Server & Application Monitor

Version 6.9
© 2019 SolarWinds Worldwide, LLC. All rights reserved.

This document may not be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the prior written consent of SolarWinds. All right, title, and interest in and to the software, services, and documentation are and shall remain the exclusive property of SolarWinds, its affiliates, and/or its respective licensors.

SOLARWINDS DISCLAIMS ALL WARRANTIES, CONDITIONS, OR OTHER TERMS, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, ON THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION NONINFRINGEMENT, ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION CONTAINED HEREIN. IN NO EVENT SHALL SOLARWINDS, ITS SUPPLIERS, NOR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, WHETHER ARISING IN TORT, CONTRACT OR ANY OTHER LEGAL THEORY, EVEN IF SOLARWINDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The SolarWinds, SolarWinds & Design, Orion, and THWACK trademarks are the exclusive property of SolarWinds Worldwide, LLC or its affiliates, are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other SolarWinds trademarks, service marks, and logos may be common law marks or are registered or pending registration. All other trademarks mentioned herein are used for identification purposes only and are trademarks of (and may be registered trademarks) of their respective companies.
# Table of Contents

- Monitor your environment with SAM ................................................................. 12
- Monitor your environment with SAM ............................................................. 12
- Orion Platform features available in SAM ......................................................... 13
- SAM licensing model ...................................................................................... 15
  - Verify the component monitors consumed per template ............................. 16
  - Manage and activate your SAM license ....................................................... 16
- Prepare and install SAM ................................................................................ 17
  - Install an evaluation version of SAM ......................................................... 18
- Log into the Orion Web Console to use SAM ................................................. 19
- Uninstall SAM ............................................................................................... 20
- Before you begin .......................................................................................... 20
- Uninstall SAM ............................................................................................... 20
- Uninstall the whole Orion Platform deployment .......................................... 21

## Discover and add network devices ............................................................... 22

- Manage devices in the Orion Web Console ................................................... 22
  - Promote a node from ICMP to SNMP monitoring in SAM .................... 23
  - Promote a Node to WMI Monitoring ......................................................... 24

## Discover applications in your environment ............................................... 25

- Application Discovery .................................................................................. 25
- Add application monitors to nodes ............................................................... 26
- Customize Application Details views ............................................................ 27
  - Create a custom Application Details view ................................................. 27
  - Select a view for a template ................................................................... 28
  - Select a view for an application .............................................................. 28
- Customize Application Summary views ....................................................... 28
- Viewing node and application data in tooltips .......................................... 30
  - Node tooltips .......................................................................................... 30
  - Application tooltips .............................................................................. 30
**Add SAM data to the Node Details view** ........................................................................................................... 31

**Manage Asset Inventory in SAM** ......................................................................................................................... 33
- Asset Inventory polling requirements .................................................................................................................. 36
- Download agent software ......................................................................................................................................... 36
- Enable Asset Inventory polling for individual nodes .............................................................................................. 37
- Adjust Asset Inventory polling frequency ............................................................................................................. 38
- Configure how SAM handles certificate errors during Asset Inventory polling ....................................................... 38
- Troubleshoot asset inventory issues .................................................................................................................... 39

**Manage processes, services, tasks, and events in real time with SAM** ................................................................. 40
- Real-Time Process Explorer .................................................................................................................................... 40
  - Monitor unmonitored processes in SAM .............................................................................................................. 41
  - Use the Real-Time Process Explorer in SAM ...................................................................................................... 41
  - Real-Time Process Explorer alerts .................................................................................................................... 43
  - Real-Time Process information in top 10 alerts .................................................................................................. 43
  - Real-Time Event Log Viewer .................................................................................................................................. 44
- Service Control Manager ........................................................................................................................................... 48
- Windows Scheduled Task Monitor .......................................................................................................................... 50
  - Windows Scheduled Task Monitor status codes ................................................................................................ 52
  - Windows Scheduled Task Monitor status ........................................................................................................... 54

**Monitor your network in SAM** ............................................................................................................................... 57
- Monitor with Orion agents in SAM .......................................................................................................................... 58
  - Overriding agent and agentless monitoring ........................................................................................................ 58
  - Communication type ................................................................................................................................................... 59
  - Agent resource consumption .................................................................................................................................. 59
  - Templates and component monitors ..................................................................................................................... 59
  - Reports ..................................................................................................................................................................... 61
  - Configure SNMP for agents deployed by SAM ..................................................................................................... 61
    - Configure SNMP ..................................................................................................................................................... 62
    - Configure SNMP v3 ............................................................................................................................................... 62
    - Enable the SNMP daemon on AIX computers .................................................................................................. 64
Monitor your virtual infrastructure in SAM .............................................................. 65
Monitor your environment using AppStack ............................................................ 66
Troubleshoot with AppStack .................................................................................. 67
Introduction to the AppStack Environment view .................................................... 68
The AppStack Environment view Analogy ............................................................... 69
AppStack categories and data ................................................................................. 73
Supported categories and data ................................................................................ 73
AppStack data overview ......................................................................................... 77
Object status and colors ......................................................................................... 77
Quick links ................................................................................................................ 78
Overview bar ........................................................................................................... 80
Customize the AppStack view in SAM ................................................................. 80
Use filters for AppStack .......................................................................................... 81
Filter options ........................................................................................................... 81
Creating filters ......................................................................................................... 82
Create an AppStack layout ...................................................................................... 83
Use alerts to monitor your environment with SAM ............................................. 84
Use SAM alert variables ......................................................................................... 85
Component monitor properties for alerts ............................................................. 85
Component Monitor variables for alerts ............................................................... 88
Application Monitor properties for alerts ........................................................... 92
Application Monitor variables for alerts ............................................................... 92
Monitor application dependencies in SAM ........................................................ 95
Application Dependencies polling overview .................................................... 97
How SAM uses Orion agents to monitor application dependencies .................. 98
Agent plug-in overview .......................................................................................... 98
Configure the Application Dependencies feature ............................................. 99
Before you begin ..................................................................................................... 100
Enable Application Dependency polling ............................................................ 101
Deploy Orion agents to nodes that host applications you want to monitor .......... 102
Assign application monitors to nodes ................................................................. 102
Customize polling settings .................................................................................. 102
Set up Application Dependencies alerts .............................................................. 103
Use the Application Dependencies feature with Windows Failover Clusters (WFCs) ........ 103
Access application dependency data in SAM ....................................................... 105
Application Connections widget ........................................................................... 105
Connection Details page ....................................................................................... 108
Manage polling for application dependencies ...................................................... 110
Manage Application Dependency polling ............................................................. 111
Manage Connection Quality polling ................................................................... 112
Troubleshoot application dependency issues in SAM ............................................ 114
Troubleshoot agent-related issues for application dependencies in SAM .............. 114
Troubleshoot data-related issues for application dependencies in SAM ................. 117
Troubleshoot miscellaneous issues for application dependencies ........................... 118
Locate Application Dependency log files ............................................................. 120
Monitor hardware health in SAM ......................................................................... 120
Configure hardware health monitoring ................................................................. 121
Hardware health monitoring requirements in SAM ................................................. 121
OS and protocol requirements ............................................................................. 122
Download third-party software .......................................................................... 122
Enable hardware health monitoring for individual nodes in SAM ......................... 122
Configure how SAM handles certificate errors during Hardware Health polling ........ 123

**Monitor with AppInsight applications** ................................................................ 124
The template and application relationship ............................................................. 125
AppInsight for Active Directory .......................................................................... 126
AppInsight for Active Directory licensing ............................................................. 127
AppInsight for Active Directory requirements and permissions ............................ 128
Configure AppInsight for Active Directory ......................................................... 129
Add AppInsight for Active Directory to domain controller nodes ....................... 129
Edit the AppInsight for Active Directory template .............................................. 131
Use the "Enable total counters" feature ................................................................. 132
AppInsight for Active Directory alerts ................................................................. 132
Troubleshoot AppInsight for Active Directory ....................................................... 133
AppInsight for Exchange ......................................................................................... 135
Navigating AppInsight for Exchange .................................................................... 135
Recommendations for AppInsight for Exchange ..................................................... 137
AppInsight for Exchange licensing ....................................................................... 137
AppInsight for Exchange requirements and permissions ....................................... 138
Configure AppInsight for Exchange on nodes ....................................................... 139
Edit the AppInsight for Exchange template ......................................................... 143
Advanced manual configuration of AppInsight for Exchange .............................. 144
Verify Microsoft Exchange credentials .................................................................. 144
Manually configure an Exchange server .................................................................. 145
Set PowerShell permissions for Exchange ............................................................. 151
Find Exchange URL settings .................................................................................. 152
Attachment extensions acknowledged by AppInsight for Exchange ..................... 154
AppInsight for IIS .................................................................................................... 156
Navigate to the AppInsight for IIS details view ..................................................... 156
AppInsight for IIS requirements and permissions .................................................. 157
  AppInsight for IIS technologies and ports ............................................................. 158
AppInsight for IIS licensing .................................................................................... 159
Configure AppInsight for IIS on nodes .................................................................. 160
  Configure AppInsight for IIS on nodes with the Add Node wizard ....................... 160
  Configure AppInsight for IIS via the Node Details page ...................................... 161
Integrate SolarWinds APM with IIS nodes ............................................................ 163
  Integrate APM with an IIS node in SAM ............................................................... 165
  Use the APM subview ........................................................................................... 168
Edit the AppInsight for IIS template ...................................................................... 170
Advanced manual configuration of AppInsight for IIS ......................................... 171
  Install PowerShell on target servers .................................................................. 171
Set the execution policy on target servers ................................................................. 172
Create a self-signed certificate .................................................................................... 172
Create a firewall rule for AppInsight for IIS ............................................................... 173
Update WsMan limits .................................................................................................... 173
Create a WinRM listener for AppInsight for IIS .......................................................... 174
Find IIS URL settings ................................................................................................... 175
AppInsight for SQL ....................................................................................................... 177
AppInsight For SQL licensing ....................................................................................... 181
Add AppInsight for SQL to a node ............................................................................. 181
AppInsight for SQL requirements and permissions .................................................... 184
AppInsight for SQL permissions .................................................................................. 184
AppInsight for SQL licensing ....................................................................................... 186
Monitor a SQL named instance with AppInsight for SQL ............................................ 186
Monitor clusters with AppInsight for SQL .................................................................... 188
Edit the AppInsight for SQL template .......................................................................... 190
Customize AppInsight for SQL monitoring ................................................................. 190
Hide databases from AppInsight for SQL in the All Databases widget ....................... 192
AppInsight for SQL alerts ............................................................................................ 192
View Alerts .................................................................................................................... 193

