

# Microsoft SCOM Tips & Tricks

Actionable guidance for your Microsoft SCOM operations

A Whitepaper by NiCE IT Management Solutions

---

## Overview

This document is designed to support Microsoft System Center Operations Manager (SCOM) users, architects, and administrators in their daily work. Whether you are new to the platform or have been running SCOM for years, this collection provides practical guidance to improve stability, efficiency, and operational maturity.

The tips compiled here draw from community experts, SCOM-focused blogs, Microsoft's official documentation, and the hands-on experience at NiCE. The insights are based on real-world implementations, migrations, optimizations, and troubleshooting across diverse environments.

This guide focuses on actionable recommendations you can apply directly in day-to-day operations, from governance and architecture to tuning, automation, and long-term sustainability.

Where applicable, we have included references for additional resources so you can explore topics in more depth.

We hope this collection serves as both a quick reference and a source of inspiration for continuously improving your SCOM deployment.

---

# Content

Management Pack (MP) Governance & Best Practices .....	6
1. Rename the Default Management Pack .....	6
2. Always Use Dedicated Override Management Packs .....	7
3. Implement Management Pack Version Control and Backups .....	7
4. Always Test MPs and Overrides in a Staging Environment Before Production.....	8
5. Limit Custom MP Authoring.....	9
6. Use Clear, Consistent Naming Conventions for MPs, Groups, and Overrides .....	9
7. Use Overrides Instead of Editing Management Packs .....	10
8. Document Custom Monitoring Decisions.....	11
9. Validate MP Compatibility Before Upgrades .....	11
10. Keep Management Packs Up to Date .....	12
11. Migrate Overrides When Upgrading SCOM.....	12
12. Audit Management Pack Changes with Change Tracking.....	13
13. Use Custom Management Packs for Authoring.....	13
14. Target Community MPs Explicitly.....	14
15. Review Default Rules Regularly .....	14
Overrides Strategy & Technical Debt Management.....	15
16. Review and Clean Overrides Regularly .....	15
17. Document Why Overrides Exist.....	15
18. Treat Overrides as Technical Debt.....	16
19. Review and Prune Disabled Monitors, Rules, and Objects Regularly.....	16
20. Use Groups for Targeting Overrides .....	17
21. Document Your Override Strategy .....	17
22. Visualize Override Sprawl with Power BI Sankey Diagrams.....	18
Alert Design, Noise Reduction & Actionability.....	19
23. Disable Monitors by Default After Importing MPs .....	19
24. Design Alerts for Actionability — Focus on What Operators Can Fix.....	19
25. Reduce Alert Noise Before Adding More Monitoring .....	21
26. Tune Heartbeat and Health Service Alerts .....	22

27. Regularly Review Alert Volume Trends .....	22
28. Don't Treat SCOM Alerts as Tickets.....	22
29. Suppress Duplicate or Cascading Alerts .....	23
30. Don't Ignore Warning Alerts.....	23
31. Avoid Monitoring Everything "Just in Case" .....	24
32. Use Severity Levels Consistently .....	24
33. Disable Noisy Rules Instead of Raising Thresholds .....	25
34. Avoid Alerting on Every Performance Rule .....	25
35. Tune Security Monitoring to Reduce Noise and Improve Signal Quality.....	26
36. Use Easy Tune to Reduce Alert Noise Quickly .....	27
37. Focus on High-Value Monitoring Signals .....	28
Monitoring Strategy & Operational Philosophy .....	29
38. Use Service-Centric Monitoring .....	29
39. Treat SCOM as a Living System.....	29
40. Align Monitoring With SLAs .....	30
41. Don't Over-Monitor "Green" Systems.....	30
42. Periodically Revalidate the Original Monitoring Goals.....	31
43. Avoid "Set and Forget" Monitoring .....	31
44. Measure SCOM Success by Outcomes, Not Alerts .....	32
45. Treat SCOM Updates as an Observability Improvement, Not Just Patching.....	32
Groups, Targeting & Scoping.....	33
46. Scope Views and Dashboards by Group.....	33
47. Use Dynamic Groups and Validate Group Membership for Accurate Targeting .....	33
48. Validate Group Membership Logic.....	34
49. Create Custom Dynamic Groups Based on Registry Keys .....	34
50. Populate Custom Attributes via PowerShell .....	35
Views, Dashboards & Operator Experience.....	36
51. Separate Operator and Admin Views.....	36
52. Use Health Explorer for Root Cause Analysis.....	36
53. Use Custom Views Sparingly .....	37
54. Don't Ignore Console Performance.....	37

Maintenance Mode & Operational Automation.....	38
55. Automate Maintenance Mode to Prevent Alert Noise and Improve Accuracy.....	38
56. Automate Agent Maintenance Mode via PowerShell .....	39
57. Script Maintenance Mode Based on SCCM Collections.....	39
58. Place Agents into Maintenance Mode from the Agent Computer .....	40
Performance, Scale & Platform Efficiency.....	41
59. Tune Discovery Intervals Carefully.....	41
60. Optimize Performance Data Collection .....	41
61. Prefer Agent-Based Over Agentless Monitoring .....	42
62. Avoid Using the “Management Servers Resource Pool” for Everything.....	42
63. Understand and Respect Cookdown.....	43
64. Limit Event Log Collection.....	43
65. Avoid Overusing PowerShell Script Monitors .....	44
66. Plan Event Collection Capacity.....	44
Data Retention, Grooming & Database Health.....	45
67. Regularly Groom Databases .....	45
68. Clean Up Decommissioned Objects Regularly .....	45
69. Review Data Retention Settings Regularly .....	46
70. Clean Up the SCOM Database Using Remove-SCOMDisabledClassInstance .....	46
Security, Permissions & Accounts .....	48
71. Regularly Review Run As Accounts .....	48
72. Validate Permissions After Security Hardening .....	48
73. Avoid Special Characters in SCOM Service Account Passwords .....	49
74. Enable Agent Proxy Only When Required.....	49
Platform Health & Reliability.....	51
75. Monitor the SCOM Infrastructure Itself to Ensure Reliable Monitoring.....	51
76. Use the SCOM Health Check / Assessment .....	52
77. Test SCOM Connectivity and Dependencies Early.....	52
78. Regularly Test Alert Notifications.....	53
Integrations, Reporting & Ecosystem .....	54
79. Integrate SCOM With ITSM or Dashboards.....	54

80. Use the SCOM Reporting Series to Unlock the Data Warehouse.....	54
81. Report on Availability Using State Views Instead of Alerts .....	55
82. Use Bi-Directional Integration with ITSM Tools.....	55
83. Use SCOM for Infrastructure Health and Integrate SIEM for Security.....	56
Lifecycle Operations & Change Management.....	57
84. Validate Monitoring After Patching .....	57
85. Always Review Update Rollup Fix Lists Before Applying.....	57
86. Apply Hotfixes Separately When Microsoft Releases UR Add-Ons.....	58
87. Use Community SCOM Blogs as an Early Warning System .....	58
Cross-Platform (Linux/UNIX) Tips.....	60
88. Fix Linux Agent Upgrade Failures Caused by Apache.....	60
89. Always Install Linux/AIX Agents Using the <code>-enable-opsmgr</code> Flag .....	60
90. Fix "SCOM Agent Failed During SSH Discovery. Exit Code: 1" .....	61
91. Use Proven UNIX/Linux Troubleshooting Techniques.....	61
Automation, Scripting & PowerShell Tips .....	63
92. Use Scripts to Manage Proxy-Enabled Agents .....	63
93. Monitor Robocopy and Other Logs Using NiCE MP .....	63
94. Extend Monitoring with SQL Custom Query-Based Monitors.....	64
Misc Practical / Field Tips .....	65
95. Use Scheduled Reports Sparingly.....	65
96. Understand Monitor Reset Behavior .....	65
97. Restart the Agent After Removing a Logical Disk or Filesystem.....	66
98. Resolve Multiple Critical Event Log Entries (ID 17178) After Agent Restart .....	66
99. Align SCOM Ownership Clearly .....	67
Microsoft SCOM Community Blogs & Resources .....	68
About NiCE.....	69

---

# Management Pack (MP) Governance & Best Practices

Core MP lifecycle, hygiene, and governance

---

## 1. Rename the Default Management Pack

**Level:** Intermediate (Admin)

**Versions:** All (SCOM 2012+)

**Why:**

The Default Management Pack is intended only for temporary testing and should never be used for permanent overrides or custom monitors in production. Saving overrides there is a long-standing bad practice that leads to clutter, hidden dependencies, and painful cleanup later. Many administrators unintentionally store overrides in the Default MP simply because it is preselected in the console, which over time turns it into an unmanageable dumping ground.

**How:**

In the SCOM console, go to **Administration** → **Management Packs**, locate **Default Management Pack**, and rename its *Display Name* (it is unsealed, so renaming is supported) to something like **“DO NOT USE – Default MP”**. From that point forward, always create or select a dedicated unsealed MP for overrides or custom monitoring, ideally aligned to the sealed MP or workload being customized.

**When & Where:**

This should be done immediately after deploying a new SCOM management group or as an early cleanup task in existing environments. It is especially valuable in environments with multiple admins or consultants, as the renamed MP acts as a persistent visual safeguard against accidental misuse.

