



# MP UNIVERSITY

WEDNESDAY OCTOBER 27

# MP University Agenda

| Time (ET)     | Session Title  | Presenter                            |
|---------------|--|--------------------------------------|
| 8:00          | Introductions & Kick-off   | Silect                               |
| 8:10 – 9:00   | Monitoring Roadmap with SCOM   | Aakash Basavaraj<br>Microsoft        |
| 9:00 – 10:00  | Management Pack Authoring Tools and Techniques   | Mike Sargent<br>Silect               |
| 10:15 – 11:00 | Monitoring web-based API's in SCOM – POC Ninja Style   | Bob Cornelissen<br>Topqore           |
| 11:00 – 11:30 | M365 MP Deep Dive  | Harold Dyck & Mike Sargent<br>Silect |
| 11:45 – 12:45 | Demo using advanced Management Pack fragments for monitoring applications: processes, ports, SQL queries, and adding automated recoveries. | Kevin Holman<br>Microsoft            |
| 12:45         | MP University Wrap-Up  | Silect                               |

# Tools and Techniques

- Quick look at available tools
- Deep dive into solving a real problem
  - Monitoring CPU overheating and fan failure
  - New WMI rules/monitors in MP Studio/Author Pro 10.3

# MP Author Free

- Free download, no fees
- Targeted at IT professionals
- Wizard driven
- Intermediate level management packs
- <https://silect.com/mp-author/>

# MP Author Pro

- Licensed product
- Install a license file
- Additional authoring over MP Author Free
- <https://silect.com/mp-author-professional/>

# MP Studio

- Licensed product
- Manages full MP lifecycle
- Full authoring (same as MP Author Pro)
- Stores MPs in SQL database
- Full version control
- Connects to one or more management servers
- <https://silect.com/mp-studio/>

# VSAE - Visual Studio Authoring Extensions

- Targeted at programmers
- Free extensions to Visual Studio
- Mostly XML editing, with IntelliSense
- Works with MPX files
- Fragments really help
- <https://www.microsoft.com/en-ca/download/details.aspx?id=30169>

# Fragments

- Watch Kevin Holman's talk later in the day
- Fragments are "templates" with a small number of replaceable parameters
- Kevin shares a large library of fragments
- Can use with VSAE using search and replace
- Can use with MP Author Pro or MP Studio using browser dialogs
- <https://github.com/thekevinholman/FragmentLibrary>



# Authoring Techniques

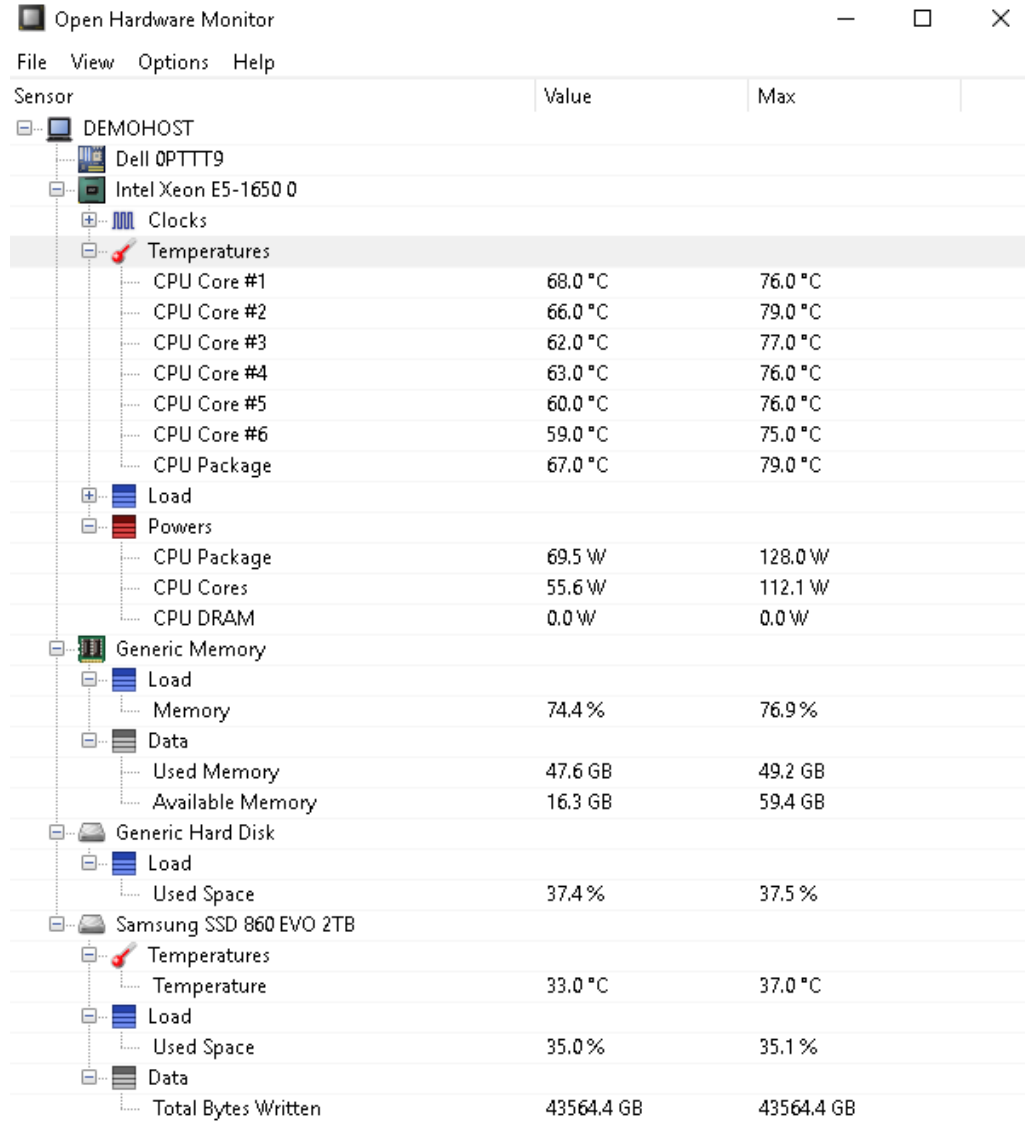
- Scenario (from real life)
  - Monitoring CPU overheating and fan failure
  - Nothing available out of the box to monitor
  - New WMI rules/monitors in MP Studio/Author Pro 10.3