Manage SAM templates and component monitors ...................................................... 194
Learn more .................................................................................................................... 198
Manage SAM templates ............................................................................................... 198
Create a template ........................................................................................................... 199
Create a custom template (Example) ............................................................................ 200
Create a SolarWinds Windows service application template ........................................ 201
Assign a template to a node .......................................................................................... 201
Edit a template .............................................................................................................. 202
Add Component Monitors ............................................................................................ 202
Copy a template ............................................................................................................. 203
Export and import templates ......................................................................................... 203
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete templates</td>
<td>206</td>
</tr>
<tr>
<td>Tag a template</td>
<td>206</td>
</tr>
<tr>
<td>Change between 32-bit and 64-bit polling</td>
<td>206</td>
</tr>
<tr>
<td>Best practices for SAM templates</td>
<td>207</td>
</tr>
<tr>
<td>Performance enhancements</td>
<td>207</td>
</tr>
<tr>
<td>Script, monitor, and template testing</td>
<td>208</td>
</tr>
<tr>
<td>Script best practices</td>
<td>208</td>
</tr>
<tr>
<td>Import and export SAM templates</td>
<td>210</td>
</tr>
<tr>
<td>Export a template</td>
<td>211</td>
</tr>
<tr>
<td>Import a template</td>
<td>212</td>
</tr>
<tr>
<td>Understand the Credentials Library</td>
<td>213</td>
</tr>
<tr>
<td>Manage credentials in the Credentials Library</td>
<td>213</td>
</tr>
<tr>
<td>Store SSH keys in the Certificate Credentials Library</td>
<td>214</td>
</tr>
<tr>
<td>Share application monitor templates on THWACK</td>
<td>215</td>
</tr>
<tr>
<td>Export a template</td>
<td>216</td>
</tr>
<tr>
<td>Import a template</td>
<td>217</td>
</tr>
<tr>
<td>Manage templates and groups</td>
<td>218</td>
</tr>
<tr>
<td>Assign an application template to a group</td>
<td>218</td>
</tr>
<tr>
<td>Create a template for an application</td>
<td>220</td>
</tr>
<tr>
<td>Unassign application monitors from a group</td>
<td>222</td>
</tr>
<tr>
<td>Work with component monitors in SAM templates</td>
<td>223</td>
</tr>
<tr>
<td>Report status through exit codes</td>
<td>227</td>
</tr>
<tr>
<td>Assign application monitors to nodes</td>
<td>227</td>
</tr>
<tr>
<td>Scan nodes for applications to apply templates automatically in SAM</td>
<td>228</td>
</tr>
<tr>
<td>Unmanage and manage assigned application monitors</td>
<td>229</td>
</tr>
<tr>
<td>Modify templates, application monitors, and component monitors in SAM</td>
<td>230</td>
</tr>
<tr>
<td>Create component monitors in the Component Monitor Wizard</td>
<td>231</td>
</tr>
<tr>
<td>Linux/Unix system configurations for component monitors</td>
<td>232</td>
</tr>
<tr>
<td>Scripts with text output</td>
<td>244</td>
</tr>
<tr>
<td>Use Windows Script monitors</td>
<td>245</td>
</tr>
</tbody>
</table>
Field descriptions ........................................................................................................ 246
Create custom Windows scripts in SAM ................................................................. 247
Create custom component script monitors ............................................................. 249
Create a Windows Script monitor .......................................................................... 250
Linux/Unix Script monitor ..................................................................................... 254
Nagios Script monitor ............................................................................................ 260
Create a Windows PowerShell monitor .................................................................... 264
Example tasks for application monitors ................................................................. 269
Create your first application and alert ...................................................................... 269
Scan for applications to monitor ........................................................................... 271
Monitor a specific URL ............................................................................................ 271
Monitor VMware performance counters using an application monitor .................... 272
Monitor and restart stopped Windows services ....................................................... 273
Monitor large directories using the Windows Script Monitor ................................... 274
Monitor IIS application pools .................................................................................. 275
Use the JMX component monitor wizard ............................................................... 276
Configure the SOAP monitor .................................................................................. 277
Understand application monitor thresholds ............................................................ 279
Inheriting thresholds in SAM .................................................................................. 280
Multi-Value scripts and thresholds ......................................................................... 281
Apply baseline thresholds at the template level ....................................................... 282
Apply baseline thresholds at the application level .................................................. 284
Adjust threshold settings and apply baseline data .................................................. 285
View latest baseline details ..................................................................................... 289
Use PowerShell in SAM .......................................................................................... 290
PowerShell security considerations ........................................................................ 291
PowerShell requirements ....................................................................................... 291
Enable remote access for PowerShell with WinRM ............................................... 292
Working with Exchange Management Tools ......................................................... 293
Learn more about PowerShell .................................................................................. 293
Monitor your environment with SAM

SAM gives you the tools to monitor your servers and applications through a single web console. It provides custom collections of templates, application monitors, and alerts to intelligently monitor application status and issues. Monitor over 200 application types including application servers, authentication servers, database servers, and more.

SAM monitors applications in public, private, and hybrid environments using:

- **AppStack**: Provides an interactive visual mapping within Orion with an in-depth perspective through the entire environment to help identify the root cause of performance and availability issues.
- **AppInsight**: Monitors Microsoft Active Directory, Exchange, IIS, and SQL servers in an environment, displaying metrics, status, and issues to manage and maintain applications and servers.
- **Application monitor templates**: Combine process monitors, port availability, and performance counters to assess the status of every aspect of an application, including overall health.
- **Alerts**: Identify issues using customizable triggers, static or baseline thresholds, and actions.
- **Cloud monitoring**: Provides visibility into cloud environments through the Orion Web Console.
- **Application dependencies**: Detect how applications and nodes interact to ensure important data for specific applications is monitored, and identify unmonitored applications and processes that require attention.
- **PerfStack**: Troubleshoot environment issues in Performance Analysis (PerfStack) dashboards based on data collected by your Orion Platform products.

For an overview of how SAM works, see the [SAM Getting Started Guide](#).

To expand SAM's monitoring capabilities, you can integrate it with other Orion Platform products:

- Display advanced performance metrics for IIS-based, .NET applications with [Application Performance Monitor (APM)](#).
- Add response-time analysis to see the root cause of application response issues with the [Application Performance Optimization Pack](#). By bundling [Database Performance Analyzer (DPA)](#) with SAM, you can use historic analysis and dynamic baselines to spot SQL tuning problems.
- Find and fix both web and Software-as-a-Service application performance issues before users are affected with [Web Performance Monitor (WPM)](#).

Monitor your environment with SAM

SAM gives you the tools to monitor your servers and applications through a single web console. It provides custom collections of templates, application monitors, and alerts to intelligently monitor application status and issues. Monitor over 200 application types including application servers, authentication servers, database servers, and more.
SAM monitors applications in public, private, and hybrid environments using:

- **AppStack**: Provides an interactive visual mapping within Orion with an in-depth perspective through the entire environment to help identify the root cause of performance and availability issues.
- **AppInsight**: Monitors Microsoft Active Directory, Exchange, IIS, and SQL servers in an environment, displaying metrics, status, and issues to manage and maintain applications and servers.
- **Application monitor templates**: Combine process monitors, port availability, and performance counters to assess the status of every aspect of an application, including overall health.
- **Alerts**: Identify issues using customizable triggers, static or baseline thresholds, and actions.
- **Cloud monitoring**: Provides visibility into cloud environments through the Orion Web Console.
- **Application dependencies**: Detect how applications and nodes interact to ensure important data for specific applications is monitored, and identify unmonitored applications and processes that require attention.
- **PerfStack**: Troubleshoot environment issues in Performance Analysis (PerfStack) dashboards based on data collected by your Orion Platform products.

For an overview of how SAM works, see the [SAM Getting Started Guide](#).

To expand SAM's monitoring capabilities, you can integrate it with other Orion Platform products:

- Display advanced performance metrics for IIS-based, .NET applications with [Application Performance Monitor (APM)](#).
- Add response-time analysis to see the root cause of application response issues with the [Application Performance Optimization Pack](#). By bundling [Database Performance Analyzer (DPA)](#) with SAM, you can use historic analysis and dynamic baselines to spot SQL tuning problems.
- Find and fix both web and Software-as-a-Service application performance issues before users are affected with [Web Performance Monitor (WPM)](#).

**Orion Platform features available in SAM**

The [Orion Platform](#) is at the core of the SolarWinds IT Management Portfolio. It provides a stable and scalable architecture that includes data collection, processing, storage, and presentation. It provides common features such as user accounts, views, dashboards, reporting, alerting, and more that you can use across all Orion Platform products and access from the Orion Web Console.

SAM includes many features that are common to all Orion Platform products, as summarized here and documented in the [Orion Platform Administration Guide](#).

**Orion Platform basics**

Log into your Orion Platform product in a web browser and [meet the Orion Web Console](#).

Review [events](#), [Syslogs](#), or [SNMP traps](#) to know what's going on.

[Get alerts](#) about issues in your environment and generate [reports](#) about its status.

Review [Performance Analysis dashboards](#), also known as PerfStack™.
Manage Orion Web Console user accounts to set user rights, reset passwords, limit access to network segments, and enable authentication with Active Directory.

View monitored objects on maps in the Orion Web Console.

Add devices for monitoring and manage monitored devices

Specify which devices to monitor and the information you need, then select the way you get this information. See Discover and add devices.

Add single nodes, use Active Directory domain controllers to add nodes, or discover devices on your network automatically. Available polling methods include ICMP, WMI, SNMP, or agents deployed on Windows, Linux, and UIX devices.

Manage monitored devices by editing properties, setting polling methods, toggling monitoring on and off, or muting alerts.

Customize your Orion Web Console

Customize dashboards, colors, logo, views, widgets, and charts in the Orion Web Console. Specify objects that other users can see on views, or what type of data appears for different device types.

Create custom properties for monitored network objects and display custom details for devices.

Create groups and dependencies to organize how monitored data is presented in the Orion Web Console. Set up dependencies to better represent the relationships between network objects and account for constraints on the network.

Set thresholds for monitored metrics. Customize general thresholds or use baselines.

Monitor additional metrics and devices

Monitor hardware health to gain insight into hardware issues on the network.

Monitor virtual environments in the Orion Web Console.

Use the Quality of Experience dashboard to monitor traffic on your network.

Expand Orion Platform functionality or scale your deployment

Use SolarWinds High Availability (HA) to provide failover protection for your Orion server and Additional Polling Engines (APEs).

Do you need to scale your deployment? See Scalability Engine Guidelines.

Review tips for optimizing your deployment.

Balance the load on polling engines by specifying nodes to be polled by individual polling engines. See also Manage Additional Polling Engines.
Troubleshoot your environment

See Troubleshoot your SolarWinds Orion database and use the Active Diagnostics tool.

SAM licensing model

Your SAM license determines how many assigned component monitors you are allowed. A component monitor provides a statistic that you want to monitor in SAM. You can use as many application templates and monitors as needed, as long as the number of assigned component monitors doesn’t exceed the license count.

The value returned by a component monitor is a measurement of application health and availability that needs to be monitored. It can be a process, service, performance counter, port, URL, User Experience Monitor, Nagios script, and so on.

- An application typically consumes 15 — 25 component monitors.
- AppInsight for Active Directory consumes 50 component monitors per monitored domain controller.
- AppInsight for Exchange consumes 50 component monitors per monitored mailbox role server.
- AppInsight for IIS consumes 30 component monitors per monitored IIS server.
- AppInsight for SQL consumes 50 component monitors per database instance.
- The Windows Scheduled Task Monitor consumes 5 component monitors per Windows server.

Asset Inventory data collection does not count against your SAM license.

If you exceed the license limit, monitors beyond the limit are not activated and statuses change to Not Licensed. Either disable assigned component monitors to reduce your total assigned monitor count, or contact SolarWinds Sales to upgrade your license.

<table>
<thead>
<tr>
<th>SAM LICENSE TIERS</th>
<th>NUMBER OF COMPONENTS TO MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL150</td>
<td>150</td>
</tr>
<tr>
<td>AL300</td>
<td>300</td>
</tr>
<tr>
<td>AL700</td>
<td>700</td>
</tr>
<tr>
<td>AL1100</td>
<td>1100</td>
</tr>
<tr>
<td>AL1500</td>
<td>1500</td>
</tr>
<tr>
<td>AL2500</td>
<td>2500</td>
</tr>
<tr>
<td>ALX</td>
<td>Unlimited number of items to monitor standard polling throughput.</td>
</tr>
</tbody>
</table>

To verify the number of consumed and available component monitors in your license, access the SAM License Summary.

1. In the Orion Web Console, click Settings > All Settings.
2. Under Product Specific Settings, click SAM Settings.
3. Click SAM License Summary.

Verify the component monitors consumed per template

SAM licenses do not have to mirror the license count of other installed SolarWinds products. For example, you can install SAM with a 150 component license on an NPM server with an unlimited node license.

**Example:** The Apache template has 8 available licensed component monitors. If you have an AL150 license and assign all 8 component monitors in this template to a node, the balance of remaining licenses is 142 (150-8 = 142). If you assign only 5 of the 8 component monitors to a node, the balance of remaining licenses is 145 (150-5=145).

To view the number of licensed component monitors per template in SAM, click Settings > All Settings > SAM Settings > Manage Templates. Review the Licensed Components column amount per template.

Manage and activate your SAM license

Note these details when the **License Manager** prompts you to activate your SAM license:

- When using your Customer ID (also called a "SWID") and password, use your individual profile information. If you do not know your Customer ID or password, enter a Support ticket.
- To display licenses and activation keys that are not yet activated, log into the **Customer Portal**.
- Your SAM license interacts additively with other SolarWinds licenses. For example, if you have NPM SL500 (500 nodes and 500 volumes) installed with SAM A1L50, you can monitor:
  - 650 nodes (500 NPM nodes + 150 SAM nodes)
  - 650 volumes (the number of volumes matches the node count)
  - 500 interfaces monitored with SNMP
To manage SAM and other licenses:

1. Click Settings > All Settings in the Orion Web Console.
2. Click License Details in the Details section.
3. Click License Manager.

You can sort licenses by product name, version, license type, or expiration date. You can also add a license, upgrade a license to a new tier, and activate licenses on new servers. The License Manager on the Orion server controls all licenses for your SAM environment, including scalability engines and High Availability.

If you license your product before you install it, click Add/Upgrade License, enter the details, and complete the activation to see the license in the License Manager.