**Additional Resources:**

<https://kevinholman.com/2011/02/15/renaming-your-default-management-pack/>

<https://ds.squaredup.com/blog/20-operations-manager-tips-in-20-minutes/>

---

## 2. Always Use Dedicated Override Management Packs

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Mixing overrides for multiple sealed MPs into a single override MP creates tight coupling and makes troubleshooting, exporting, or migrating configurations extremely difficult. Microsoft best practice is to isolate overrides per workload or per sealed MP to keep changes traceable and portable.

**How:**

When creating overrides, always select “**New Management Pack**” and name it clearly (e.g., *Overrides – Windows Server MP*). Maintain a consistent naming convention so that override MPs can be easily identified, backed up, and migrated between environments.

**When & Where:**

This practice should be enforced whenever new MPs are introduced or tuned. It is critical in environments with separate test, acceptance, and production management groups where overrides must be promoted cleanly between tiers.

**Additional Resources:**

<https://learn.microsoft.com/troubleshoot/system-center/scom/best-practices-configure-overrides>

<https://blog.rjz.de/category/scom/>

---

## 3. Implement Management Pack Version Control and Backups

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Management Packs (MPs) define the logic, discoveries, monitors, and rules that drive SCOM. Over time, environments accumulate outdated, unused, or poorly maintained MPs.

- **Unsealed MPs** contain critical customizations that can be lost if not backed up, making recovery difficult.
- **Unused or legacy MPs** introduce unnecessary workflows, increase complexity, and degrade performance as the environment evolves.

Effective MP governance ensures stability, reduces operational risk, and keeps the monitoring footprint clean and efficient.

### How:

- **Regularly export and back up unsealed MPs** to source control or a secure repository. This protects your override logic and custom authoring from accidental deletion or corruption.
- **Perform annual reviews of all imported MPs**, identifying which are still relevant. Remove MPs that are unused, outdated, or replaced by newer versions — but only after validation.
- As part of governance cycles, review customizations, overrides, dependencies, and documentation to ensure MPs remain aligned with current operational requirements.

### When & Where:

- Immediately after major tuning cycles or MP updates
- During annual or pre-upgrade governance reviews
- When retiring technologies or decommissioning workloads

### Additional Resources:

<https://thoughtsonopsmgr.blogspot.com/>

<https://kevinholman.com/2017/07/07/scom-2012-and-2016-unsealed-mp-backup/>

---

## 4. Always Test MPs and Overrides in a Staging Environment Before Production

**Level:** Advanced (Admin)

**Versions:** All

### Why:

Testing Management Pack (MP) changes or overrides directly in production introduces unnecessary risk. A single incorrect override, faulty discovery, or poorly written script can break monitoring **at scale**, causing outages, alert storms, or blind spots. A structured, tiered approach—using **Test** → **Acceptance** → **Production**—prevents instability and ensures monitoring quality before changes reach business-critical environments.

### How:

- Maintain separate **test**, **acceptance**, and **production** SCOM environments (or management groups).
- Validate all changes in a staging or pilot environment before importing into production, including:
  - Overrides
  - Vendor or custom MPs
  - Custom scripts or discoveries

- Promote MPs through the environments using exports, ensuring each step functions as expected before proceeding.

**When & Where:**

Apply this practice **before importing new vendor MPs**, deploying large tuning changes, onboarding new technologies, or refining overrides. This is especially critical in **regulated, large, or high-availability** enterprise environments where stability is essential.

**Additional Resources:**

<https://nathangau.wordpress.com/>

---

## 5. Limit Custom MP Authoring

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Poorly written custom MPs can severely impact performance and stability if cookdown and best practices are ignored.

**How:**

Follow Microsoft MP authoring guidelines and reuse existing modules where possible.

**When & Where:**

Only when vendor MPs are insufficient and internal knowledge is strong.

---

## 6. Use Clear, Consistent Naming Conventions for MPs, Groups, and Overrides

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Consistent naming conventions make long-term management, navigation, automation, and troubleshooting significantly easier. Without a structured naming approach, SCOM environments become confusing as they grow, especially when multiple administrators contribute over time. Clear naming helps ensure override packs, groups, and views remain understandable and maintainable throughout the lifecycle of the monitoring environment.

**How:**

Apply standardized, descriptive naming patterns across all SCOM components, including:

- **Management Packs (MPs):** Use names that reflect purpose and scope (e.g., *MP Overrides – Windows – Prod*).
- **Groups:** Name groups based on technology or functional role (e.g., *Group – SQL Servers – Prod*).
- **Overrides:** Store overrides in clearly named, workload-specific override MPs and follow the same naming pattern consistently.
- **Views & Folders:** Use names that help operators quickly identify ownership, target systems, and function.

Document your standards and ensure all administrators follow them.

#### **When & Where:**

Adopt naming conventions **from the beginning of your SCOM deployment** and reinforce them consistently during onboarding, MP imports, custom authoring, and governance cycles.

#### **Additional Resources:**

<https://cireson.com/blog/scsm-management-pack-naming-best-practice/>  
<https://blog.rjz.de/category/scom/>

---

## **7. Use Overrides Instead of Editing Management Packs**

**Level:** Beginner / Intermediate

**Versions:** All

#### **Why:**

Editing sealed Management Packs (MPs) breaks supportability, complicates upgrades, and introduces unsupported configurations. Direct changes to vendor MPs can also overwrite customizations during updates and create long-term maintenance issues. Overrides provide a safe and supported way to customize monitoring without modifying vendor logic.

#### **How:**

Always apply changes via overrides stored in dedicated unsealed Management Packs rather than modifying sealed MPs directly.

Best practices include:

- Create overrides in clearly named unsealed MPs (e.g., *Overrides – Windows Server*)
- Store overrides in dedicated override MPs instead of the Default MP
- Use overrides for thresholds, disabling monitors, tuning rules, and scoping behavior
- Never edit sealed MPs directly

#### **When & Where:**

This rule applies universally across all SCOM environments and scenarios, especially:

- During troubleshooting when quick fixes are tempting
- When tuning vendor MPs
- When adjusting thresholds or disabling noisy monitors
- During long-term governance and upgrade preparation

**Additional Resources:**

<https://thoughtsonopsmgr.blogspot.com/>

<https://www.nice.de/wp-content/uploads/2023/11/Microsoft-SCOM-Compendium-by-NiCE-2023Q4.pdf>

---

## 8. Document Custom Monitoring Decisions

**Level:** Intermediate

**Versions:** All

**Why:**

Years later, no one remembers *why* a monitor was disabled or a threshold changed, leading to confusion and rework.

**How:**

Use MP descriptions, naming conventions, or external documentation to explain why changes were made by whom and when.

**When & Where:**

Whenever creating custom rules, monitors, or overrides.

---

## 9. Validate MP Compatibility Before Upgrades

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Some older MPs are not compatible with newer SCOM versions and can cause instability after upgrades.

**How:**

Review MP compatibility matrices and update or remove unsupported MPs before upgrading SCOM.

**When & Where:**

Mandatory step in any SCOM upgrade project.

---

## 10. Keep Management Packs Up to Date

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Outdated MPs may contain bugs, inefficient workflows, or missing coverage.

**How:**

Regularly review vendor and Microsoft MP updates and apply them after testing.

**When & Where:**

Quarterly reviews or before platform upgrades.

---

## 11. Migrate Overrides When Upgrading SCOM

**Skill level:** Advanced (Admin)

**SCOM versions:** All upgrades (e.g., 2019 → 2022)

**Why:**

When upgrading a management group, override settings are often scattered across many MPs — sometimes even in the default MP. Capturing and migrating them manually is error prone. Cookdown highlights the benefits of tools like Easy Tune that can `capture effective overrides` and help reapply them in the new environment.

**How:**

Export all override MPs from the old environment. Use Easy Tune PRO or similar tooling to capture `effective tuning` (even if scattered across many override MPs), and tailor it to the target environment before importing into the upgraded group.

**When & Where:**

Plan this step as part of every SCOM upgrading project — between exporting MPs from the old group and importing them into the new.

**Additional Resources:**

<https://www.cookdown.com/blog/alert-tuning-for-your-upgraded-scom-environment>

---

## 12. Audit Management Pack Changes with Change Tracking

**Skill level:** Intermediate (Admin)

**SCOM versions:** 2019 UR2

**Why:**

Before SCOM 2019 UR2, there was no built-in way to see `who` changed MPs or overrides. Tracking such changes is critical for accountability, troubleshooting, and audit compliance — particularly in environments with multiple administrators. Update Rollup 2 introduced reports that show management pack installs, modifications, and override changes along with user context.

**How:**

Open the **Reporting** workspace in the SCOM console. Under the **Microsoft Generic Report Library**, find reports for `Management Pack History`, `Management Pack Objects`, and `Overrides Tracking`. Use filters like date, username, and MP name to dissect changes.

**When & Where:**

Run these reports after governance reviews, before major production changes, and as part of security audits. They are especially valuable in environments undergoing frequent tuning.

