recommended (15s for detailed short captures; 60s for long-term baselines).
→ **OK** → **Next**



Step 4 — Choose Log Format and Storage Location

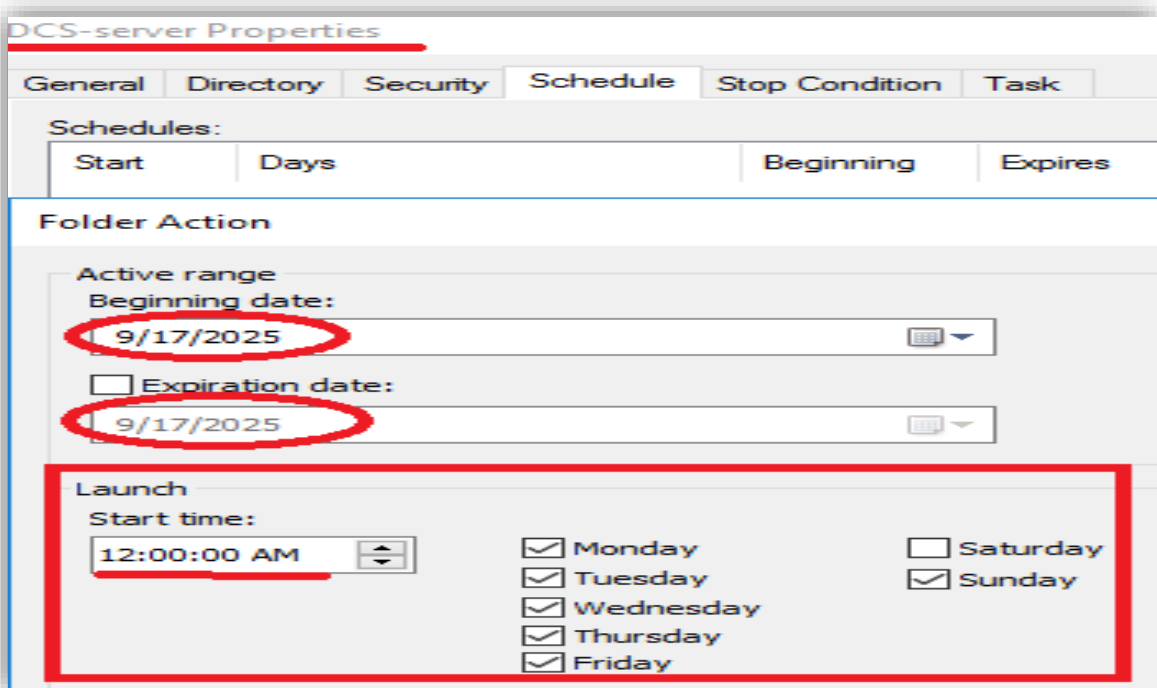
1. Select a **log format**: **BLG** (binary performance log) for high-fidelity/perfmon analysis; **CSV** if you want easy Excel import.

- **Recommendation:** use **BLG** for collection, convert/export to CSV for Excel/reporting as needed.
- 2. Specify **log folder** and filename pattern. Example:
D:\PerfLogs\Server1\DCS_Server1_Baseline\ or \\fileserver\PerfLogs\.
- 3. **Ensure the folder exists** and that the account which will run the DCS has **write permissions**. Plan storage size (logs can grow quickly).