Prepare and install SAM

This overview is for new installations of SolarWinds SAM. To upgrade an existing installation of SAM, see the SolarWinds Upgrade Guide.

To prepare and install SAM, refer to the SAM Getting Started Guide and use the SolarWinds Orion Installer, an all-in-one application that allows you to install and upgrade one or more Orion Platform products simultaneously. The installer:

- Provides an easy-to-follow installation path. The installer checks for product updates, and then presents installation steps to complete complex upgrades and installations with ease. No need to figure out your upgrade path before you begin - the installer does it for you.
- Runs preflight checks to ensure your environment specifications match the system requirements for selected product installations. If you need to make any changes, the installer provides guidance to resolve the issues.
- Guides you through product installation and upgrades by providing upgrade paths needed for your current environment.
- Automatically runs the Configuration Wizard as needed after product installations to complete database and configuration tasks.
- Advises on additional installations for SolarWinds High Availability servers, Additional Polling Engines, and Additional Web Servers.

Check if your environment meets the recommended requirements

Review the SAM release notes and system requirements.

If you plan to use SAM to monitor and manage a large environment, see Scalability Engine Guidelines for SolarWinds Orion Products.
Download and use the Orion Installer

Use the Orion Installer to install SAM.

- If evaluating SAM, the Orion Installer is included with the free [SAM trial download](https://www.solarwinds.com/products/sam-trial).
- The Orion Installer is included in the latest install package for SAM, available in the Customer Portal.
  2. Navigate to Downloads > Download Product.

After the download is complete, run the installer to begin installation. The Orion Platform includes several wizards to walk you through important steps, including:

- The [Orion Installer](https://www.solarwinds.com/products/sam-trial) guides the installation of one or more Orion Platform products.
- The [Configuration wizard](https://www.solarwinds.com/products/sam-trial) opens during installation if database configuration is required.
- The [Discovery Wizard](https://www.solarwinds.com/products/sam-trial) automatically detects and adds devices to lay the groundwork for your SAM environment. Later, you can [schedule Discovery jobs to add new nodes automatically](https://www.solarwinds.com/products/sam-trial).


After SAM is installed, your next step is to [log into the Orion Web Console](https://www.solarwinds.com/products/sam-trial).

Install an evaluation version of SAM

You can install an evaluation version of SAM that remains functional for 30 days. The evaluation period begins when you install the product if you do not have a commercial license activated. You can verify the days remaining for the evaluation in the [License Manager](https://www.solarwinds.com/products/sam-trial).

Installation on a desktop operating system is possible, but not supported. This product is intended for use in a production environment that meets SAM [system requirements](https://www.solarwinds.com/products/sam-trial).

**If you are new to SolarWinds**, follow these steps to install an evaluation version of SAM:

2. Enter information in the fields provided and click Proceed to Free Download.
3. Click Download Now.
5. Follow the onscreen instructions to complete the installation.

**If you already have a Customer Portal account**, follow these steps to install an evaluation version:

2. Click Downloads > Download Product.
3. Select Server & Application Monitor (SAM) in the Products field, then select a license.
4. Click a Download link in the Latest Release Downloads section.
5. Double-click the downloaded file to launch the Orion Installer.
6. Follow the onscreen instructions to complete the installation.

Use the SAM Getting Started Guide to learn about the SAM environment and how you can tailor it to your internal processes so you can more effectively respond to performance issues.

To purchase a full SAM license, contact SolarWinds Sales. Then use the License Manager to update the license with the purchased key.

Log into the Orion Web Console to use SAM

As with all Orion Platform products, you can access SAM by logging into the Orion Web Console to do everything from adding devices to discovering applications for monitoring, and setting up alerts.

In a web browser, navigate to http://hostnameOrIPAddress:port where:

- hostnameOrIPAddress is the hostname or IP address of the Orion server where SAM is installed.
- port is the Orion Web Console port defined for the website. This is configurable during installation. The default port is 8787.

On the Orion Web Console login page, provide a User name and Password, and then click Login.

If you're logging into a new installation of SAM, you'll be prompted to add a password to the main Admin account.
Uninstall SAM

Uninstalling products may require uninstalling files and editing registries. For technical assistance, contact Support.

You may need to uninstall SolarWinds NPM to reinstall for resolving issues or to move to a new server during a migration.

Before you begin

Prior to uninstalling, SolarWinds recommends the following preparation:

- **Back up the existing database**
  To preserve your data, back up your database(s). If you need help with backups, check your vendor’s site for documentation and instructions.
  - If you have your database on a VM, create a snapshot or copy of your VM.
  - If you need software to perform backups and maintenance for Microsoft SQL databases, you can install SQL Server Management Studio Express for your specific version of Microsoft SQL on your database server.

- **Back up product folders**
  Create copies and backups of your product folders. You may have customizations in these folders specific to your installations. For example, save custom code or reports.

Uninstall SAM

1. **Remove product licenses**
   1. In the Orion Web Console, select Settings > All Settings and click License Manager.
   2. In the License Manager, select the SAM license to remove.
   3. Click Deactivate to remove the SAM license activation and server assignment. With the license deactivated, it is parked, or available but unused.

   ![Important Note]
   Depending on how many products you need to uninstall, you may want to deactivate licenses for additional SolarWinds products such as Additional Polling Engines (APEs).

2. **Uninstall SolarWinds products**
   1. Open Programs and Features in the Windows Control Panel.
   2. Select the product(s) to remove one at a time and click Uninstall.

   ![Important Note]
   If you have multiple Orion Platform products installed but are not uninstalling all of them, skip to step 5, Restart and reinstall. Deleting SolarWinds folders and registries affects the operating of all Orion Platform products.
### 3. Delete or rename SolarWinds folders

Delete files from the following locations to fully clear all files. If you installed the Orion server on a different domain, look on that location instead of C:\.

- C:\Program Files (x86)\SolarWinds
- C:\Program Files (x86)\Common Files\SolarWinds
- C:\inetpub\SolarWinds
- C:\ProgramData\SolarWinds
- C:\ProgramData\SolarWindsAgentInstall

### 4. Remove specific Registry keys

**SolarWinds strongly recommends that you back up your registry before editing your system registry.** You should only edit the registry if you are experienced and confident in doing so. Using a registry editor incorrectly can cause serious issues with your OS, which could require you to reinstall your OS to correct them. SolarWinds cannot guarantee resolutions to any damage resulting from making registry edits. For assistance, contact Support.

1. Open the command line interface on the server.
2. Type `regedit`, and click OK.
3. Expand `HKEY_LOCAL_MACHINE` > `Software`.
4. Delete both the SolarWinds and the SolarWinds.net folders.
5. If uninstalling from a 64-bit computer, expand `HKEY_LOCAL_MACHINE` > `Software` > `Wow6432Node`, and delete both the SolarWinds and the SolarWinds.net folders.

### 5. Restart and reinstall

Restart the server. Use the [SolarWinds Orion Installer](https://www.solarwinds.com) to reinstall SAM and other products.

---

### Uninstall the whole Orion Platform deployment

Starting with Orion Platform 2019.2, you can uninstall the whole Orion Platform deployment using the [SolarWinds Orion Installer](https://www.solarwinds.com).

If you have scalability engines deployed, run the following steps on your scalability engines (that is, Additional Polling Engines and Additional Web servers) and then repeat the steps on the main polling engine — the Orion server.

1. In the command line, run the following command:

   `'path to exe'\SolarWinds.Orion.Installer.exe -uninstall`

2. Follow the onscreen instructions to complete the uninstallation.

All Orion Platform products are removed, including all relevant registry entries and folders.
Discover and add network devices

After you install SAM, you can identify the devices you want to monitor and add them to the SolarWinds Orion database. There are several ways to do this, including

- Use the Discovery Wizard to detect a large number of devices across your enterprise, and then add discovered devices with the Network Sonar Results Wizard.
- Add nodes by querying your Active Directory domain controller
- Add individual nodes on the Manage Nodes page.

To learn more, review the following Orion Platform Administrator Guide topics:

- Discover your network with the Discovery wizard
- Add nodes using Active Directory
- Automatically add discovered nodes
- Add discovered devices
- Add a single node for monitoring
- Choose polling methods
- Import nodes from a list of IP addresses
- Manage scheduled discovery results
- Manage devices in the Orion Web Console
- Group objects and mirror network dependencies

This section also includes the following topics:

- Promote a node from ICMP to SNMP monitoring in SAM
- Promote a Node to WMI Monitoring

When finished adding and configuring devices, you can start Discover applications in your environment.

Manage devices in the Orion Web Console

This section includes details about promoting nodes from ICMP to SNMP monitoring, as well as promoting nodes to WMI monitoring. If you have node management rights, you can add and remove devices, and quickly view and edit device properties from the Manage Nodes view in the Orion Web Console.

To access the Manage Nodes view:

- Click Settings > Manage Nodes.
- Click Manage Nodes in the All Nodes widget.

The Orion Summary Home view includes the All Nodes widget, by default, but you can include that widget on other views also.

See the Orion Platform Administrator Guide to learn about managing devices in the Orion Web Console, including how to:
Promote a node from ICMP to SNMP monitoring in SAM

Orion Platform products only use ICMP to poll devices for status, average response time, and packet loss. If a node which you added to the SolarWinds Orion database as an ICMP only node also supports SNMP, and you want to start collecting additional statistics, change the polling method to SNMP.

1. Log into the Orion Web Console as an administrator.
2. Click Settings > Manage Nodes.
3. Select Nodes from the Show drop-down list, and locate the node which you want to edit.
4. Select the node, and click Edit Properties.
5. In the Polling Method section, select Most Devices: SNMP and ICMP.
6. Select the version of SNMP to use. The default is SNMPv2c.
7. If you have installed multiple polling engines, select the Polling Engine you want to use to collect statistics from the added node. This option is not displayed if you are only using one polling engine.
8. If the SNMP port on the added node is not the Orion default of 161, enter the actual port number.
9. If the added node supports 64-bit counters and you want to use them, select Allow 64-bit Counters.

   ① If you notice erratic or incorrect data when using 64-bit counters, clear the Allow 64 Bit Counters box for the device, and contact the hardware manufacturer.

10. For SNMPv1 or SNMPv2c, enter the Community String and, optionally, the Read/Write Community String. Click Test to validate the strings.

    ① The Community String is a password to authenticate data sent between the SNMP management station and the device. The default is usually "public". Use the strings configured on the device.

11. For SNMPv3, provide the credentials and click Test to validate the credentials. See the vendor documentation for your network device for further information.
12. Click Submit.
Promote a Node to WMI Monitoring

💡 To learn about WMI monitoring, check out the Portocalypse - WMI Demystified product blog in the online SolarWinds IT community, THWACK.

ℹ️ You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, navigate to Settings > Manage Nodes.
2. Check the node you want to promote, click Edit Properties, and then select Windows Servers: WMI and ICMP.
3. Enter the appropriate WMI credentials.
4. Click Test to confirm your settings. When done, click Submit, then click OK.
Discover applications in your environment

SolarWinds SAM can scan nodes and automatically assign the Application Monitors suitable for each scanned node. You control the nodes to be scanned, the application templates used in the scan, and the scanning parameters that determine a match.

- **Application Discovery**: Discover and add applications through discovery. Set up application monitoring with application monitors and templates.
- **Add application monitors to nodes**: Learn how to add application monitors to monitored nodes.
- **Customize Application Details views**: Learn how to customize the application and template views.
- **Customize Application Summary views**: Modify widgets displayed in the Application Summary page.
- **Add SAM data to the Node Details view**: Modify Node Details to display SAM widgets.
- **Viewing node and application data in tooltips**: Hover over nodes and application data to display more data.

To expand monitoring by detecting how applications and nodes interact with each other, check out SAM's **Application Dependencies feature**.

Application Discovery

In SolarWinds SAM, you can use the Discover Applications wizard to scan nodes and automatically assign the Application Monitors it deems suitable for each scanned node. You control the nodes to be scanned, the application templates used in the scan, and the scanning parameters that determine a match.

- **Click Settings > All Settings > Discovery Central > Discover Applications.** To use the wizard, click Discover Applications.
- **Click [+]:** expand the node groups, and then select the nodes you want to scan, then click Next.
- **Select the applications to locate in the scan.** You can use the Show only drop-down list to filter the application template list. As you select templates across the filter groups, they display in the selected applications section.

To adjust the template assignment criteria, expand Advanced Scan Settings and move the slider to the desired setting:

- **Minimal Match** - At least one component must match to assign the template.
- **Partial Match** - At least some of the components must match to assign the template.
- **Strong Match** - Most of the components must match to assign the template.
4. When you are finished selecting applications, click Next.