**Additional Resources:**

<https://www.cookdown.com/blog/a-quick-look-into-the-change-tracking-of-management-packs-in-scom>

---

## 13. Use Custom Management Packs for Authoring

**Level:** Intermediate (Author/Admin)

**Versions:** All

**Why:**

Combining discoveries, overrides, and custom logic in the same MP creates long-term maintenance challenges.

**How:**

Create dedicated MPs, for example:

- Custom Monitoring
- Custom Discoveries
- Custom Rules

**When & Where:**

Before creating custom monitors, discoveries, or scripts.

**Additional Resources:**

<https://www.walshamsolutions.com/technical-blog>

---

## 14. Target Community MPs Explicitly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Community MPs sometimes discover more than expected if left untargeted.

**How:**

Use precise groups and target objects explicitly.

**When & Where:**

During deployment or testing of community MPs.

**Additional Resources:**

<https://nathangau.wordpress.com/>

---

## 15. Review Default Rules Regularly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Default rules are not always appropriate for every environment.

**How:**

Audit and adjust or disable low-value rules.

**When & Where:**

Quarterly MP tuning or after new MP imports.

**Additional Resources:**

<https://nathangau.wordpress.com/>

---

# Overrides Strategy & Technical Debt Management

Override discipline, lifecycle, and long-term maintainability

---

## 16. Review and Clean Overrides Regularly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Obsolete overrides remain long after systems are decommissioned.

**How:**

Audit override MPs and remove unused management packs and their override MPs.

**When & Where:**

During quarterly or annual maintenance.

---

## 17. Document Why Overrides Exist

**Level:** Intermediate

**Versions:** All

**Why:**

Overrides without context are confusing years later.

**How:**

Use MP descriptions or external documentation to explain intent, author and date.

**When & Where:**

Whenever an override is created.

---

## 18. Treat Overrides as Technical Debt

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Every override adds complexity. Over time, excessive overrides make environments fragile and harder to upgrade.

**How:**

Periodically reassess overrides and remove those that are no longer required.

**When & Where:**

Before upgrades and during annual cleanups.

---

## 19. Review and Prune Disabled Monitors, Rules, and Objects Regularly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Static or instance-based targeting is fragile, labor-intensive, and quickly becomes inaccurate as environments evolve. Dynamic groups allow overrides, views, dashboards, and maintenance mode to automatically follow changes in your environment. This improves reliability and reduces long-term operational overhead.

**How:**

Build **dynamic groups** using discovery attributes such as OS version, registry keys, naming conventions, or OU placement.

Target overrides to these groups instead of individual servers or objects. This ensures that new systems automatically receive intended tuning and that decommissioned systems no longer receive overrides.

**When & Where:**

Use this approach whenever applying overrides to a set of servers, services, or workloads — especially in large or frequently changing environments where manual targeting is error-prone.

**Additional Resources:**

<https://michelkamp.wordpress.com/>

<https://blog.rjz.de/category/scom/>

<https://thoughtsonopsmgr.blogspot.com/>

---

## 20. Use Groups for Targeting Overrides

**Level:** Beginner (Admin)

**Versions:** All

**Why:**

Targeting overrides to specific instances is fragile. Groups allow scalable, maintainable override targeting.

**How:**

Create dynamic groups based on attributes like registry values or OU placement. Target overrides to those groups.

**When & Where:**

When tuning alerts for sets of servers or services.

**Additional Resources:**

<https://michelkamp.wordpress.com/>

---

## 21. Document Your Override Strategy

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Override strategy often becomes tribal knowledge and gets lost over time.

**How:**

Document:

- MP structure
- Naming standards
- Override policies

**When & Where:**

During governance cycles or handovers.

**Additional Resources:**

<https://kevingreeneitblog.blogspot.com/>

---

## 22. Visualize Override Sprawl with Power BI Sankey Diagrams

**Skill level:** Intermediate (Admin / Analyst)

**SCOM versions:** All

**Why:**

As SCOM environments grow, override sprawl becomes hard to reason about — especially when overrides span multiple MPs. Cookdown provides a downloadable Power BI Sankey diagram that visualizes override scope and relationships, helping admins identify tuning hotspots and unnecessary overrides.

**How:**

Download the Power BI Sankey template. Connect it to your SCOM Data Warehouse. Use filters to focus on a specific MP, group, or object. Analyze how overrides flow from MPs to targets to identify optimization opportunities.

**When & Where:**

Use this during quarterly reviews, governance sessions, or before major tuning campaigns. Great for environments with many custom overrides.

**Additional Resources:**

<https://www.cookdown.com/blog/powerbi-sankey-diagrams-for-visualizing-overrides>

---

# Alert Design, Noise Reduction & Actionability

**Alert quality, signal-to-noise, and operator trust**

---

## 23. Disable Monitors by Default After Importing MPs

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Most vendor MPs are designed to be broadly applicable, not environment-specific. Enabling everything immediately often results in alert storms, noise, and operator fatigue. This causes teams to distrust SCOM alerts altogether.

**How:**

After importing a new MP, create overrides to disable all non-critical monitors and rules. Then selectively enable only those monitors that are meaningful for your environment, applications, and support model.

**When & Where:**

This approach should be applied immediately after importing any new MP, especially infrastructure MPs (Windows, SQL, IIS). It is most effective in large environments where alert noise quickly becomes unmanageable.

**Additional Resources:**

<https://ds.squaredup.com/blog/20-operations-manager-tips-in-20-minutes/>

<https://www.nice.de/2025/09/30/reducing-alert-fatigue-in-microsoft-scom/>

---

## 24. Design Alerts for Actionability — Focus on What Operators Can Fix

**Level:** Intermediate (Operator)

**Versions:** All

**Why:**

Alert fatigue originates not from the number of alerts, but from the number of **non-actionable** alerts.

Three core problems contribute to this:

1. **Alerts without a clear owner or required action** slow down triage and increase operator frustration.
2. **Alerts for issues outside the team's control** generate noise but no meaningful response, reducing trust in monitoring.
3. **Untuned or overly broad rules** surface telemetry instead of problems, burying high-value alerts under low-value noise.

An actionability-first alert strategy improves response times, reduces wasted effort, and ensures SCOM becomes a system operators **trust** rather than **ignore**.

#### **How:**

Design alerts so that every alert answers **two critical questions**:

#### **1. Who should react?**

Assign a clear owner or team to each alert. If an alert has no natural owner or no one ever acts on it, it should be tuned, suppressed, or removed.

#### **2. What should they do?**

Each alert must include:

- Meaningful descriptions
- Clear resolution steps
- Accurate severity level
- Useful context operators can act on

#### **3. Alert only on what your team can fix**

If your team cannot take action on the issue, the alert should be:

- Disabled
- Re-routed
- Or replaced with a service-level indicator instead

#### **4. Tune alerts to surface only actionable events**

Use:

- Targeted overrides
- Filters
- Event scoping
- Dependency suppression

This ensures operators see only meaningful alerts rather than raw telemetry that cannot drive decisions.

**When & Where:**

Apply this actionability-first philosophy during:

- MP import reviews
- Override tuning sessions
- Alert hygiene cycles
- Onboarding of new applications or services
- Post-incident reviews where alert noise contributed to delays

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-heat-sheet/>

<https://nathangau.wordpress.com/>

---

## 25. Reduce Alert Noise Before Adding More Monitoring

**Level:** Intermediate

**Versions:** All

**Why:**

Adding monitoring on top of noisy alerts amplifies problems rather than solving them.

**How:**

Stabilize existing alerts first, then expand coverage.

**When & Where:**

Before onboarding new workloads.

**Additional Resources:**

<https://www.nice.de/2025/09/30/reducing-alert-fatigue-in-microsoft-scom/>

See also [33. Disable Noisy Rules Instead of Raising Thresholds](#)

---

## 26. Tune Heartbeat and Health Service Alerts

**Level:** Intermediate (Admin / Operator)

**Versions:** All

**Why:**

Heartbeat failure alerts are important, but overly aggressive thresholds can generate false positives during patching, reboots, or network hiccups, creating unnecessary noise.

**How:**

Adjust heartbeat thresholds and timeouts via overrides so that alerts reflect real outages rather than expected maintenance events.

**When & Where:**

Tune these settings early in deployment and revisit whenever patching strategies or maintenance windows change.

---

## 27. Regularly Review Alert Volume Trends

**Level:** Intermediate (Operator / Admin)

**Versions:** All

**Why:**

Gradual alert creep often goes unnoticed until alert fatigue becomes severe.

**How:**

Analyze alert trends over time and identify monitors responsible for repeated noise.

**When & Where:**

Monthly or quarterly operational reviews.

---

## 28. Don't Treat SCOM Alerts as Tickets

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

SCOM alerts indicate symptoms, not root causes. Treating every alert as a ticket often leads to duplicate work, slow resolution, and frustrated operators. Alerts should *inform* decisions, not automatically create workload.

**How:**

Define which alerts warrant ticket creation and which are informational or aggregated. Use alert suppression or correlation where appropriate before integrating with ITSM tools.

**When & Where:**

This is critical when integrating SCOM with service desks. Apply during alert design and periodically review integration rules.

---

## 29. Suppress Duplicate or Cascading Alerts

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

One failure can trigger dozens of downstream alerts, obscuring the root cause and overwhelming operators.