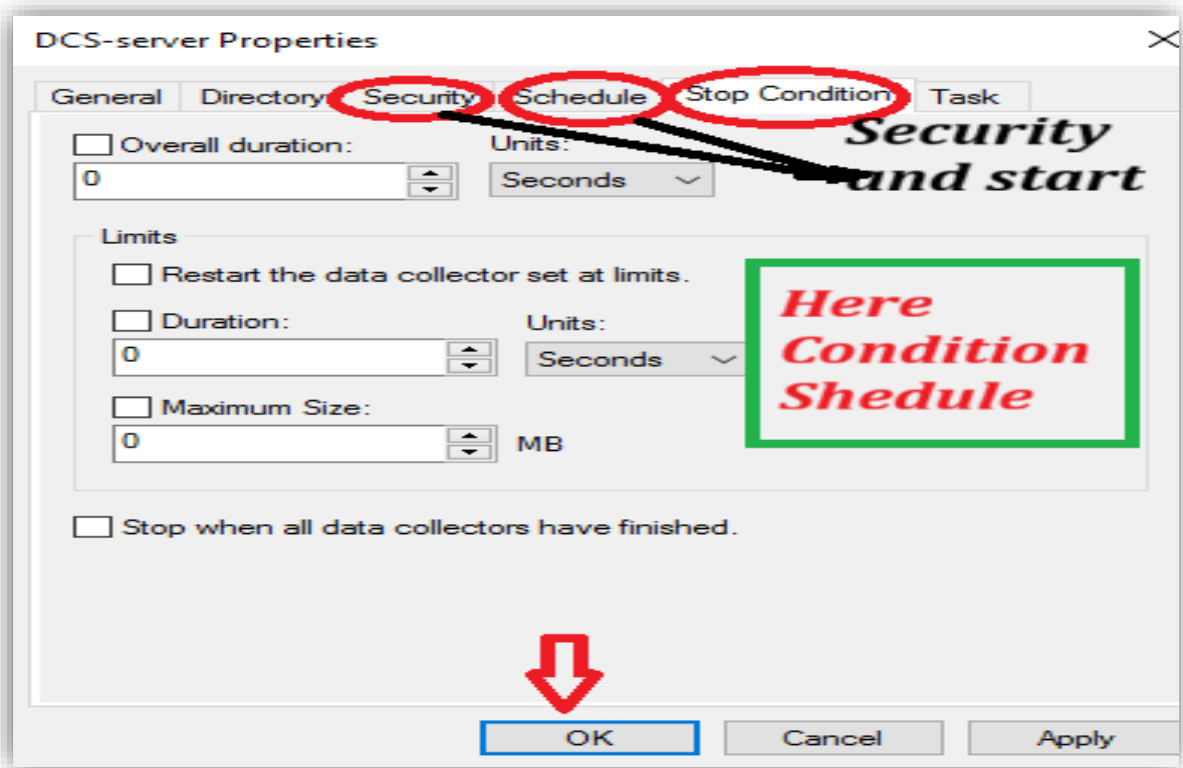


Step 5 — Configure Run As, Schedule & Stop Conditions

1. In the wizard or immediately after creation, set the **Run As** account (if needed) — use an account with **local admin** on the target server for remote collection.
 - You can set credentials during the wizard or later: **right-click the DCS → Properties → (Run As / Security / Credentials)**.

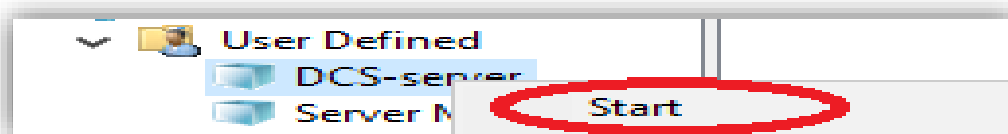


1. Add **Schedule** (Right-click DCS → **Properties** → **Schedule** or via the wizard):
 - For baseline: schedule to run **24–72 hours** (start overnight and capture a few business cycles).
 - For continuous monitoring: schedule **Start at system startup**.
2. Configure **Stop Condition** (Properties → Stop Condition): set **maximum duration** or **maximum file size** to prevent disk exhaustion (e.g., 7 days or 20 GB).