# Tools

- Open Hardware Monitor
- <https://openhwaremonitor.org/>
- <https://openhwaremonitor.org/documentation/>
- Publishes data to WMI: Documentation
- <http://openhwaremonitor.org/wordpress/wp-content/uploads/2011/04/OpenHardwareMonitor-WMI.pdf>
- Options, Run On Windows Startup

# Open HW Monitor

- Shows CPU, GPU, memory, motherboards, disks, etc.
- Temperatures, Power, Fan Speeds, Voltages, Load, Data, etc.
- Mozilla Public License 2.0 (MPL).



The screenshot shows the Open Hardware Monitor application window. The window title is "Open Hardware Monitor" and it has standard Windows window controls (minimize, maximize, close). The menu bar includes "File", "View", "Options", and "Help". The main content area displays a tree view of sensors for a system identified as "DEMOHOST". The sensors are organized into categories: "Dell OPTTT9", "Intel Xeon E5-1650 0", "Clocks", "Temperatures", "Load", "Powers", "Generic Memory", "Generic Hard Disk", and "Samsung SSD 860 EVO 2TB". Each category is expanded to show its sub-sensors. The "Temperatures" category is currently selected and highlighted. The data is presented in a table with columns for "Sensor", "Value", and "Max".

| Sensor                  | Value      | Max        |
|-------------------------|------------|------------|
| DEMOHOST                |            |            |
| Dell OPTTT9             |            |            |
| Intel Xeon E5-1650 0    |            |            |
| Clocks                  |            |            |
| Temperatures            |            |            |
| CPU Core #1             | 68.0 °C    | 76.0 °C    |
| CPU Core #2             | 66.0 °C    | 79.0 °C    |
| CPU Core #3             | 62.0 °C    | 77.0 °C    |
| CPU Core #4             | 63.0 °C    | 76.0 °C    |
| CPU Core #5             | 60.0 °C    | 76.0 °C    |
| CPU Core #6             | 59.0 °C    | 75.0 °C    |
| CPU Package             | 67.0 °C    | 79.0 °C    |
| Load                    |            |            |
| Powers                  |            |            |
| CPU Package             | 69.5 W     | 128.0 W    |
| CPU Cores               | 55.6 W     | 112.1 W    |
| CPU DRAM                | 0.0 W      | 0.0 W      |
| Generic Memory          |            |            |
| Load                    |            |            |
| Memory                  | 74.4 %     | 76.9 %     |
| Data                    |            |            |
| Used Memory             | 47.6 GB    | 49.2 GB    |
| Available Memory        | 16.3 GB    | 59.4 GB    |
| Generic Hard Disk       |            |            |
| Load                    |            |            |
| Used Space              | 37.4 %     | 37.5 %     |
| Samsung SSD 860 EVO 2TB |            |            |
| Temperatures            |            |            |
| Temperature             | 33.0 °C    | 37.0 °C    |
| Load                    |            |            |
| Used Space              | 35.0 %     | 35.1 %     |
| Data                    |            |            |
| Total Bytes Written     | 43564.4 GB | 43564.4 GB |

# Demonstration

- Create a class and discover Open Hardware Monitor
- On each system with Open Hardware Monitor
  - Create and discover a class for CPUs
  - For each CPU discovered
    - Create and discover a class for temperature sensors
  - Relationship: Temperature sensor is hosted by CPU
- Create a folder to put the views under
- Create a rule to collect the temperature information
- Create a performance view
- Create a monitor to raise alerts when CPU temperature is too high

# Q&A





# Silect

**Trusted solutions.**

Maximized value.