**How:**

Use alert suppression, dependency monitoring, and distributed applications to ensure only the *root cause* alerts are raised.

**When & Where:**

Apply after service modeling is in place and during alert noise reduction initiatives.

---

## 30. Don't Ignore Warning Alerts

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Warning alerts often indicate early signs of degradation. Ignoring them leads to critical outages that could have been prevented.

**How:**

Ensure warning alerts are meaningful and reviewed, even if they don't trigger immediate action.

**When & Where:**

In proactive operations models and SLA-driven environments.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 31. Avoid Monitoring Everything “Just in Case”

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Monitoring without purpose creates noise and wastes resources. Monitoring should always have a clear objective..

**How:**

Define monitoring goals first, then enable only what supports those goals.

**When & Where:**

During initial design and whenever scope expands.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 32. Use Severity Levels Consistently

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Inconsistent severity usage makes it impossible to prioritize incidents effectively.

**How:**

Define clear criteria for Information, Warning, and Critical alerts and enforce them across MPs.

**When & Where:**

During alert design and operational onboarding.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 33. Disable Noisy Rules Instead of Raising Thresholds

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Raising thresholds often masks real issues instead of preventing noise. Some rules offer low operational value.

**How:**

Assess alert usefulness and disable rules that are informational noise rather than endlessly tuning them.

**When & Where:**

During alert cleanups and tuning projects.

**Additional Resources:**

<https://monitoringguys.com/>

See also [25. Reduce Alert Noise Before Adding More Monitoring](#)

---

## 34. Avoid Alerting on Every Performance Rule

**Level:** Beginner (Admin)

**Versions:** All

**Why:**

Not all performance counters justify alerting. Excessive performance alerts lead to fatigue and distrust.

**How:**

Collect performance data for reporting, but alert only on actionable thresholds.

**When & Where:**

During MP tuning and baseline creation.

**Additional Resources:**

<https://kevinjustin.com/blog/tag/scom/>

---

## 35. Tune Security Monitoring to Reduce Noise and Improve Signal Quality

**Level:** Intermediate (Admin)

**Versions:** All

Security-related management packs can generate extremely high alert volumes when deployed without careful scoping. Many default or broad audit rules surface raw telemetry instead of actionable security events, leading to false positives, overwhelmed operators, and degraded SCOM performance. This consolidated tip provides a structured, multi-part approach to tuning security monitoring effectively.

---

### 1. Scope Security Event Rules Appropriately

**Why:**

Default security rules often capture wide ranges of events that are irrelevant to most environments, creating excessive alert noise.

**How:**

- Filter rules by **user**, **service**, or **server context** so only meaningful events trigger alerts.
- Focus on high-value signals such as privilege escalation, failed logons, or critical configuration changes.
- Avoid enabling rules globally without considering operational relevance.

---

### 2. Plan Audit Policies Before Enabling Them

**Why:**

Enabling broad Windows auditing without planning can generate massive event volume, overwhelming both SCOM and administrators.

**How:**

- Enable only **essential** audit events.
- Coordinate tuning with security teams to ensure SCOM alerts complement — rather than duplicate — SIEM telemetry.
- Review which events are required for compliance vs. those that create noise without value.

---

### 3. Minimize False Positives Through Targeted Tuning

**Why:**

High false-positive rates erode operator trust and can hide legitimate security incidents. This makes tuning essential for maintaining SCOM's value as a security-adjacent monitoring tool.

**How:**

- Apply filters and correlation logic to reduce unnecessary alerts.
- Scope MPs and rules to only the systems where security telemetry is needed.
- Periodically review noisy security alerts and adjust rules or disable low-value ones.

**When & Where:**

Apply this multi-part tuning philosophy:

- During **security MP deployment**
- During onboarding of new servers or applications
- During quarterly MP review cycles
- When alert queues indicate excessive security-event noise

**Additional Resources:**

<https://nathangau.wordpress.com/>

---

## 36. Use Easy Tune to Reduce Alert Noise Quickly

**Skill level:** Intermediate (Admin / Operator)

**SCOM versions:** All (2012+) but most useful in 2016+ environments

**Why:**

Manual overrides are time-consuming and inconsistent. The Easy Tune utility provides community-driven best practice tuning packs — from minimal discovery to full alerting. It speeds tuning, reduces noise, and avoids override sprawl.

**How:**

Download the Easy Tune management pack. In the SCOM console, choose the MP workload (e.g., SQL Server), select a tuning level (Discovery Only, Essential, Balanced, Full), and apply. Easy Tune auto-creates all necessary overrides based on your chosen level.

**When & Where:**

Apply Easy Tune early in a deployment to prevent unnecessary alerts. Revisit tuning whenever new MPs are imported or alert noise increases.

**Additional Resources:**

<https://www.cookdown.com/blog/introducing-easy-tune-the-new-way-to-tune-scom>

---

## 37. Focus on High-Value Monitoring Signals

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Monitoring everything creates noise and hides true issues.

**How:**

Identify top-value signals and integrate SCOM with SIEM tools for broader coverage.

**When & Where:**

During enterprise monitoring design.

**Additional Resources:**

<https://nathangau.wordpress.com/>

---

# Monitoring Strategy & Operational Philosophy

**Big-picture monitoring maturity**

---

## 38. Use Service-Centric Monitoring

**Level:** Intermediate (Operator/Admin)

**Versions:** 2012+

**Why:**

Operators care about service health, not individual components. Without service views, alerts lack context and troubleshooting takes longer.

**How:**

Create distributed applications or service maps that represent real business services and include all dependent components.

**When & Where:**

Best applied once core infrastructure monitoring is stable and for customer-facing or business-critical applications.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 39. Treat SCOM as a Living System

**Level:** All

**Versions:** All

**Why:**

SCOM degrades if left unattended.

**How:**

Continuously review MPs, alerts, and performance.

**When & Where:**

Always — SCOM requires active stewardship.

**Additional Resources:**

<https://www.nice.de/wp-content/uploads/2023/11/Microsoft-SCOM-Compendium-by-NiCE-2023Q4.pdf>

---

## 40. Align Monitoring With SLAs

**Level:** Intermediate (Admin / Operator)

**Versions:** 2012+

**Why:**

If alerts don't align with SLAs, teams spend time on low-impact issues while missing critical service breaches.

**How:**

Map monitoring thresholds and alert severity to SLA definitions.

**When & Where:**

During service onboarding and SLA reviews.

---

## 41. Don't Over-Monitor "Green" Systems

**Level:** Intermediate

**Versions:** All

**Why:**

Healthy systems don't need excessive scrutiny. Over-monitoring wastes resources and attention.

**How:**

Focus deep monitoring on unstable or business-critical systems.

**When & Where:**

After establishing baseline stability.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 42. Periodically Revalidate the Original Monitoring Goals

**Level:** All

**Versions:** All

**Why:**

Over time, monitoring drifts away from original business goals as environments evolve.

**How:**

Revisit why SCOM exists in your organization and adjust monitoring accordingly.

**When & Where:**

Annually, or after major organizational or platform changes.

**Additional Resources:**

<https://www.nice.de/wp-content/uploads/2023/11/Microsoft-SCOM-Compendium-by-NiCE-2023Q4.pdf>

---

## 43. Avoid “Set and Forget” Monitoring

**Level:** All

**Versions:** All

**Why:**

Environments evolve, but monitoring often doesn't. This leads to blind spots and obsolete alerts.

**How:**

Schedule regular monitoring reviews aligned with infrastructure and application lifecycle changes.

**When & Where:**

Ongoing operational practice.

**Additional Resources:**

<https://www.nice.de/wp-content/uploads/2023/11/Microsoft-SCOM-Compendium-by-NiCE-2023Q4.pdf>

---

## 44. Measure SCOM Success by Outcomes, Not Alerts

**Level:** All

**Versions:** All

**Why:**

The goal of SCOM is service stability and faster resolution—not high alert counts.

**How:**

Track outcomes such as reduced MTTR, improved uptime, and fewer surprise outages.

**When & Where:**

In operational reporting and management reviews.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## 45. Treat SCOM Updates as an Observability Improvement, Not Just Patching

**Skill level:** Intermediate (Admin)

**SCOM versions:** 2019+

**Why:**

SCOM URs often include improvements to monitoring quality, performance, and scalability — not just bug fixes. Viewing updates only as “maintenance” leads to missed value.

**How:**

Review UR release notes with a monitoring mindset: look for improvements in alert accuracy, agent stability, Linux monitoring, and performance counters. Communicate benefits to stakeholders.

**When & Where:**

During planning phases for upgrades or when justifying change windows to management.

**Additional Resources:**

<https://blog.topqore.com/scom-2025-update-rollup-1-full-list-of-fixes-and-improvements/>

---

## Groups, Targeting & Scoping

Dynamic groups, targeting accuracy, and scoping

---

### 46. Scope Views and Dashboards by Group

**Level:** Intermediate (Operator)

**Versions:** All

**Why:**

Global views showing “everything” overwhelm operators and make it harder to focus on what they own or support.

**How:**

Create dynamic groups (e.g., by application, environment, or support team) and scope views, dashboards, and alerts to those groups.

**When & Where:**

Most useful in NOC or operations teams where responsibilities are clearly segmented.