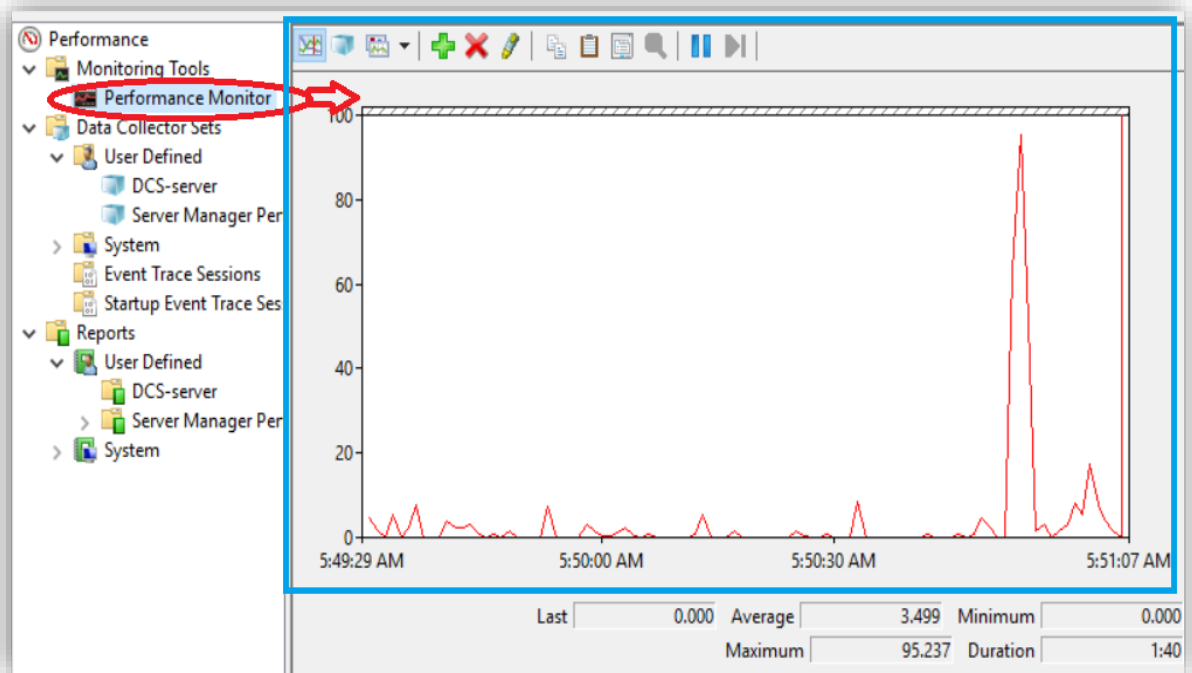
Step 6 — Start the Data Collector Set

1. Right-click your DCS → **Start**.
2. Confirm status changes to **Running** in the right pane and that a new log file appears in the configured folder.



Step 7 — View Collected Logs & Generate Reports

1. Method A — **Built-in report:** Expand **Reports** → **User Defined** → [**Your DCS**] and open the most recent report to view summary graphs and statistics.
2. Method B — **Open the log in Performance Monitor:** In the toolbar select "**View Log Data**" (or from Performance Monitor right-click → **View Log Data**), browse to the BLG/CSV file, and load it.
3. Export or save the data as CSV if you need Excel analysis, or use external tools (e.g., PAL — Performance Analysis of Logs) for automated scoring.



Step 8 — Analyze Baselines & Capture Test Periods

- **Baseline capture:** record **idle** (overnight) and **peak** (business hours) periods to define normal behavior. Recommended baseline window: **24–72 hours**.
- **Troubleshooting capture:** short high-frequency sampling (5–15s) for **15–60 minutes** while reproducing the issue.
- Compare baseline vs peak to identify anomalies (e.g., sustained high CPU, rising disk queue length, memory pressure).

Validation Checks :

- Ensure the Data Collector Set status is Running.
- Verify logs are generated with correct timestamps.
- Open reports → confirm CPU, Memory, Disk, and Network counters show expected baselines.

Troubleshooting Tips :

- No logs created: check folder permissions, DCS Run As credentials, and available disk space.
- Remote counters failing: ensure firewall allows necessary remote performance traffic and the Run As account is an admin on the remote host.
- Too much data / disk full: reduce sampling frequency, enable stop condition (max size), or store logs on a larger/remote disk.
- Reports look empty: confirm sampling interval and that capture actually started during test periods.

Conclusion :

This project successfully demonstrated how to use Performance Monitor and Data Collector Sets for tracking server health and performance. By collecting metrics such as CPU, Memory, Disk, and Network usage, administrators gain real-time visibility into system behavior and can create baselines for troubleshooting and capacity planning.

Such proactive monitoring is a key best practice for system administrators, ensuring that servers remain stable, efficient, and ready to scale as business needs grow.

Author: Bhavesh Mayekar ([linkedin.com/in/bhaveshmayekar](https://www.linkedin.com/in/bhaveshmayekar))

Course: Microsoft Certified Solutions Associate

Date: 2025-09-11