5. Some application templates require credentials either to access restricted resources, or to run within the context of a specific user. To scan for these templates, add the necessary credentials to the list. If a template you are scanning for requires credentials, the credentials in this list are tried in the order in which they appear.

6. If you domains share user names with different passwords, SolarWinds recommends that you run separate application discoveries for each domain. After entering credentials, click Next.

7. Review the summary for the scan. If the automatic discovery matches templates that are already assigned to the node, by default the template is not assigned a second time. If you want to assign duplicate templates, select Yes, Assign Anyway from the Do you want to assign duplicates list.

8. Click Start Scan to begin the scan, running in the background. A banner message near the top of the screen indicates progress.

A message appears when scanning is complete. Click More Details to see the results of the scan. Click Application Summary to display the summary page.

### Add application monitors to nodes

After adding individual nodes, you are prompted to add application monitors for application(s) on the new nodes. Later, you can modify the assigned application monitors for a node at any time, as necessary.

1. Add a node by clicking Settings > All Settings > Add Node.

2. On the Add Application Monitors step, use the Show Only drop-down to select a category of Application Monitors and display a list of Component Types.

For a list of Component Types, see the [SAM Template Reference](#).
3. Select the Application Monitor(s) you want to assign.
4. Select or enter the appropriate credentials, and then click Next.
5. On the Change Properties page, modify any of the information as needed, such as the Node Status Polling interval (in seconds) and the Collect Statistics Every frequency (in minutes).
6. Click OK and complete adding the node.

**Customize Application Details views**

You may need to log in with an administrator account to perform this action.

Applications inherit the custom view setting from their templates in the same way as other template settings. You can customize the templates through overrides or by creating a copy of the template and editing that content.

1. Click Settings > All Settings > SAM Settings > Manage Application Monitors.
2. Find an assigned Application Monitor without a Custom View.
3. Return to Applications > SAM Summary then click the Application Monitor to view its Application Details page.
4. Click Customize Page, then proceed with your customizations.

**Create a custom Application Details view**

You can select and create a custom Application Details view for different applications. This custom view is applied at the template level. Applications inherit the custom view setting from their templates in the same way as other template settings.

1. Click Settings > All Settings > SAM Settings > Manage Application Monitors.
2. Select an Application Monitor and then click Edit Properties.
3. Click Modify Template Settings in the field Custom Application Details View.
4. Select "Yes, use <TemplateName> Details View in the field Custom Application Details View".
5. Click Submit, then click the name of the assigned Application Monitor to view its Application Details page.
6. Click Customize Page and proceed with your customizations.

After creating a customized application details view for a template, you can change the Custom Application Details View setting in the template properties to switch between the default view and the custom view.

Select a view for a template

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Select a template and then click Edit.
3. If you want applications based on this template to use the default view, set Custom Application Details View to "No, use Default Application Details View".
4. If you want applications based on this template to use the custom view, set Custom Application Details View to "Yes, use <TemplateName> Details View" where <TemplateName> is the name of the selected template.
5. Click Submit.

Select a view for an application

1. Click Settings > All Settings > SAM Settings > Manage Application Monitors.
2. Select an Application Monitor and then click Edit Properties.
3. Click Modify Template Settings in the field Custom Application Details View.
4. If you want this application to use the default view, set Custom Application Details View to "No, use Default Application Details View."
5. If you want this application to use the custom view, set Custom Application Details View to "Yes, use <TemplateName> Details View" where <TemplateName> is the name of the selected template.
6. Click Submit.

Customize Application Summary views

Orion Web Console views are configurable presentations of network information that can include maps, charts, summary lists, reports, events, and links to other resources.

In SAM, the Application Summary page provides additional details about your applications and servers, such as related alerts, potential warranty issues on servers, and Top 10 lists to keep you up-to-date on your busiest servers. The page includes various default widgets (formerly called "resources") that you can customize as necessary.

You can edit the current SAM Application Summary view, or create a copy on the Manage Views page to refine with filters and widgets for your own summary.

💡 Check out this video about the customizing SAM views and widgets.
You may need to log in with an administrator account to perform this action.

1. Click Settings > All Settings > Manage Views.
2. Locate and select Application Summary, and click Copy.
3. Select the Copy of the Application Summary and click Edit.
4. Edit the view name, add and remove widgets, modify column layouts, and more, as necessary.
5. Click [+] in a column where you want to add SAM widgets. The Add Resource page displays a list of available widgets.

To list SAM-specific widgets such as Top XX Monitored Processes, AppStack Environment, and AppInsight widgets, select Classic category in the Group By field and then scroll down to select SAM Application Summary Resources.

6. Click Preview to verify the data and layout. If satisfied, click Done.

To learn more about views, including how to limit data shown on views access or how to add NOC views, see the Orion Platform Administrator Guide.
Viewing node and application data in tooltips

You can hover over objects such as nodes and groups to view additional information. These tooltips also include SAM-specific data, providing at-a-glance updates and data for quick responses. You can click the objects to access details pages for deeper data or to manage alerts and use management options.

Node tooltips

<table>
<thead>
<tr>
<th>TOOLTIP DATA</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Status</td>
<td>Current status of the node: up, down, warning, unplugged, or unmanaged</td>
</tr>
<tr>
<td>IP Address</td>
<td>The IP address currently assigned to the node</td>
</tr>
<tr>
<td>Machine Type</td>
<td>The vendor icon and vendor description of the node</td>
</tr>
<tr>
<td>Average Response Time</td>
<td>The measured average response time of selected node, as of the last node poll</td>
</tr>
<tr>
<td>Packet Loss</td>
<td>The percent of all transmitted packets that are lost by the node, as of the last node poll</td>
</tr>
<tr>
<td>CPU Load</td>
<td>The percent of available processing capacity on the node that is currently used, as of the last node poll</td>
</tr>
<tr>
<td>Memory Used</td>
<td>The percent of available memory on the node that is currently used, as of the last node poll</td>
</tr>
<tr>
<td># of Running VMS</td>
<td>Number of running VMS and total VMS</td>
</tr>
<tr>
<td>ESX Host Status</td>
<td>Status of the ESX Host</td>
</tr>
</tbody>
</table>

Application tooltips

<table>
<thead>
<tr>
<th>TOOLTIP DATA</th>
<th>DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Name</td>
<td>The name of the application</td>
</tr>
<tr>
<td>App Status</td>
<td>The status of the application: up, down, unknown, warning, or critical</td>
</tr>
</tbody>
</table>
### ToolTip Data | Details
--- | ---
Server Status | Operational status of the server: up, down, warning, unplugged, or unmanaged
Components with Problems | List of the components with problems and their statuses

## Add SAM data to the Node Details view

To view SAM data, including application and component monitor information, you may need to edit the Node Details - Summary view to include relevant views and widgets (formerly called "resources").

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > Manage Views.
2. Select the Node Details - Summary and click Edit.
3. On the Customize Node Details page, click [+] in the column where you want to add SAM widgets.

To list SAM specific widgets on the Add Resource page, select the Group By Classic Category and select SAM Application Summary Resources. These options include widgets such as the All Applications Tree, AppStack Environment, and AppInsight widgets.
4. Click Add Selected Resources to save all selections to the column.

5. Click Done to update the view.

To check the view, navigate to a node you edited to see the latest layout. Edit the view again to modify if necessary.

To learn more about customizing views and widgets, see the Orion Platform Administrator Guide.
Manage Asset Inventory in SAM

As you add more hardware, software, and applications to your environment, your assets increase along with the need to adjust the existing setup to expand and scale your infrastructure. To keep up, you must proactively monitor hardware and application health, while at the same time maintaining an up-to-date inventory of all your IT assets.

To handle these demands, you can use SAM’s Asset Inventory dashboard to maintain a current and detailed inventory of a node's hardware and software, including both physical and virtual assets. This information can also benefit those interested in tracking asset depreciation, gathering information for insurance purposes, or managing and maintaining your infrastructure.

Additional benefits of the Asset Inventory dashboard include:

- Gain visibility into current hardware and software on a node.
- Track warranty information for critical hardware components.
- Be notified about important software, operating system, and firmware updates.
- Monitor overall hardware life cycles and maintenance timelines

Here are some examples of Asset Inventory data available for individual nodes:

![Server Warranty Summary](image)

- Expired Warranties (5 warranties)
- Warranties Due to Expire (6 warranties)

<table>
<thead>
<tr>
<th>Next 5 Warranties Set to Expire.</th>
<th>EXPIRATION DATE</th>
<th>DAYS LEFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>JABBUTILITY5</td>
<td>3/13/2014</td>
<td>164</td>
</tr>
<tr>
<td>stp-esx-03.lab.txt</td>
<td>3/14/2014</td>
<td>164</td>
</tr>
<tr>
<td>CHEWBMAIL</td>
<td>3/14/2014</td>
<td>164</td>
</tr>
<tr>
<td>cal-vcenter-01</td>
<td>3/14/2014</td>
<td>165</td>
</tr>
<tr>
<td>TUL-WDSRV-01</td>
<td>3/22/2014</td>
<td>172</td>
</tr>
</tbody>
</table>
Note the following details about Asset Inventory data collection:

- Inventory details displayed depend on the polling method used to collect data from the node.
- SAM polls nodes for Asset Inventory data daily, but you can adjust the default interval.
- You can configure how SAM handles certificate errors during Asset Inventory polling.
- Asset Inventory polling functions independently of hardware health monitoring; either feature can be used on its own.
- Asset Inventory monitoring does not count against your licenses.

To start collecting Asset Inventory data in SAM, you'll need to:

1. Review Asset Inventory polling requirements.
2. Download, install, and configure agent software from third-party vendors. Click here for details.
3. Run Discovery to detect host agent software from supported hardware vendors and enable Asset Inventory polling across multiple nodes automatically.

You can also enable Asset Inventory polling for individual nodes.
When SAM enables Asset Inventory polling for discovered nodes, it also enables **hardware health monitoring**. Polling jobs begin immediately, and are later scheduled to occur at random times over a 24-hour period. If necessary, you can **adjust the default interval for Asset Inventory polling**.

**To display Asset Inventory data in the Orion Web Console:**

1. Navigate to the Node Details page for a node that is being polled for Asset Inventory data.
2. Click the Asset Inventory option in the left menu.

See also [Troubleshoot asset inventory issues](#).
Asset Inventory polling requirements

The following systems support data collection for the Asset Inventory dashboard:

- Dell servers with OpenManage Server Administrator Managed Node 7.2 or later
- HP servers with HP System Insight Manager 6.2 or higher
- HPE ProLiant Gen10 Servers with SIM 8.0 or later (via SNMP protocol only)
- IBM server with IBM Director (Common Agent, 6.3 or higher)

Additional hardware may be supported with a limited amount of data returned by polling.

SAM supports the following operating system and protocols to collect asset inventory data:

<table>
<thead>
<tr>
<th>OPERATING SYSTEM</th>
<th>PROTOCOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>SNMP, WMI</td>
</tr>
<tr>
<td>Linux</td>
<td>SNMP</td>
</tr>
<tr>
<td>AIX v7 and higher</td>
<td>SNMP</td>
</tr>
<tr>
<td>VMware ESX/ESXi v4.x and v5.x</td>
<td>CIM, VMware API</td>
</tr>
<tr>
<td>VMware API for ESX/ESXi hosts</td>
<td>HTTPS</td>
</tr>
<tr>
<td>polling via vCenter</td>
<td></td>
</tr>
</tbody>
</table>

Use SNMP for VMware nodes not polled via CIM or VMware API.

Download agent software

SAM uses standard protocols such as SNMP and WMI to collect Asset Inventory data, but not all information is available natively from an OS without installing the hardware vendor's required agent software. For example, if Windows cannot recognize the serial number of a machine, SAM cannot determine the machine's warranty status unless the vendor updates its agent software to extend APIs and gather additional data.

To download agent software for supported devices, visit the SolarWinds Success Center and see Third-party software required to monitor hardware health and collect asset inventory information in SAM.
Enable Asset Inventory polling for individual nodes

During Discovery, the Asset Inventory dashboard (also called a "subview") is automatically enabled for each node that meets Asset Inventory polling requirements. There are numerous ways for you to adjust this setting (for example, if someone disabled it).