---

### 47. Use Dynamic Groups and Validate Group Membership for Accurate Targeting

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Dynamic groups are essential for scalable, maintainable monitoring. Unlike static groups, which require constant manual updates and quickly become inaccurate, dynamic groups automatically adapt as systems are added, removed, or reconfigured.

However, dynamic groups **must be regularly validated** to ensure their membership logic remains accurate—incorrect or outdated group criteria can silently cause overrides, monitors, and views to target the wrong systems or miss the intended ones entirely.

**How:**

- Build **dynamic groups** using reliable discovery attributes such as OS version, naming conventions, registry keys, or installed software.

- Use these groups for targeting overrides, views, maintenance mode schedules, and reporting scopes.
- Regularly **review and validate** group membership to ensure dynamic queries still function as expected—especially after changes to naming conventions, discovery logic, Active Directory structure, or MP updates.

**When & Where:**

Apply this practice broadly across your SCOM environment:

- During initial setup of any override strategy
- When onboarding new workloads or applications
- After changes to discovery logic, naming standards, or infrastructure

**Additional Resources:**

<https://michelkamp.wordpress.com/>

<https://blog.rjz.de/category/scom/>

---

## 48. Validate Group Membership Logic

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Incorrect dynamic group logic leads to missing alerts or incorrect scoping.

**How:**

Periodically review group membership and discovery rules to ensure accuracy.

**When & Where:**

After naming convention changes or discovery modifications.

---

## 49. Create Custom Dynamic Groups Based on Registry Keys

**Level:** Advanced (Admin / MP Author)

**Versions:** All

**Why:**

Static groups are brittle. Stefan Roth demonstrates how you can create dynamic groups based on registry key values, which allows highly flexible scoping for overrides, dashboards, and alerting based on runtime system properties.

**How:**

Use the Visual Studio Authoring Extensions (VSAE) to extend the Windows Computer class with a custom attribute based on registry keys and then use that attribute as a dynamic group membership rule.

**When & Where:**

Apply this when you need grouping granularity that isn't supported by default discovery criteria — e.g., grouping by configuration states or custom installed software versions.

---

## 50. Populate Custom Attributes via PowerShell

**Level:** Advanced (Admin / MP Author)

**Versions:** All

**Why:**

Dynamic groups and targeted overrides often depend on class attributes that aren't discovered by default. For servers in DMZs or with custom configuration data, registry-based discoveries may not be practical. PowerShell can be used to populate class attributes dynamically, enabling powerful grouping and targeting.

**How:**

Write a PowerShell script that updates SCOM class instance attributes using the Operations Manager SDK. Use these attributes for dynamic groups, views, and override targeting.

**When & Where:**

Useful when dynamic group logic goes beyond built-in discovery, especially for environments

**Additional Resources:**

<https://www.stefanroth.net/2014/05/04/scom-populate-attributes-through-powershell/>

---

# Views, Dashboards & Operator Experience

**Usability and console clarity**

---

## 51. Separate Operator and Admin Views

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Operators need clarity and simplicity, while admins need depth and diagnostics. Mixing both leads to confusion.

**How:**

Create role-specific views and dashboards tailored to operators versus administrators.

**When & Where:**

In environments with dedicated NOC or first-line support teams.

---

## 52. Use Health Explorer for Root Cause Analysis

**Level:** Beginner / Intermediate (Operator)

**Versions:** All

**Why:**

Many operators jump straight to alerts without understanding health rollup logic. Health Explorer provides valuable insight into which monitors are contributing to an unhealthy state.

**How:**

Train operators to open Health Explorer from alerts and review monitor state changes rather than focusing only on alert text.

**When & Where:**

Daily operational use, especially for complex or service-based alerts.

---

## 53. Use Custom Views Sparingly

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Too many custom views overwhelm users and dilute value.

**How:**

Create only views that answer specific operational questions.

**When & Where:**

During console customization and operator onboarding.

---

## 54. Don't Ignore Console Performance

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Slow console performance often indicates underlying SQL or workflow issues and impacts operator productivity.

**How:**

Monitor console responsiveness and investigate database, network, or MP-related causes.

**When & Where:**

When operators report slowness or after scale increases.

---

# Maintenance Mode & Operational Automation

## Maintenance handling and operational automation

---

### 55. Automate Maintenance Mode to Prevent Alert Noise and Improve Accuracy

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Forgetting to place systems into Maintenance Mode leads to **false alerts, alert storms, and distorted SLA reporting**. Manual activation is easy to miss during patching or planned work. Automating Maintenance Mode ensures consistent behavior, reduces operational noise, and eliminates human error.

**How:**

- Use **scheduled maintenance mode** for routine patch windows and predictable maintenance periods.
- Integrate automation using **PowerShell, Orchestrator, or scheduled tasks** to automatically start and stop Maintenance Mode.
- Apply automation especially in environments that rely on frequent deployments, patch orchestration, or infrastructure-as-code practices.

**When & Where:**

Use automated Maintenance Mode:

- During **regular patch cycles**
- During **planned outages** or deployments
- In **high-change** or **automation-driven** environments where manual updates are unreliable

**Additional Resources:**

<https://www.stefanroth.net/2017/11/26/scom-2016-start-scomagentmaintenancemode-powershell-way/>

<https://blog.tyang.org/>

See also [56. Automate Agent Maintenance Mode via PowerShell](#)

---

## 56. Automate Agent Maintenance Mode via PowerShell

**Level:** Intermediate (Operator / Admin)

**Versions:** SCOM 2016+

**Why:**

Manually placing agents into maintenance mode is error-prone and easy to forget during patching windows. Starting with SCOM 2016, you can programmatically schedule maintenance, which reduces false alerts and missed maintenance periods. Stefan Roth explains that SCOM 2016 added cmdlets to script this, giving automation flexibility beyond the GUI.

**How:**

Use the **Start-SCOMAgentMaintenanceMode** PowerShell cmdlet (introduced in SCOM 2016) to place agents into maintenance mode directly, including scheduling for future windows — for example from automation tools or deployment scripts.

**When & Where:**

Apply this where patching is frequent or orchestrated (e.g., monthly updates). It's especially useful when integrating maintenance mode into broader CI/CD or configuration workflows in hybrid infrastructures.

**Additional Resources:**

<https://www.stefanroth.net/2017/11/26/scom-2016-start-scomagentmaintenancemode-powershell-way/>

<https://blog.tyang.org/>

---

## 57. Script Maintenance Mode Based on SCCM Collections

**Level:** Intermediate (Admin / Automation)

**Versions:** All

**Why:**

SCOM maintenance mode improves alert accuracy during planned patching. Stefan Roth provides a script that ties maintenance mode scheduling to SCCM collections so that when SCCM pushes updates and reboots, SCOM automatically places those same systems into maintenance mode, reducing noise and manual steps.

**How:**

Use a WMI connection to the SCCM server to retrieve target collection members and then schedule maintenance mode for those systems via script.

**When & Where:**

Use this when SCCM/MECM is your primary patching tool and you want synchronized SCOM maintenance mode with minimal manual intervention.

**Additional Resources:**

<https://www.stefanroth.net/2011/12/30/scom-2012-maintenance-mode-script-ndash-computer-maintenance-mode-depending-on-sccm-2012-collection-membership/>

---

## 58. Place Agents into Maintenance Mode from the Agent Computer

**Skill level:** Intermediate (Operator / Admin)

**SCOM versions:** 2016 and later

**Why:**

Traditionally, maintenance mode was initiated from the SCOM console. SCOM 2016 introduced the ability to `trigger maintenance mode on an agent directly` via PowerShell. This helps reduce friction and makes it easier for local operators to silence alerts during planned work.

**How:**

Ensure the **Agent Initiated Maintenance Mode Rule** is enabled via override. Then, on the monitored machine, import the SCOM PowerShell module and run:

```
Start-SCOMAgentMaintenanceMode -Duration
```

Verify via Event Viewer (Event ID 2222) and in the SCOM console.

**When & Where:**

Use this when planned maintenance happens at the OS or application level and central console access isn't convenient. Ideal for field technicians or distributed environments.

**Additional Resources:**

<https://www.cookdown.com/blog/enabling-scom-maintenance-mode-from-a-monitored-computer>

---

# Performance, Scale & Platform Efficiency

## SCOM platform performance and scale optimization

---

### 59. Tune Discovery Intervals Carefully

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Discovery workflows are among the most expensive operations in SCOM. Running them too frequently consumes unnecessary CPU, memory, and network resources on agents and management servers, especially when the discovered objects rarely change.

**How:**

Review discovery rules in each MP and increase their interval (for example, from hourly to daily) unless rapid rediscovery is explicitly required. Override discovery intervals rather than modifying sealed MPs.

**When & Where:**

This tuning should be done after the initial discovery phase of a new deployment and revisited during performance optimization exercises, especially in large or highly virtualized environments.

**Additional Resources:**

<https://janscman.wordpress.com/2012/11/19/optimizing-your-management-packs-performance/>

---

### 60. Optimize Performance Data Collection

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Unoptimized performance collection can dramatically increase Operations DB and Data Warehouse size while providing little operational value. Excessive data also slows reports and increases SQL maintenance overhead.

**How:**

Use optimized collection settings, reduce sample frequency, and disable unnecessary performance counters. Only collect metrics that are actually used for alerting, dashboards, or capacity planning.

**When & Where:**

Apply this tuning once baseline monitoring is established and during periodic database growth reviews. It is particularly important in environments with long data retention requirements.

**Additional Resources:**

<https://janscman.wordpress.com/2012/11/19/optimizing-your-management-packs-performance/>

---

## 61. Prefer Agent-Based Over Agentless Monitoring

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Agentless monitoring relies heavily on remote calls from management servers, which does not scale well and creates blind spots. Agent-based monitoring provides better performance, reliability, and richer monitoring data.

**How:**

Deploy the SCOM agent wherever possible and reserve agentless monitoring only for corner cases such as network devices or systems where agents are explicitly unsupported.

**When & Where:**

This decision should be made during initial design and revisited when onboarding new platforms. It is especially relevant in large Windows server environments.

**Additional Resources:**

<https://learn.microsoft.com/system-center/scom/manage-agentless-monitoring>

<https://www.nice.de/wp-content/uploads/2023/11/Microsoft-SCOM-Compendium-by-NiCE-2023Q4.pdf>

---

## 62. Avoid Using the “Management Servers Resource Pool” for Everything

**Level:** Advanced (Admin)

**Versions:** 2012+

**Why:**

By default, many workflows target the *All Management Servers Resource Pool*, which can cause unnecessary load on every management server. This reduces scalability and can introduce instability when a single workflow misbehaves.

**How:**

Create dedicated resource pools for specific workloads (e.g., network monitoring, Unix/Linux monitoring, or third-party MPs) and explicitly target workflows to those pools.

**When & Where:**

Apply this in medium to large environments, especially when onboarding MPs that rely heavily on SDK or PowerShell workflows. Review pool usage whenever performance issues arise.

---

## 63. Understand and Respect Cookdown

**Level:** Advanced (Admin / MP Author)

**Versions:** All

**Why:**

Ignoring cookdown principles leads to duplicate workflows running on every agent, significantly impacting performance.

**How:**

Design workflows so that data is collected once and shared across multiple monitors or rules using optimized modules.

**When & Where:**

Critical when authoring custom MPs or troubleshooting unexplained agent CPU usage.

---

## 64. Limit Event Log Collection

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Collecting large volumes of event logs increases database size and often provides little actionable value.

**How:**

Disable unnecessary event collection rules and focus only on events that trigger alerts or reports.

**When & Where:**

Review after importing infrastructure MPs and during database growth investigations.

---

## 65. Avoid Overusing PowerShell Script Monitors

**Level:** Advanced (Admin / MP Author)

**Versions:** All

**Why:**

PowerShell scripts are powerful but resource-intensive. Excessive or poorly optimized scripts can cause agent performance issues.

**How:**

Use native modules and workflows wherever possible. If PowerShell is required, optimize scripts and ensure proper cookdown.

**When & Where:**

When authoring custom MPs or reviewing agent CPU spikes.

---

## 66. Plan Event Collection Capacity

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

High event volume without capacity planning can overload SCOM.

**How:**

Filter events early, plan event collectors, and design scalable workflows.

**When & Where:**

During large-scale monitoring deployments.

**Additional Resources:**

<https://nathangau.wordpress.com/>

See also [54. Don't Ignore Console Performance](#)

---

# Data Retention, Grooming & Database Health

## Database sustainability

---

### 67. Regularly Groom Databases

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Without proper grooming, SCOM databases grow excessively, impacting performance and maintenance windows.

**How:**

Review and adjust grooming settings for alerts, events, and performance data according to business and compliance requirements.

**When & Where:**

Configured early in deployment and reviewed quarterly or after major monitoring scope changes.

**Additional Resources:**

<https://techcommunity.microsoft.com/t5/system-center-blog/system-center-operations-manager-assessment/ba-p/351679>

<https://learn.microsoft.com/services-hub/unified/health/getting-started-scom>

---

### 68. Clean Up Decommissioned Objects Regularly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Decommissioned servers can linger in SCOM, creating stale alerts and unnecessary database growth.

**How:**

Enable and tune grooming settings and periodically review deleted object retention in both the Operations DB and Data Warehouse.

**When & Where:**

Perform during regular maintenance cycles or after large decommissioning projects.

---

## 69. Review Data Retention Settings Regularly

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Default retention settings may not align with compliance, reporting, or capacity planning needs and can cause unnecessary database growth.

**How:**

Adjust Operations DB and DW retention based on business requirements, not defaults.

**When & Where:**

During capacity planning and compliance reviews.

---

## 70. Clean Up the SCOM Database Using Remove-SCOMDisabledClassInstance

**Skill level:** Intermediate to Advanced (Admin)

**SCOM versions:** All

**Why:**

Over time, disabled discoveries leave behind unused class instances. These can inflate the operational DB and affect performance.

**How:**

Use the PowerShell cmdlet:

```
Remove-SCOMDisabledClassInstance
```

It removes class instances associated with disabled discoveries and helps maintain a clean database.

**When & Where:**

Run during maintenance windows on large environments.

**Additional Resources:**

<https://learn.microsoft.com/en-us/powershell/module/operationsmanager/remove-scomdisabledclassinstance?view=systemcenter-ps-2025>

---

# Security, Permissions & Accounts

## Security posture and monitoring continuity

---

### 71. Regularly Review Run As Accounts

**Level:** Advanced (Admin)

**Versions:** All

**Why:**

Expired or overprivileged Run As accounts are common sources of monitoring failures and security risk.

**How:**

Audit Run As accounts for scope, permissions, and password expiry.

**When & Where:**

During security reviews and after credential changes.

---

### 72. Validate Permissions After Security Hardening

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Security hardening and GPO changes often break monitoring by restricting agent or Run As permissions.

**How:**

After security changes, validate discoveries, scripts, and Run As profiles to ensure monitoring still functions.

**When & Where:**

After security audits, hardening projects, or domain policy changes.

---

## 73. Avoid Special Characters in SCOM Service Account Passwords

**Skill level:** Intermediate to Advanced (Admin)

**SCOM versions:** All

**Why:**

SCOM interprets quotes and some symbols incorrectly during validation, causing “wrong credentials” errors even when the password is correct.

**How:**

SCOM (especially during setup and Linux agent onboarding) may fail authentication if the passwords for service accounts contain certain special characters.

Do NOT use in SCOM passwords:

\$  
&  
?  
#  
@  
"  
,

**When & Where:**

Use strong passwords without those characters.

---

## 74. Enable Agent Proxy Only When Required

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Unnecessary proxying can cause duplicate discoveries and increase security risks.

**How:**

Enable agent proxy only for systems running:

- Cluster monitoring
- Network device monitoring
- Distributed applications

**When & Where:**

When onboarding clusters or specialized workloads.

**Additional Resources:**

<https://blog.tyang.org/>

---

# Platform Health & Reliability

Keeping SCOM itself healthy

---

## 75. Monitor the SCOM Infrastructure Itself to Ensure Reliable Monitoring

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

SCOM's own health directly determines the reliability of everything it monitors. When core SCOM components (such as the SDK service, management servers, data warehouse, workflows, or agent channels) experience issues, the entire monitoring ecosystem becomes unreliable — resulting in missed alerts, delayed data, and false or stale health states.

Proactively monitoring the SCOM infrastructure ensures early detection of internal failures before they cascade into blind spots across your environment.

**How:**

Monitor SCOM using SCOM by enabling and tuning the internal management packs that track the health of:

- **Management servers** (CPU, memory, workflow failures)
- **SDK service health**
- **Data Warehouse and SQL performance**
- **Agent heartbeat stability**
- **Workflow reliability and failures**

Use Health Explorer and diagnostic views to identify root causes within the SCOM infrastructure and address underlying issues such as SQL latency, workflow failures, or management server overload.

**When & Where:**

This should be part of **continuous operational practice**, used daily and especially after:

- Upgrades or topology changes
- Infrastructure maintenance
- Performance complaints from operators
- Changes affecting SQL, storage, or network components

**Additional Resources:**

<https://techcommunity.microsoft.com/t5/system-center-blog/system-center-operations-manager-assessment/ba-p/351679>

<https://learn.microsoft.com/services-hub/unified/health/getting-started-scom>

<https://monitoringguys.com/>

---

## 76. Use the SCOM Health Check / Assessment

**Level:** Intermediate (Admin)

**Versions:** 2016+

**Why:**

Configuration drift and legacy settings accumulate over time. Microsoft's SCOM assessment identifies risks and misconfigurations early.

**How:**

Run the SCOM Assessment via Microsoft Services Hub and apply recommendations selectively.

**When & Where:**

Ideal before upgrades, during performance issues, or as part of annual health reviews.

---

## 77. Test SCOM Connectivity and Dependencies Early

**Skill level:** Intermediate to Advanced (Admin)

**SCOM versions:** All

**Why:**

Many SCOM issues don't originate in SCOM itself, but in external dependencies such as DNS, firewalls, certificates, or SQL permissions.