- For existing nodes, enable Asset Inventory in the Management widget on the Node Details page, as described below.
- For new nodes added via the Manage Nodes page, enable Asset Inventory on the List Resource page in the Add Node wizard.
- To enable Asset Inventory polling for a large group of nodes, run a Discovery to "re-import" the nodes. This will not affect anything currently monitored, but Asset Inventory will be enabled on all hosts it's applicable to during the import process. See Enable Asset Inventory in bulk for details.

The Asset Inventory option is available only for nodes that meet Asset Inventory dashboard requirements, and host agent software from supported hardware vendors; click here for details.

To enable Asset Inventory data collection for a node via the Management widget:

1. Navigate to the Node Details page by clicking a node.
2. In the Management widget, click List Resources.
3. Select the Asset Inventory check box.

To disable Asset Inventory data collection for a node, clear the Asset Inventory check box. To disable Asset Inventory across all nodes, see Remove Asset Inventory data collection in the Success Center.
Adjust Asset Inventory polling frequency

Asset Inventory data collection is automatically enabled during Discovery. The data collected uses less than 100 KB of database space per node. For a larger environment consisting of roughly 1,000 servers, the total Asset Inventory data should total just under 100 MB.

Required sub-views are enabled by default for nodes with Asset Inventory enabled.

Inventory data does not need to be collected with the same degree of regularity as status information so the impact on your polling engine is minimal. Asset Inventory data collection occurs once daily and you can configure the polling interval to suit your needs.

To change the default Asset Inventory collection period:

You may need to log in with an administrator account to perform this action.

1. Click Settings > All Settings > Orion Polling Settings.
2. Adjust the number of days for the Default Asset Inventory Poll Interval field.
   Note: 1 is the minimum allowable value for this field.

3. Click Submit.

Configure how SAM handles certificate errors during Asset Inventory polling

By default, SAM ignores certificate errors during Asset Inventory polling. You can update that setting so users are prompted to verify untrusted connections before proceeding.

To configure SAM to prompt users to verify connections with invalid certificates during Asset Inventory polling:

1. Log into the Orion Web Console as an administrator.
2. Navigate to the default folder, C:\Program Files (x86)\SolarWinds\Orion\AssetInventory\SolarWinds.AssetInventory.Collectors.dll.config
3. Using a text editor, open SolarWinds.AssetInventory.Collectors.dll.config
4. Scroll to `<appSettings>` and change "false" to "true" in the AssetInventoryVerifyServerCertificate line, as shown here:

   `<add key="AssetInventoryVerifyServerCertificate" value="true" />`

5. Save your changes.

6. Use the Orion Service Manager to shut down and restart the SolarWinds Collector Service.

**Troubleshoot asset inventory issues**

This section provides various tips for handling issues you may encounter when using SAM's Asset Inventory dashboard.

- SAM uses standard protocols such as SNMP and WMI to collect Asset Inventory data, but not all information is available natively from an OS without installing the hardware vendor's required agent software. For example, if Windows cannot recognize the serial number of a machine, SAM cannot determine the machine's warranty status unless the vendor updates its agent software to extend APIs and gather additional data. See [Third-party software required to collect Asset Inventory information in SAM](#).
- If expected asset data does not appear for a node that hosts an Orion agent, check the agent configuration.
- Internet Information Services (IIS) use drivers and other "non-software" records that may result in duplicate records being displayed in the Orion Web Console and saved to the Orion SQL database.
- Certificate errors found during polling are ignored by default, but you can change that setting.
- If multiple registry keys share the same display name, filter by display name, version, and related Windows updates so only records for a specific asset appear in the Software Inventory widget and are saved in the Orion SQL database.
- WMI polls many WMI classes and registers for asset data. For the Software Inventory resource, SAM polls information from the following registry branches:
  - SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall
  - SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- Up to 2100 WMI nodes can be monitored on a single polling engine. If you exceed that limit, consider adding another polling engine. See [Scalability Engine Guidelines for SolarWinds products](#).

The following Success Center articles contain additional tips:

- [Asset Inventory polling of Windows Updates via WMI may stop processes on monitored machines](#)
- [Asset Inventory Polling is not showing all installed Software on Linux Hosts](#)
- [Asset Inventory of SAM is not providing Manufacturer, Model or Serial Number](#)
- [Warranty status displays as Unknown for HP servers in the Asset Inventory widget](#)
- [Object Identifiers (OIDs) used by SAM to poll for Asset Inventory data using SNMP polling method](#)

You can also visit the SolarWinds online IT community, THWACK.
Manage processes, services, tasks, and events in real time with SAM

SAM provides several ways to monitor real-time events, including options to create component monitors from events and handle Windows tasks for monitored servers.

- **Real-Time Process Explorer**
- **Service Control Manager**
- **Windows Scheduled Task Monitor**
- **Real-Time Event Log Viewer**

**Real-Time Process Explorer**

The Real-Time Process Explorer (RTPE) is available for WMI and SNMP monitored nodes and displays monitored and unmonitored processes directly in SAM. The advantage of the RTPE is that you no longer need to physically or remotely log in to a computer and run the Task Manager to retrieve that machine's vital statistics.

You can view the Real-Time Process Explorer displays on the details page for an application through the Component Details pages and Node Details page.

- Only SAM administrators can end processes as well as enable and disable the RTPE.
- The User Name and Command Line columns are hidden by default.
- Using the RTPE on a node monitored via ICMP, which has no working component, requires you to select Windows credentials manually. Consider promoting the selected node to SNMP or WMI to avoid this prompt.

*Pop-ups must be enabled for the Real-Time Process Explorer to be viewed.*

To access the Real-Time Process Explorer:

1. In the Orion Web Console, click My Dashboards > SAM Summary page.
2. In the All Applications group, click an Application.
A new window opens, accessing the server and application data with administrative credentials. To use different credentials, select the Use Different Credentials option. All processes list by name with checkboxes to select and end processes.

Monitor unmonitored processes in SAM

Processes currently monitored by SAM are indicated by the application icon and name of the assigned application. Processes that are not currently monitored by SAM are indicated by the [+] symbol, followed by the words, Start monitoring.

1. From the RTPE, click Start monitoring.
2. From the Edit Properties section of the Component Monitor Wizard you can begin setting up the selected component monitor.

Use the Real-Time Process Explorer in SAM

Different information is displayed in the Real-Time Process Explorer (RTPE) depending on which protocol you use to monitor a node. The table below shows the differences in the information gathered based on the protocol used.

<table>
<thead>
<tr>
<th>Monitor Process</th>
<th>WMI</th>
<th>SNMP</th>
<th>Orion Agent for Linux/Unix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Process ID</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Assigned Application</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CPU usage</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Monitor Process</td>
<td>WMI</td>
<td>SNMP</td>
<td>Orion Agent for Linux/Unix</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----</td>
<td>------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Physical Memory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Virtual Memory</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Disk I/O</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>User Name</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Command Line</td>
<td>Yes*</td>
<td>Yes*</td>
<td>Yes*</td>
</tr>
<tr>
<td>Polling interval</td>
<td>Five seconds</td>
<td>Up to two minutes</td>
<td></td>
</tr>
</tbody>
</table>

Windows servers update their SNMP statistics every two minutes. It takes two updates to these statistics to provide an accurate calculation. Data displayed in the RTPE via SNMP can take up to four minutes to display.

* Information on certain processes may not be exposed which can result in certain rows being blank.

By default, all available columns, with the exception of the User Name and Command Line, are shown for the top ten running processes. Each column can be sorted, added, or removed by clicking the column head and then clicking the drop down arrow.

Show all running processes

Click Show All in the bottom-left of the window.

Show a set number of running processes

Change the number in the text box next to the Show All button.

Pause polling

If available, click Pause Polling in the top-left of the window.

End processes

Check the boxes next to the processes you want to end and then click End Process. This option is only available when the RTPE is using a WMI connection.

Start polling

Click Start Polling in the top-left of the window.

For Windows-based nodes, change the credentials by clicking the Use Different Credentials button to bring up the credential library dialog box.

Clicking Refresh will re-poll the running processes.
Real-Time Process Explorer alerts

Three alerts are included with the Real-Time Process Explorer:

- **High CPU Percent Utilization with Top 10 Processes**
  This alert sends an email when the CPU utilization is greater than 80%.

- **High Physical Memory Utilization with Top 10 Processes**
  An alert is sent when physical memory usage is at or above 90%.

- **High Virtual Memory Utilization with Top 10 Processes**
  An alert is sent when virtual memory usage is at or above 90%.

Use the Alert Manager to create, edit, delete, enable, or disable alerts. You can access the Alert Manager in one of three ways:

- **Settings Page (Recommended)**
  - Click Settings > All Settings in the menu bar. Under Alerts & Reports, click Manage Alerts.

- **Active Alerts Details**
  - From the Active Alerts Details page, click Manage Alerts in the Management resource.

- **Node Details**
  - On the Node Details page, navigate to the All Alerts this Object can trigger resource, and then click Manage Alerts.

For details, see the Orion Platform Administrator Guide.

ℹ️ Alerts may lag when monitoring hardware via SNMP. It takes two updates to these statistics to provide an accurate calculation, meaning an alert can take up to four minutes to reach its recipient. To expedite this process, change the protocol to WMI, which updates every five seconds. Also, consider adjusting the trigger time to a value greater than two minutes.

Real-Time Process information in top 10 alerts

SAM provides additional troubleshooting information for high CPU, memory, and virtual memory by sending email alerts. This is done by utilizing the Top Offending Processes metric running on the server at the time of the alert.

Find the executable path `SolarWinds.APM.RealTimeProcessPoller.exe` and its command line arguments below.

**Command line argument syntax:**

```
```

**Command line argument variables**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>ID of a Node (NodeID), which is polled.</td>
</tr>
</tbody>
</table>
### VARIABLE

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-count</td>
<td>The number of processes to show.</td>
</tr>
<tr>
<td>-sort</td>
<td>The criteria used for the selection of top processes. Process:</td>
</tr>
<tr>
<td></td>
<td>• CPU - Processor time. This is the default value if the command line argument is not specified.</td>
</tr>
<tr>
<td></td>
<td>• PhysicalMemory - Process physical memory.</td>
</tr>
<tr>
<td></td>
<td>• VirtualMemory - Process virtual memory.</td>
</tr>
<tr>
<td></td>
<td>• DiskIO - Process disk I/O per second.</td>
</tr>
<tr>
<td>-timeout</td>
<td>Timeout for polling in seconds.</td>
</tr>
<tr>
<td>-alert</td>
<td>The AlertDefID of associated triggered alert. If this argument is provided, then alert notes are updated with the results from polling.</td>
</tr>
<tr>
<td>-activeObject</td>
<td>The ActiveObject property of the associated triggered alert. If this argument is not provided, NodeID is used.</td>
</tr>
</tbody>
</table>

### Example 1

This example returns the top 20 processes with the highest virtual memory consumption running on the host with node ID 123.

```bash
SolarWinds.APM.RealTimeProcessPoller.exe -n=123 -count=20 -sort=VirtualMemory -timeout=300
```

### Example 2

This example uses the Execute an External Program alert action:

```bash
SolarWinds.APM.RealTimeProcessPoller.exe -n=${NodeID} -alert=${AlertDefID}
```

### Example 3

This example uses the Execute an External Program alert action for an alert defined for the Volume object type:

```bash
SolarWinds.APM.RealTimeProcessPoller.exe -n=${NodeID} -alert=${AlertDefID} -activeObject=${NetObjectID} -sort=VirtualMemory
```

### Real-Time Event Log Viewer

View Windows event logs in real-time using the WMI protocol with the Real-Time Event Log Viewer (RTEV). Event logs can be filtered by log type, event source, and the level of severity.
The viewer allows you to:

- Start monitoring selected real-time Windows event log entries
- Pause and restart polling
- Log into the selected server with different credentials

You may need to log in with an administrator account to perform this action.

Access the Real-Time Event Log Viewer

2. Select an application from the All Applications widget.

The Real-Time Event Log Viewer opens with a filterable list of all events and logs.

Pop-ups must be enabled for the RTEV to be viewed.
Filter Real-Time Events

After the events of the selected Log Type are collected and appear in the window on the right, you can filter the results with various criteria.

1. Select an option from the log type drop-down menu:

   ![Select Log Type](image)

2. Select Custom Sources from the Event Sources drop-down menu. Select only the sources you want to keep when the filter is applied, as highlighted:

   ![Event Sources](image)

3. Select the type of messages you want the filter to keep by selecting the Event Level.
4. Click Apply Filter to have the events filtered and displayed.

After filtering is complete, hide and unhide the filtering pane on the left by clicking either of the two arrows, highlighted in red:

The display window shows a list of the most recent events. Should any new events occur while this window is open, a green bar at the top of the window indicates that new events have arrived. Click the green bar to add these new events to the display window.