**How:**

Before deploying agents or management packs, make sure you validate:

- DNS forward and reverse lookups
- Port connectivity (TCP 5723, SQL ports, gateway ports)
- Certificate trust chains
- Service account permissions
- Account Distribution Security settings (use the "More secure" option)
- Credential distribution to all computers and resource pools that require them

**When & Where:**

Catching these dependencies early helps avoid long and frustrating troubleshooting sessions later on.

---

## 78. Regularly Test Alert Notifications

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Notification channels break silently due to credential, SMTP, or integration changes

**How:**

Periodically test email, webhook, or ticket notifications using test alerts.

**When & Where:**

Monthly or after infrastructure changes.

---

## Integrations, Reporting & Ecosystem

ITSM, reporting, and external value

---

### 79. Integrate SCOM With ITSM or Dashboards

**Level:** Advanced

**Versions:** All

**Why:**

Standalone monitoring limits operational value.

**How:**

Integrate with ticketing systems and visualization tools like dashboards.

**When & Where:**

Once monitoring is stable and trusted.

**Additional Resources:**

<https://squaredup.com/features/>

---

### 80. Use the SCOM Reporting Series to Unlock the Data Warehouse

**Skill level:** Beginner–Intermediate (Operator / Admin)

**SCOM versions:** All

**Why:**

Many SCOM environments underuse reporting, relying only on real-time alerts. The Data Warehouse contains valuable historical data for trend analysis, SLA reporting, and capacity planning.

**How:**

Access built-in SSRS reports, understand state vs. performance data, and schedule recurring reports for stakeholders. Use reports to identify long-term issues rather than reacting only to alerts.

**When & Where:**

Ideal for monthly service reviews, SLA validation, and long-term infrastructure planning.

**Additional Resources:**

<https://blog.topqore.com/scom-reporting-series-home/>

---

## 81. Report on Availability Using State Views Instead of Alerts

**Skill level:** Intermediate (Operator / Admin)

**SCOM versions:** All

**Why:**

Alerts alone do not accurately reflect availability. They may be closed, suppressed, or overridden. State views provide a more accurate representation of system health over time. Reporting directly from state views improves accuracy.

**How:**

Use state-based reports from the Data Warehouse instead of alert-based reports. Filter by class, group, or monitor to reflect real availability.

**When & Where:**

Use this approach for management reporting, audits, and SLA documentation.

**Additional Resources:**

<https://blog.topqore.com/scom-reporting-series-reporting-from-a-state-view/>

---

## 82. Use Bi-Directional Integration with ITSM Tools

**Skill level:** Intermediate (Admin / Operator)

**SCOM versions:** All

**Why:**

Many organizations use ITSM tools (ServiceNow, Cherwell, etc.) to manage incidents. Cookdown's `Connection Center` enables bi-directional synchronization — alerts become incidents, and incident state (resolve/close) flows back into SCOM, automatically resolving or resetting alerts. This improves SLA compliance and reduces manual coordination.

**How:**

Configure Connection Center to integrate SCOM with your ITSM of choice. Select inbound/outbound sync options, define alert filters, and map incident fields. Once enabled, closing an incident in ITSM can automatically close or reset monitors in SCOM.

**When & Where:**

Best implemented when SCOM is part of an enterprise ticketing process, reducing alert-to-ticket friction and keeping SCOM as the `single source of truth` across toolchains.

**Additional Resources:**

<https://www.cookdown.com/blog/introducing-easy-tune-the-new-way-to-tune-scom>

---

## 83. Use SCOM for Infrastructure Health and Integrate SIEM for Security

**Level:** Intermediate → Advanced (Admin)

**Versions:** All

**Why:**

SCOM excels at health monitoring, but is not a full SIEM replacement.

**How:**

Use SCOM for infrastructure alerts and integrate with SIEM for deep security analytics.

**When & Where:**

During enterprise security strategy development.

---

# Lifecycle Operations & Change Management

Ongoing operational hygiene

---

## 84. Validate Monitoring After Patching

**Level:** Intermediate

**Versions:** All

**Why:**

Patches and upgrades can break discoveries, scripts, or permissions silently.

**How:**

Spot-check monitoring health after patch cycles.

**When & Where:**

After monthly patching or application upgrades.

---

## 85. Always Review Update Rollup Fix Lists Before Applying

**Skill level:** Intermediate (Admin)

**SCOM versions:** 2016, 2019, 2022, 2025

**Why:**

SCOM Update Rollups (URs) often contain fixes that silently resolve long-standing issues such as agent failures, console crashes, SDK instability, or security gaps. Administrators frequently apply URs without fully understanding what problems they solve — or worse, delay URs unnecessarily due to fear of change. Reviewing UR fix lists helps justify updates and prevents running into already-fixed issues.

**How:**

Before installing a UR, review the published fix list and known issues. Identify fixes that apply to your environment (agents, Linux monitoring, SQL, console). Validate prerequisites and follow the documented installation order for management servers, gateways, agents, and consoles.

**When & Where:**

Do this as part of every SCOM maintenance cycle. Especially important in large or business-critical environments where update justification is required.

---

**Additional Resources:**

<https://blog.topqore.com/scom-2022-ur3-hotfix-kb5071859-whats-fixed-why-it-matters-and-how-to-check/>

---

## 86. Apply Hotfixes Separately When Microsoft Releases UR Add-Ons

**Skill level:** Intermediate (Admin)

**SCOM versions:** 2022+

**Why:**

Some SCOM URs are later followed by **standalone hotfixes** that address newly discovered or critical issues. These fixes are not always included in the next UR and may be required immediately. Missing them can leave environments unstable even after a UR update.

**How:**

Monitor official and trusted community sources for post-UR hotfix announcements. Validate whether the hotfix applies to your environment and install it according to Microsoft guidance, often without needing to wait for the next UR.

**When & Where:**

Use this approach when experiencing unexplained issues after a UR or when Microsoft explicitly recommends a hotfix for your scenario.

**Additional Resources:**

<https://blog.topqore.com/scom-2022-ur3-hotfix-kb5071859-whats-fixed-why-it-matters-and-how-to-check/>

---

## 87. Use Community SCOM Blogs as an Early Warning System

**Skill level:** Beginner–Intermediate

**SCOM versions:** All

**Why:**

Microsoft documentation often lags behind real-world issues. Community blogs and forums like <https://www.reddit.com/r/scom/> frequently publish fixes, workarounds, and insights before official guidance is updated. Leveraging these sources reduces downtime and troubleshooting time.

**How:**

Follow trusted SCOM-focused blogs and review posts after each UR release. Cross-reference issues you see in your environment with community findings.

**When & Where:**

Especially valuable immediately after updates, during unexplained issues, or when troubleshooting rare edge cases.

---

## Cross-Platform (Linux/UNIX) Tips

### Non-Windows operations

---

## 88. Fix Linux Agent Upgrade Failures Caused by Apache

**Skill level:** Intermediate (Admin)

**SCOM versions:** 2016+

**Why:**

SCOM Linux agent upgrades can fail when Apache is installed on the monitored system due to dependency or package conflicts. Failed upgrades leave agents outdated, unsupported, or partially functional — often without clear error messages. This is a repeatable and solvable issue.

**How:**

Identify Linux systems running Apache where agent upgrades fail. Adjust package dependencies as documented, then re-run the agent upgrade. Validate agent health after installation using SCOM cross-platform views.

**When & Where:**

Apply this during Linux agent upgrade campaigns or when troubleshooting stubborn agent version mismatches.

**Additional Resources:**

<https://blog.topqore.com/how-to-fix-scom-linux-agent-upgrade-failures-when-apache-installed/>

---

## 89. Always Install Linux/AIX Agents Using the `-enable-opsmgr` Flag

**Skill level:** Intermediate (Admin)

**SCOM versions:** All

**Why:**

Manual installation of Linux or AIX agents without the `-enable-opsmgr` flag will fail to properly register with the management group. This flag ensures that the agent is enabled for Operations Manager communication and can participate in discovery and monitoring.

**How:**

When performing a manual installation of a Linux/AIX agent, use the following command syntax:

```
-install -enable-opsmgr
```

Follow the official documentation for additional parameters such as management server configuration and agent proxy settings.

**When & Where:**

Previously, Linux/AIX agents could be installed without this flag, but this is no longer supported. Always apply this flag during initial agent installation, particularly in environments with multiple Unix/Linux systems.

---

## 90. Fix “SCOM Agent Failed During SSH Discovery. Exit Code: 1”

**Skill level:** Intermediate (Admin)

**SCOM versions:** All

**Why:**

SSH discovery failures with exit code 1 often occur when the root user’s default shell is set to `/usr/bin/csh`. The `csh` shell does not support the `$?` variable, which is required for SCOM’s discovery command pipeline. This prevents the agent from completing discovery of Unix/Linux systems.

**How:**

Change the default shell for the root user to `/usr/bin/sh`

After this change, rerun the SSH discovery. The agent should now successfully discover all objects.

**When & Where:**

Use this tip when SSH discovery fails for Unix/Linux systems with exit code 1 and standard error indicating “Variable syntax.” Common in environments where `csh` is the default shell for root.

**Reference:**

Internal operational knowledge / SCOM SSH discovery best practices

---

## 91. Use Proven UNIX/Linux Troubleshooting Techniques

**Skill level:** Intermediate to Advanced (Admin)

**SCOM versions:** All

**Why:**

UNIX and Linux monitoring can be one of the more challenging areas in SCOM due to certificates, sudoers files, agent logs, and timeout behaviors.