The Level column icons correspond to the Event Level icons in the legend of the events pane.

Clicking any message in the display window brings up a message box providing the entire message along with additional details.

**Monitor events**

You can monitor selected events from the Real-Time Event Log Viewer by selecting an event and creating a component monitor:

1. Select an event and click Start Monitoring in the Message Details view.
   
   The Add Component Monitor wizard opens with the selection.

2. Modify the Component Monitor based on the application you selected, following the wizard steps.

3. After you begin the Component Monitor Wizard, use the option to Disable Keyword Matching. The Include Events drop-down menu provides options to help you filter results.

4. Save and assign the component monitor.
Service Control Manager

The Service Control Manager (SCM) is similar to the Real Time Process Explorer, except you can use it to manage the services of monitored Windows nodes instead of processes. The advantage of enabling SCM is that you do not have to physically, or remotely, log into a Windows computer to view and control its services. The status of services appears directly in SAM via the Service Control Manager.

Services viewed in the Service Control Manager are polled every 25 seconds using WMI.

Accessed via the Management widget on Node Details pages, the Service Control Manager shows all services on the monitored computer. Hover over the row of any service to display details about it.
Enable the Service Control Manager

To enable the tool:

1. Click Settings > All Settings > Manage Accounts.
2. Locate the account and click Edit.
3. Expand the Server & Application Monitor Settings section.
4. For the Service Control Manager option, select Yes.
5. Save changes to the account.

Access the Service Control Monitor

You can access the Service Control Manager through the Management widget on Node Details pages.

- You may need to log in with an administrator account to perform this action.
- Pop-ups must be enabled in your browser to view the Service Control Manager.

The Service Control Manager button is not available on the Node Details page if the Top 10 resources are hidden.

1. Click My Dashboards > Home.
2. In the All Nodes group, click a node.
3. On the Node Details page, locate the Management widget.
4. Click Service Control Manager.

The Service Control Manager display services, sorted alphabetically.

Start and stop a service

Services monitored by SAM are indicated by the application icon and name of the assigned application. Services that are not monitored by SAM are indicated by the [+] symbol, followed by the words, "Start monitoring this service."

1. Click the row of a service to select it.
2. At the top of the window, select Stop Service, Start Service, or Restart service.
Monitor a service with SAM

1. Click Start monitoring this service in the row of the service to monitor.
2. When the Edit Properties section of the Component Monitor wizard opens, begin customizing the selected component monitor.

The table below shows the default settings for the Service Control Manager:

<table>
<thead>
<tr>
<th>ACTION</th>
<th>DESCRIPTION</th>
<th>DEFAULT SETTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Job Life Time</td>
<td>Specifies how long the information for a service exists.</td>
<td>3 min</td>
</tr>
<tr>
<td>Service Poll Interval</td>
<td>Specifies the refresh frequency of data reported by the service.</td>
<td>25 sec</td>
</tr>
<tr>
<td>Service Action Job Timeout</td>
<td>Reports the timeout for a service.</td>
<td>3 min</td>
</tr>
</tbody>
</table>

Windows Scheduled Task Monitor

The Windows Scheduled Task Monitor (WSTM) is a widget that provides quick visual access to the status of scheduled tasks configured on your Windows nodes.

- Only tasks from root directory are monitored. The WSTM does not work for tasks from every level of the Task Scheduler Library.

- The WSTM consumes five SAM license units per node.

Within the WSTM widget, you can:

- Hover over any item in the Task Name column to display details about a task.
- Sort the displayed tasks by clicking the head of each column.
You can use the Edit option to:

- Change the polling frequency
- Change the polling timeout period
- Change credentials
- Add custom notes

You can also use the Windows Task Monitor in alerting: Alert me when task last run result is non successful

Requirements

The WSTM includes an alert that can notify you about task execution failures, as well as web-based reports that show all scheduled tasks configured across all monitored servers in your environment. Additionally, there is a dedicated Task Failure Report you can view on-screen or have emailed to you.

ℹ️ This widget is hidden when the WSTM is not being monitored on a node.

Enable the Windows Scheduled Task Monitor

To enable the Windows Scheduled Task Monitor:

ℹ️ You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > Manage Nodes.
2. Click [+ ] Add Node and then select the Windows Servers: WMI and ICMP option and enter the credentials.
3. Click Next and then select Windows Scheduled Tasks.
4. Complete the wizard as instructed.

The Windows Schedule Tasks option is:

- Checked when tasks are already being monitored
- Unchecked by default when at least one task is found on the target machine
- Hidden when a task is not found on the target machine

Access the Windows Scheduled Task Monitor

You can access the Windows Scheduled Task Monitor widget via the All Nodes widget.
1. Click My Dashboards > Home > All Nodes resource.

2. In the All Nodes widget on the Orion Summary Home page, expand the Windows tree by clicking Windows, and then click a Windows node to access its Node Details page.

3. On the Node Details page, find the Windows Scheduled Task widget, as shown here:

![Windows Scheduled Tasks widget](image)

When monitored, the Windows Schedule Task Monitor widget is only on the Node Details view of the monitored server.

See also:
- Windows Scheduled Task Monitor status codes
- Windows Scheduled Task Monitor status
- Windows Scheduled Tasks (THWACK)

## Windows Scheduled Task Monitor status codes

This table includes status codes that may appear in the Windows Schedule Task Manager widget.

<table>
<thead>
<tr>
<th>HEX Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0x0)</td>
<td>The operation completed successfully.</td>
</tr>
<tr>
<td>(0x1)</td>
<td>Incorrect or unknown function called.</td>
</tr>
<tr>
<td>(0x2)</td>
<td>File not found.</td>
</tr>
<tr>
<td>(0xA)</td>
<td>The environment is incorrect.</td>
</tr>
<tr>
<td>(0x41300)</td>
<td>The task is ready to run at its next scheduled time.</td>
</tr>
<tr>
<td>(0x41301)</td>
<td>The task is currently running.</td>
</tr>
<tr>
<td>(0x41302)</td>
<td>The task will not run at the scheduled times because it was disabled.</td>
</tr>
<tr>
<td>(0x41303)</td>
<td>The task has not yet run.</td>
</tr>
<tr>
<td>(0x41304)</td>
<td>There are no more runs scheduled for this task.</td>
</tr>
<tr>
<td>(0x41305)</td>
<td>One or more of the properties needed to run this task on a schedule were not configured.</td>
</tr>
<tr>
<td>HEX CODE STATUS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(0x41306)</td>
<td>The last run of the task was terminated by the user.</td>
</tr>
<tr>
<td>(0x41307)</td>
<td>Either the task has no triggers or the existing triggers are disabled or not set.</td>
</tr>
<tr>
<td>(0x41308)</td>
<td>Event triggers do not have set run times.</td>
</tr>
<tr>
<td>(0x4131B)</td>
<td>The task is registered, but not all specified triggers will start the task.</td>
</tr>
<tr>
<td>(0x4131C)</td>
<td>The task is registered, but may not start. Enable the Batch log in privilege for the task principal.</td>
</tr>
<tr>
<td>(0x41325)</td>
<td>The Task Scheduler service has asked the task to run.</td>
</tr>
<tr>
<td>(0x8004020D)</td>
<td>Cannot modify or delete an object that was not added using the COM+ Admin SDK.</td>
</tr>
<tr>
<td>(0x80041309)</td>
<td>A task's trigger is not found.</td>
</tr>
<tr>
<td>(0x8004130A)</td>
<td>One or more of the properties required to run this task were not configured.</td>
</tr>
<tr>
<td>(0x8004130B)</td>
<td>There is no running instance of the task.</td>
</tr>
<tr>
<td>(0x8004130C)</td>
<td>The Task Scheduler service is not installed on this computer.</td>
</tr>
<tr>
<td>(0x8004130D)</td>
<td>The task object could not be opened.</td>
</tr>
<tr>
<td>(0x8004130E)</td>
<td>The object is either an invalid task object or is not a task object.</td>
</tr>
<tr>
<td>(0x8004130F)</td>
<td>No account information was found in the Task Scheduler security database for the task indicated.</td>
</tr>
<tr>
<td>(0x80041310)</td>
<td>Unable to establish existence of the account specified.</td>
</tr>
<tr>
<td>(0x80041311)</td>
<td>Corruption was detected in the Task Scheduler security database. The database was reset.</td>
</tr>
<tr>
<td>(0x80041312)</td>
<td>Task Scheduler security services are available only on Windows NT.</td>
</tr>
<tr>
<td>(0x80041313)</td>
<td>The task object version is either unsupported or invalid.</td>
</tr>
<tr>
<td>(0x80041314)</td>
<td>The task was configured with an unsupported combination of account settings and run time options.</td>
</tr>
<tr>
<td>(0x80041315)</td>
<td>The Task Scheduler Service is not running.</td>
</tr>
<tr>
<td>(0x80041316)</td>
<td>The task XML contains an unexpected node.</td>
</tr>
<tr>
<td>(0x80041317)</td>
<td>The task XML contains an element or attribute from an unexpected namespace.</td>
</tr>
<tr>
<td>(0x80041318)</td>
<td>The task XML contains a value which is incorrectly formatted or out of range.</td>
</tr>
<tr>
<td>(0x80041319)</td>
<td>The task XML is missing a required element or attribute.</td>
</tr>
<tr>
<td>HEX CODE STATUS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(0x8004131A)</td>
<td>The task XML is malformed.</td>
</tr>
<tr>
<td>(0x8004131D)</td>
<td>The task XML contains too many nodes of the same type.</td>
</tr>
<tr>
<td>(0x8004131E)</td>
<td>The task cannot be started after the trigger end boundary.</td>
</tr>
<tr>
<td>(0x8004131F)</td>
<td>An instance of this task is already running.</td>
</tr>
<tr>
<td>(0x80041320)</td>
<td>The task will not run because the user is not logged in.</td>
</tr>
<tr>
<td>(0x80041321)</td>
<td>The task image is corrupt or was tampered with.</td>
</tr>
<tr>
<td>(0x80041322)</td>
<td>The Task Scheduler service is not available.</td>
</tr>
<tr>
<td>(0x80041323)</td>
<td>The Task Scheduler service is too busy to handle your request. Please try again later.</td>
</tr>
<tr>
<td>(0x80041324)</td>
<td>The Task Scheduler service attempted to run the task, but the task did not run due to a constraint in the task definition.</td>
</tr>
<tr>
<td>(0x80041327)</td>
<td>The task has properties that are not compatible with earlier versions of Windows.</td>
</tr>
<tr>
<td>(0x80041328)</td>
<td>The task settings do not allow the task to start on demand.</td>
</tr>
<tr>
<td>(0x80070002)</td>
<td>The system cannot find the file specified.</td>
</tr>
<tr>
<td>(0x800700C1)</td>
<td>Not valid Win32 application.</td>
</tr>
<tr>
<td>(0x800704C7)</td>
<td>The operation was canceled by the user.</td>
</tr>
<tr>
<td>(0x800704DD)</td>
<td>The operation being requested was not performed because the user was not logged into the network. The specified service does not exist.</td>
</tr>
<tr>
<td>(0xC000013A)</td>
<td>The application terminated.</td>
</tr>
<tr>
<td>(0xC06D007E)</td>
<td>An unknown software exception occurred.</td>
</tr>
</tbody>
</table>

For a complete list of error codes, see [Task Scheduler Error and Success Constants](#).

### Windows Scheduled Task Monitor status

The following table lists the conditions that must be met in order for a status of Available and Not Available.

The **Undefined** status is the default status.

**Available** status triggers for:

- Task Success
- Task Ready
- Task Running
• Task Disabled
• Task Has Not Run
• Task No More Runs
• Task Not Scheduled
• Task Terminated
• Task No Valid Triggers
• Event Trigger
• Some Triggers Failed
• Batch Log on Problem
• Task Queued

**Not Available** status triggers for:

• Trigger Not Found
• Task Not Ready
• Task Not Running
• Service Not Installed
• Cannot Open Task
• Invalid Task
• Account Information Not Set
• Account Name Not Found
• Account Database Corrupt
• No Security Services
• Unknown Object Version
• Unsupported Account Option
• Service Not Running
• Unexpected Node
• Name space
• Invalid Value
• Missing Node
• Malformed XML
• Too Many Nodes
• Past End Boundary
• Already Running
• User Not Logged On
• Invalid Task Hash
• Service Not Available
• Service Too Busy
• Task Attempted
• Task Not V1 Compatible
• Start On Demand
• Service Does not Exist
- No File Specified
- Cannot Delete Object
- Not Valid Application
- Incorrect Function
- File Not Found
- Environment Is Incorrect
- Application Terminated
- Unknown Software Exception
- Operation Canceled
Monitor your network in SAM

Like all Orion Platform products, SolarWinds Orion offers immediate insight into the performance of your network. As you view data, you can access investigate further to troubleshoot alerts, forecast usage, and modify configurations.

Summary pages display high-level events and alerts, and performance data for immediate insight into all managed nodes for network devices, applications, virtual servers, and more. Use links in widgets to drill-down further into information, such as server resources and consumption, hardware, allocated virtual resources, application performance, and OS metrics.

Devices you want to monitor must be added to the SolarWinds Orion database. See Discover and Add Network Devices.

To monitor immediate needs in your environment:

- View notifications by clicking the bell icon at the top of the Orion Web Console.
- View events, alerts, traps, and Syslogs by clicking Alerts & Activity > Message Center.

This section includes the following topics:

- Monitor with Orion agents in SAM
- Monitor your virtual infrastructure in SAM
- Monitor your environment using AppStack
- Use alerts to monitor your environment with SAM
- Monitor application dependencies in SAM
- Monitor hardware health in SAM

See the Orion Platform Administrator Guide, which describes common Orion Web Console features, to learn how to:

- Monitor cloud instances and VMs
- Monitor container services
- Monitor Cisco UCS devices
- Group monitored objects
- View events, alerts, traps, and Syslogs in the Orion Web Console Message Center
- Use Performance Analysis (PerfStack™) dashboards
- Manage SNMP traps
- Monitor Quality of Experience (QOE) metrics
- Integrate SAM with ServiceNow® to generate tickets based on critical events and alerts

Additional monitoring tools are discussed in Monitor with AppInsight applications and Manage SAM templates and component monitors.
Monitor with Orion agents in SAM

An agent is a software application that provides a communication channel between the Orion server and a Windows or Linux computer. Agents are used as an alternative to WMI or SNMP to provide information about your selected key devices and applications.

When using the Orion agent, SAM collects all data from application monitors for that server. SAM can deploy and consume collected data using Orion agents for Windows, Linux, and AIX.

For full details on Orion agents, see the Orion Platform Administrator Guide, which describes:

- Agent settings and communication modes
- Configuring SNMP for agents
- Using agents with Linux-based computers
- Handling Agent deployment and migrations
- Troubleshooting Orion agent issues

Polling issues may be related to how SNMP was configured for Orion agents during deployment. See Configure SNMP for agents deployed by SAM.

Overriding agent and agentless monitoring

You can override data collection behavior in specific instances per application monitor or template. You can configure an application or template to collect data through a preferred polling method as agent or agentless. For example, for the User Experience Monitor template you may not want to measure response time locally from the server where the application is installed. If this is the case, you can switch the application to poll without using an agent.

![Advanced Preferences Panel]

Generally, agents provide richer data streams while adding load to the application, while agentless can only pull limited data (based on the API and security access permissions) but can isolate the load to an external resource and do not impact the application as severely.

Different features are available based on monitoring methods, as described in these articles:

- Comparison of Windows agent versus agentless, using SNMP or WMI
- Comparison of Linux agent versus agentless
- Comparison of AIX agent versus agentless
Communication type

Agent Communication can be deployed as either Active or Passive:

- **Agent initiated Communication (Active):** The Agent initiates communication with the server on the default port of 17778 (this can be changed if needed). This port must be opened on the server firewall so the Agent can connect. No change to the Agent firewall is required. For more information, see:

- **Orion Server initiated Communication (Passive):** The Agent waits for requests from the server on the default port of 17790 (this can be changed if needed). This port must be opened on the Agent computer's firewall so the server can connect. No change to the server firewall is required.

Agent resource consumption

The following table details agent resource consumption:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>0.24% on average under normal operating condition</td>
</tr>
<tr>
<td>Memory</td>
<td>Between 10 and 100 MB depending upon the number and types of jobs</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>The agent consumes roughly 20% (on average) of the bandwidth consumed by the WMI protocol for transmission of the same information. For example, an agent may use 1.3 KBPS versus WMI at 5.3 KBP.</td>
</tr>
</tbody>
</table>

The amount of bandwidth consumed by an agent depends on how many items are being monitored. For example, as the number of SAM components monitored by an agent increases, bandwidth increases. Agents utilize approximately 80% less bandwidth than WMI for the same number of items being monitored because agents use a more efficient protocol, HTML, while WMI uses RPC for communication. The agent protocol also utilizes compression, which is not available in WMI.

| Storage   | 100 MB                                                   |

Templates and component monitors

Application templates provide collections of application monitors with configurable settings to collect and monitor data for Orion managed nodes. These templates include a variety of options based on operating systems, applications, and services.

For details, prerequisites, and configuration requirements for application templates and monitors, see [SAM Component Monitor Types](#) and [SAM Template Reference](#).

To use component monitors in Linux environments with the Orion agent for Linux, you need to complete additional steps, as described in [Linux/Unix system configurations for component monitors](#).
After being applied to a node, a template becomes an application, which is comprised of component monitors, also known as performance counters. Applications created from templates are used to report metrics based on your needs. See Create your first application and alert to learn about building a template-based application to monitor WMI.

The following component monitors support the Orion Agent for Windows:

- Process Monitor - Windows
- DNS Monitor - TCP
- DNS Monitor - UDP
- Exchange Web Services User Experience Monitor
- HTTP Monitor
- HTTPS Monitor
- Performance Counter Monitor
- SMTP Monitor
- TCP Port Monitor
- ODBC User Experience Monitor
- Oracle User Experience Monitor
- Windows Event Log Monitor
- Windows PowerShell Monitor
- Windows Service Monitor
- Windows Script Monitor
- WMI Monitor

The following component monitors support the Orion Agent for Linux:

- Directory Size Monitor
- File Age Monitor
- File Change Monitor
- File Count Monitor
- File Existence Monitor
- File Size Monitor
- HTTP User Experience Monitor
- HTTPS User Experience Monitor
- JMX Monitor
- Linux/Unix Script Monitor
- Nagios Script Monitor
- ODBC User Experience Monitor
- Process Monitor
- Oracle User Experience Monitor
- SNMP Monitor
- TCP Port Monitor
- TomCat Server Monitor
The following component monitors support the Orion Agent for AIX:

- Directory Size Monitor
- File Age Monitor
- File Change Monitor
- File Count Monitor
- File Existence Monitor
- File Size Monitor
- HTTP User Experience Monitor
- HTTPS User Experience Monitor
- JMX Monitor
- Linux/Unix Script Monitor
- Nagios Script Monitor
- ODBC User Experience Monitor
- Process Monitor
- SNMP Monitor

  - SNMPv3 with encrypted mode is not supported.
  - To poll SNMP details correctly, you need to manually enable the SNMP daemon.

- TCP Port Monitor
- TomCat Server Monitor

Reports

The following reports can be used to track Orion agent usage:

- Agent Inventory
- Agent Plugin Version

Configure SNMP for agents deployed by SAM

SNMP is a polling method used by Orion agents to collect data about monitors and resources. Review this information to locate and resolve any issues when troubleshooting connectivity, access, and data polled in SAM.

The following components and widgets depend on SNMP connections:

- Location, Contact, and SysObjectID of the Node Details resource
- SAM SNMP component monitor
- Hardware Health data and widgets
- Asset Inventory page and widgets

  - Hardware Health and Asset Inventory are not supported on AIX devices.
Configure SNMP

To configure SNMP, you need to install the SNMP daemon if it is missing. The Orion agent configures SNMP for you during deployment. You can manually configure SNMP. If settings need to be modified for Orion agent, the configurations are verified and updated during agent deployment.

On AIX computers, the auto-configuration is not supported. Make sure that SNMP daemon is running.

If you need to install the SNMP daemon, you can install the daemon using a command according to your Linux/Unix distribution. Deploying the agent automatically configures SNMP. If the SNMP daemon is not installed, SNMP cannot be configured.

To install SNMP daemon on Linux:

- Install on Ubuntu: `sudo apt-get -y install snmpd`
- Install on Red Hat / CentOS: `yum -y install net-snmp`
- Install on SUSE: `zypper -y install net-snmp`

Auto-configuration of SNMP on Linux executes on the following actions:

- The first step during a Discovery
- Anytime you initiate a List resource on any node managed by the Orion agent for Linux

SAM completes the following changes during auto-configuration:

- Checks the `snmpd` service is configured to start automatically after system reboots.
- Checks the `snmpd.conf` file for the configuration of an agent owned community string. If the string is missing, SAM adds the community string. The string is "agent owned" = with an SNMP v2 community string in form of an agent GUID (randomly generated value that is unique for each agent).

For example:

```bash
### BEGIN SolarWinds Agent SNMP auto config
rocommunity 36343901-D61F-4C72-B860-A8E18DD892E4 localhost
### END SolarWinds Agent SNMP auto config
```

If SAM adds the community string, SNMP is restarted.

Configure SNMP v3

If you have SNMP v3 configurations detecting users through the configuration file without an SNMP v2 community string, SAM will not modify the configuration file. SNMP v3 will not be auto-configured on deployment. You can provide SNMP credentials manually at the node level to configure access.

Prerequisites

To add SNMP credentials, you can enter the credentials through the Add Node Wizard or edit an existing node. You cannot test the credentials until the agent is deployed.

page 62
For encryption, you also need to install a common python extension `pycrypto` on the target machine to make polling possible. To install:

```
PyCrypto installation on ubuntu
apt-get install python-pip
pip install pycrypto
```

**Test Credentials**

You can test any manually entered credentials by editing the Node. To test, you need a properly connected agent deployed to the target node. Review the credentials and click Test.
Enable the SNMP daemon on AIX computers

Use of the SNMP Monitor on AIX devices requires that the SNMP daemon be enabled. Otherwise, the following information will not be available in the Node Details widget:

- Location
- Contact
- SysobjectID

To enable the SNMP daemon, update the configuration file in `/etc/snmpdv3.conf` which configures snmpv1, v2c, and v3 access. Use the following configuration example as a reference:

```
# Two snmpv1 community strings: public commstr1
VACM_GROUP group1 SNMPv1 public -
VACM_GROUP group1 SNMPv1 commstr1 -
VACM_VIEW group1view internet - included -
VACM_ACCESS group1 - - noAuthNoPriv SNMPv1 group1view - -
COMMUNITY public public noAuthNoPriv 0.0.0.0 0.0.0.0 -
COMMUNITY commstr1 commstr1 noAuthNoPriv 0.0.0.0 0.0.0.0 -

# snmpv2c community string: swiagent
VACM_GROUP group2 SNMPv2c swiagent -
VACM_VIEW group2view internet - included -
VACM_ACCESS group2 - - noAuthNoPriv SNMPv2c group2view - -
COMMUNITY swiagent swiagent 0.0.0.0 0.0.0.0 -
COMMUNITY public public - 0.0.0.0 0.0.0.0 -

# snmpv3 user with no authorization: user1
USM_USER user1 - none - - - - -
VACM_GROUP group4 USM user1 -
VACM_VIEW group4view internet - included -
VACM_ACCESS group4 - - noAuthNoPriv USM group4view - -

DEFAULT_SECURITY no-access - -
```
Test the SNMP configuration

- To test the configuration for v1 requests, use `snmpdinfo`.
- To test v2c and v3 requests, use `clsnmp` for v2c and v3 requests and configure `/etc/clsnmp.conf`.

```
 v2cstring localhost snmpv2c
 v3user_noauth localhost snmpv3 user1 - - - - - -
```

This configuration matches the examples above.

**Monitor your virtual infrastructure in SAM**

You can use the Virtual Infrastructure Monitor (VIM) feature included in SAM to enable virtual monitoring directly from the Orion Web Console. For virtualization software requirements, see [Supported versions](#).

Prerequisites for monitoring your virtual infrastructure include:

- ICMP is enabled on virtual servers (recommended).
- [Credentials for the Orion server exist on ESX servers](#).
- Your virtual infrastructure was already [discovered](#).

To learn more about using VIM to manage your virtual infrastructure, see [Virtualization Manager (VMAN) online help](#). The [Orion Platform Administrator Guide](#) also contains details about monitoring virtual environments.
Monitor your environment using AppStack

The Application Stack, or AppStack, refers to the various layers that make up a complex application delivery infrastructure. The AppStack Environment view is an interactive visual mapping within the Orion Platform that provides an in-depth perspective through an entire environment to help identify the root cause of performance and availability issues.