**When & Where:**

- Certificate trust
- Log locations
- Elevation issues
- Common error patterns
- Discovery problems

**Additional Resources:**

<https://blakedrumm.com/blog/scom-unix-linux-troubleshooting-tips/>

---

## Automation, Scripting & PowerShell Tips

Hands-on admin automation

---

### 92. Use Scripts to Manage Proxy-Enabled Agents

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Some management packs or monitoring scenarios require SCOM agents to be proxy-enabled. Manually checking hundreds of agents is inefficient and prone to oversight. Stefan Roth's short PowerShell script quickly identifies proxy-enabled agents at scale, improving operational hygiene.

**How:**

On a management server, run a PowerShell query against SCOM to list agents with the proxy setting enabled. Use this in regular audits or onboarding automation.

**When & Where:**

Great to use during onboarding of new agents or when troubleshooting MP behavior that depends on proxy settings (e.g., network device monitoring).

**Additional Resources:**

<https://www.stefanroth.net/2012/07/24/scom-2012-find-proxy-enabled-agents-quickly/>

---

### 93. Monitor Robocopy and Other Logs Using NiCE MP

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Many administrators need to monitor application or service logs (e.g., Robocopy job logs) that aren't covered by built-in MPs. This tip shows a practical example of using the **NiCE Log File Library Management Pack** to detect and alert on log contents.

**How:**

Import the NiCE Log File Library MP, then configure log file monitoring rules to watch for specific events or patterns (e.g., errors in log files) and generate alerts.

**When & Where:**

Useful when native MPs don't natively capture the logs you need — common for custom services or periodic tasks (like Robocopy).

**Additional Resources:**

<https://www.stefanroth.net/2014/02/24/scom-2012-nice-log-file-library-mp-monitoring-robocopy-log-file/>

---

## 94. Extend Monitoring with SQL Custom Query-Based Monitors

**Skill level:** Advanced (Admin / Author)

**SCOM versions:** All

**Why:**

Out-of-the-box SQL MPs focus on infrastructure health, not business or application logic. Many critical KPIs exist only as SQL query results. Custom SQL query-based monitors allow SCOM to alert on application-specific data, thresholds, or business rules.

**How:**

Create SQL queries that return numeric or state-based results and integrate them into SCOM as custom monitors or rules. Define thresholds carefully to avoid alert noise.

**When & Where:**

Use this when application owners request monitoring that standard MPs cannot deliver, or when SLAs depend on database-level conditions.

**Additional Resources:**

<https://blog.topqore.com/extend-your-observability-with-sql-custom-query-based-monitors/>

---

## Misc Practical / Field Tips

**Niche but valuable operational lessons**

---

### 95. Use Scheduled Reports Sparingly

**Level:** Beginner / Intermediate

**Versions:** All

**Why:**

Unnecessary scheduled reports consume resources and are often ignored by recipients.

**How:**

Review report subscriptions regularly and eliminate unused or redundant reports.

**When & Where:**

Quarterly reporting reviews.

---

### 96. Understand Monitor Reset Behavior

**Level:** Intermediate (Admin)

**Versions:** All

**Why:**

Some monitors do not automatically reset, leading to lingering unhealthy states even after issues are resolved.

**How:**

Review monitor reset logic (manual vs automatic) and adjust where appropriate, especially for state-based monitors.

**When & Where:**

When troubleshooting alerts that appear "stuck" or after MP imports.

---

## 97. Restart the Agent After Removing a Logical Disk or Filesystem

**Skill level:** Intermediate (Admin)

**SCOM versions:** All

**Why:**

When a logical disk or filesystem is removed from a server, the SCOM agent may continue to reference it. This can cause stale monitoring objects or alerts to persist in the console. Restarting the agent ensures that the management server receives an accurate, updated inventory.

**How:**

Restart the Linux/AIX agent on the Linux/AIX system after removing a logical disk / filesystem on a AIX/Linux system.

**When & Where:**

Use this whenever a logical disk or filesystem is removed from a monitored system and you notice that the object still appears in SCOM alerts or discoveries. This is particularly important in dynamic storage environments.

---

## 98. Resolve Multiple Critical Event Log Entries (ID 17178) After Agent Restart

**Skill level:** Intermediate (Admin)

**SCOM versions:** All

**Why:**

If the Operations Manager agent on a server is restarted while a monitored Unix agent is down, multiple critical event log entries (ID 17178) may be generated in the SCOM event log. This is a known one-time event and does not indicate an ongoing issue.

**How:**

Once the Unix/Linux agent comes back online, the entries are automatically reconciled. No further remediation is typically required. Optionally, review the HealthService logs to ensure no persistent errors remain.

**When & Where:**

This is observed after restarting Windows agents while remote Unix/Linux agents are temporarily unavailable. Typical event log entries may look like:

## LogFile Monitoring fails for

- DB2MP\_AdmLogFile : system.lab.de
- DB2 Instance: db2instance
- RuleID: NiCE.DB2.X.Alert.AdmLog.Monitoring.BackupInProgress

---

## 99. Align SCOM Ownership Clearly

**Level:** Intermediate (Admin / Management)

**Versions:** All

**Why:**

Unclear ownership leads to neglected tuning, slow upgrades, and monitoring decay.

**How:**

Define clear technical and operational ownership for SCOM.

**When & Where:**

At program inception and reviewed annually.

**Additional Resources:**

<https://www.nice.de/2026/01/16/microsoft-scom-cheat-sheet/>

---

## Microsoft SCOM Community Blogs & Resources

<https://blakedrumm.com/>

<https://blog.rjz.de/category/scom/>

<https://blog.tyang.org>

<https://blog.topqore.com/>

<https://cireson.com>

<https://janscman.wordpress.com>

<https://kevingreeneitblog.blogspot.com>

<https://kevinholman.com>

<https://kevinjustin.com/blog/tag/scom/>

<https://learn.microsoft.com/>

<https://michelkamp.wordpress.com>

<https://monitoringguys.com>

<https://nathangau.wordpress.com>

<https://www.opsman.co.za/tag/scom/>

<https://scomathon.com/posts>

<https://squaredup.com/>

<https://tadgata.wordpress.com>

<https://techcommunity.microsoft.com/category/system-center/blog/systemcenterblog>

<http://thoughtsonopsmgr.blogspot.com>

<https://www.cookdown.com/blog>

<https://www.nice.de>

<https://www.reddit.com/r/scom/>

<https://www.stefanroth.net>

<https://www.walshamsolutions.com/technical-blog>

---

## About NiCE

NiCE Services for Microsoft System Center encompass consulting services tailored to System Center Operations Manager, Configurations Manager, and Service Manager. Our offerings include SCOM Health Assessments, training, advice and provisioning for third-party SCOM tools, as well as SCOM-centric monitoring solutions for business elements such as applications, databases, operating systems, services, and custom applications.

**NiCE Management Packs for Microsoft SCOM** are available for AIX, Azure AD Connect, Entra ID, Citrix VAD & ADC, Custom Applications, HCL Domino, IBM Db2, IBM HMC & VIOS, IBM Power HA, Linux on Power Systems, Log Files, MariaDB, Microsoft 365, Microsoft Teams, Microsoft SharePoint, Microsoft Exchange, Microsoft OneDrive, Mongo DB, NetApp ONTAP, Oracle, Veritas Clusters, VMware, and zLinux.

### Our commitment

1. Ongoing development, incl. latest version support
2. Top required metrics come out-of-the-box
3. Integrated source knowledge to solve issues faster
4. Custom development & coaching
5. Highly responsive support team
6. Easy onboarding & renewals
7. Largest set of Microsoft SCOM Management Packs

### About Microsoft System Center Operations Manager (SCOM)

Microsoft SCOM is a powerful IT management solution designed to help organizations monitor, troubleshoot, and ensure the health of their IT infrastructure. SCOM provides comprehensive infrastructure monitoring, offering insights into the performance, availability, and security of applications and workloads across on-premises, cloud, and hybrid environments. With its robust set of features, SCOM enables IT professionals to proactively identify and address potential issues before they impact the business, improving overall operational efficiency and reducing downtime. By leveraging SCOM, businesses can achieve greater control over their IT environment, ensuring a seamless user experience and enhancing the reliability of their services.

Take advantage of all the benefits of advanced monitoring using NiCE Management Packs for Microsoft System Center Operations Manager. Contact us at [solutions@nice.de](mailto:solutions@nice.de) (EMEA, APAC), or [solutions@nice.us.com](mailto:solutions@nice.us.com) (US, LATAM) for a quick demo, and a free 60 days trial.

#### NiCE IT Management Solutions GmbH

Liebigstrasse 9  
71229 Leonberg  
Germany

[www.nice.de](http://www.nice.de)  
[solutions@nice.de](mailto:solutions@nice.de)

#### NiCE IT Management Solutions Corporation

3478 Buskirk Avenue, Suite 1000  
Pleasant Hill, CA 94523  
USA

[www.nice.us.com](http://www.nice.us.com)  
[solutions@nice.us.com](mailto:solutions@nice.us.com)