The AppStack automatically gathers information about objects in your environment, as well as their respective relationships, and displays them in a customizable view. Some relationships, such as groups and dependencies, can be customized.

The AppStack environment supports both physical and virtual node types.

The AppStack is different from the PerfStack that you can use to build custom dashboards of time series data based on collected performance metrics.

To access the AppStack, click My Dashboards > Environment. Filter the application view with filter options and filter properties. You can also save and reuse these filters as layouts to display specific views of the AppStack.

In the AppStack view, you can hover-over each node and application to see a tooltip of data and status. This view provides quick insight into issues in the stack and overall statistics. Select each item on the view to load related items throughout the environment. See AppStack data overview for details on icons and colors of icons for at-a-glance monitoring.

The next section describes how to use AppStack as a troubleshooting tool. To further understand and customize the AppStack Environment, see:

- AppStack categories and data
- Customize the AppStack view in SAM
- Use filters for AppStack
- Create an AppStack layout
- AppStack data overview
Troubleshoot with AppStack

Using AppStack as a troubleshooting tool is fairly straightforward. Simply navigate to the AppStack Environment view, and click an object to see related objects. To investigate further, navigate to the Details page of the object or any related object.

For example, the illustration below displays all objects in the current environment. The application, MSSQLSERVER, is in a Critical state, as indicated by its icon. To see what this application is related to, click the MSSQLSERVER icon.

As you select objects in the AppStack, the system automatically updates the view to select and highlight related objects. In this example, MSSQLSERVER is selected. Once it is selected, related objects are highlighted by the fading of unrelated objects, as shown:

To completely hide unrelated objects in the view, click Spotlight in the top right corner. You can also select an individual object and click Spotlight to hide it.
To navigate to the details page of a selected object, click the icon of the selected object next to the chart at top or double-click the larger icon in the main view. You can review and managed the critical and warning issues per node.

Introduction to the AppStack Environment view

The AppStack Environment view is designed to provide you with a powerful layer of troubleshooting visibility by exposing all participating objects in your environment, as well as their relationships to one another. This view is useful in assessing the overall health of your environment, as well as troubleshooting specific and related problems.

To help you better understand how this visualization of your environment can be used for troubleshooting, the following analogy was designed to provide you with a simple, high-level overview of how AppStack processes and displays information.
The AppStack Environment view Analogy

Imagine you work for a company comprised of many employees that work in various departments. In this example, you work in the Engineering department as an engineer:

Management

- Design Team
- Engineering
- Marketing
- Finance

Operations

- Reception
- Operations
- Manufacturing
- Packaging
- Quality Control
- Security
- Suppliers

You
Now imagine you're sick and cannot work for two weeks. Certain employees in various departments are affected by your absence. Others are not be affected. The illustration below highlights the employees who *are* affected by your absence by fading those who are *not* affected.

Focusing on only the people who *are* affected by your absence can be beneficial from a task management perspective. Therefore, completely hiding those *not* affected by your absence gives a clean and concise view, displaying only the necessary information.

Now, instead of employees, imagine this is about your entire IT environment. The goal of the AppStack Environment view is to help you quickly assess the impact a given object has on its related objects. This relational view is the AppStack Environment view found in the Orion Web Console.
The AppStack Environment view

The AppStack Environment view, shown below, displays the status of individual objects in your IT environment through the Orion web console. Objects are categorized and ordered from left to right, with the worst status being shown on the left side of the view.
Clicking an object shows everything related to that object by fading unrelated objects, as shown:

Clicking Spotlight removes the faded, unrelated objects from the view entirely.

Object selected.

Object selected shown with related objects.
Clicking the selected object at the top, or double-clicking it in the view, takes you to the Details page for that object for further investigation.

**AppStack categories and data**

By default, the following AppStack Environment categories are available in the Orion Platform:

- Groups
- Containers
- Chassis
- Applications
- Database Instances (with Orion Integrated DPA)
- Servers
- Hosts
- Virtual Clusters
- Virtual Data Centers
- Virtual Centers
- Volumes

Other Orion Platform products offer additional categories. Empty categories can be hidden via the Change Layout Settings option in the drop-down menu.

The Category Status Summary (located to the right of each category) parses, enumerates, and displays the number of objects in each category, emphasizing the status as indicated by the colors. The Category Status Summary can be toggled to either display at all times, or only when categories are collapsed.

You can manage the setting through the Change Layout Settings option.

**Supported categories and data**

The following matrix outlines categories associated with the AppStack Environment view, which vary by SolarWinds product.

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>SRM</th>
<th>SAM</th>
<th>VMAN</th>
<th>WPM</th>
<th>DPA</th>
<th>NPM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>Manage your environment by organizing monitored objects logically into groups, regardless of device type or location.</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>SRM</td>
<td>SAM</td>
<td>VMAN</td>
<td>WPM</td>
<td>DPA</td>
<td>NPM</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>------------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Containers</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td>Lightweight, executable packages of software that include everything needed to run an application: code, system tools, system libraries, and settings. Containers isolate applications and their dependencies into self-contained units that can run anywhere without interfering with each other.</td>
</tr>
<tr>
<td>Chassis</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td>A physical structure that houses one or more servers.</td>
</tr>
<tr>
<td>Applications</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A collection of SAM component monitors grouped together to collect specific metrics concerning the application as a whole.</td>
</tr>
<tr>
<td>Database Instances</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td>An organized collection of data. The displayed database instances including Microsoft SQL, MySQL, and Oracle.</td>
</tr>
<tr>
<td>Transactions</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td>A recording of web browser steps assigned to a specific location.</td>
</tr>
<tr>
<td>Steps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
<td>A collection of actions. For example, the actions required to navigate to a specific URL make up one step.</td>
</tr>
<tr>
<td>Servers</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A computer capable of accepting requests from the client and giving responses accordingly. The server makes services, as access to data files, programs, and peripheral devices, available to workstations on a network.</td>
</tr>
<tr>
<td>Hosts</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A server running a hypervisor for virtualization that can host multiple VMs.</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>SRM</td>
<td>SAM</td>
<td>VMAN</td>
<td>WPM</td>
<td>DPA</td>
<td>NPM</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Virtual Clusters</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A group of VMs installed at distributed servers from one or more physical clusters. VMs in a virtual cluster are logically connected by a virtual network across several physical networks. Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.</td>
</tr>
<tr>
<td>Virtual Data Centers</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A centralized virtual repository to store, manage, and disseminate data related to a particular body of knowledge or pertaining to a particular business.</td>
</tr>
<tr>
<td>Virtual Centers</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A server that acts as a centralized management application to manage VMs and ESXi. Use a vSphere client to access vCenter Server and manage ESXi servers.</td>
</tr>
<tr>
<td>Data Stores</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>A repository is a set of data object sets modeled using classes defined in a database schema. A data store may also store simpler types such as flat files. Some data stores represent data in only one schema, while others use several schemas for this task.</td>
</tr>
<tr>
<td>Volumes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>A volume, or logical drive, is a single accessible storage area with a single file system, typically resident on a single partition of a hard disk.</td>
</tr>
<tr>
<td>LUNS</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>A Logical Unit Number (LUN) is used to identify a logical unit, which is a device addressed by the SCSI or SAN protocols that encapsulate SCSI, such as Fibre Channel or iSCSI. A LUN may be used with any device which supports read/write operations, such as a tape drive, but usually refers to a logical disk created on a SAN.</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>SRM</td>
<td>SAM</td>
<td>VMAN</td>
<td>WPM</td>
<td>DPA</td>
<td>NPM</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>------------------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NAS Volumes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Network-attached storage (NAS) is a type of dedicated file storage device</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>that provides LAN users with centralized, consolidated disk storage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>through a standard Ethernet connection.</td>
</tr>
<tr>
<td>Pools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A storage pool (also called a RAID array) is a collection of disk drives</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>that become a logical entity. When you create a storage pool, you select</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>the desired capacity (number of disk drives) and assign a RAID level to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>it which provides a redundancy level.</td>
</tr>
<tr>
<td>VServers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A virtual storage server (VServer) contains data volumes and one or more</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>logical interfaces (LIFs) through which it serves data to clients. A VServer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>can contain multiple FlexVol volumes, or a single Infinite Volume.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A VServer securely isolates the shared virtualized data storage and network,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>and appears as a single dedicated server to its clients. Each VServer has</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>a separate administrator authentication domain and can be managed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>independently by a VServer administrator.</td>
</tr>
<tr>
<td>Storage Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Storage Arrays consist of two or more disk drives built into a stand-alone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>unit. Storage arrays provide increased availability, resiliency, and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>maintainability by using existing components (controllers, power supplies,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>fans, and so on) often up to the point where all single points of failure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>are eliminated from the design.</td>
</tr>
</tbody>
</table>
This AppStack provides a sample of categories and multiple nodes in various health states:

AppStack data overview

There are several ways you can quickly determine object status in the AppStack Environment, including:

- **Object status and colors**
- **Quick links**
- **Overview bar**

Object status and colors

Objects in the AppStack have colors based on their current status, providing a quick summary of your environment. Objects will be in one of the following states, indicated by icon color and shape.

<table>
<thead>
<tr>
<th>Status</th>
<th>Color</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>Red</td>
<td><img src="example.png" alt="Red Icon" /></td>
</tr>
<tr>
<td>Critical</td>
<td>Pink</td>
<td><img src="example.png" alt="Pink Icon" /></td>
</tr>
<tr>
<td>Warning</td>
<td>Yellow</td>
<td><img src="example.png" alt="Yellow Icon" /></td>
</tr>
<tr>
<td>Unknown</td>
<td>Gray</td>
<td><img src="example.png" alt="Gray Icon" /></td>
</tr>
<tr>
<td>Status</td>
<td>Color</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Unreachable</td>
<td>Black</td>
<td>x</td>
</tr>
<tr>
<td>Up</td>
<td>Green</td>
<td>📈</td>
</tr>
<tr>
<td>Other</td>
<td>Blue</td>
<td>-</td>
</tr>
<tr>
<td>External</td>
<td>Purple</td>
<td>🌈</td>
</tr>
</tbody>
</table>

You can only filter against statuses actively monitored in the AppStack. For example, if no objects have a status of Down, the Down status is not available for filtering.

Servers can only provide an Up or Down status. To check CPU, memory, or hardware health status, hover over the server icon to view the detailed tooltip. The reported status for applications and LUNs is based on performance thresholds.

The AppStack Environment view offers a parent/child relationship. (Parents can be either servers or hosts.) This relationship is represented as a mixed icon. The parent is the larger of the two icons that is hosting the child. For example, the following illustration graphically shows that an application (child) is Down on a node (parent) that is Up.

Quick links

Use the Quick Links bar at the top of the AppStack to quickly open and close object categories. You can also show or hide the names of objects with a distressed status.
Expand All

Open all categories to reveal objects in each category. Each category can be individually expanded and collapsed by clicking [+ and -] next to the category name.

Collapse All

Close all categories, hiding the objects within each category. Each category can be individually expanded and collapsed by clicking [+ and -] next to the category name.

Show/Hide Names

Toggle between showing and hiding the names for each object in a distressed state. If more objects exist than can be displayed, the category name displays the number of displayed objects followed by the total number of objects. For example, (87 of 111), meaning only 87 objects of 111 total objects in this category are being displayed. This provides a numerical summary of your environment.

Click More (located after the last visible object in the category) to display the next 50 objects.

To change the default number of objects shown in each category, click Default Layout > Change Layout Settings.
Overview bar

The Overview bar summarizes your environment in a compact space. If your view is currently filtered, the filtered objects display above the Overview bar, as shown.

The doughnut chart in the Overview bar displays the ratio of objects to one another in all possible states using indicative colored slices. The total number of objects in your environment is also displayed.

Objects under the Issues header are parsed and enumerated according to status. Objects shown here are in one of the following distressed states: Down, Critical, Warning, Unknown and Unreachable. An icon and count only appear if one or more objects is currently in that state.

Customize the AppStack view in SAM

To customize the AppStack view, click Default Layout > Change Layout Settings in the top right of the page. Modifying settings can boost performance, such as limiting the amount of objects displayed, hiding empty categories, and adjusting the update interval.

You may need to log in with an administrator account to perform this action.
You can modify the following settings:

- Limit on Up Objects Shown per Category: Objects in the AppStack Environment view are ordered from left to right, with the worst status being shown on the left side of the view. Limit the number of Up objects to provide more room for objects needing attention.
- Object Names: Displays distressed objects.
- Align Objects: By default, all objects are left-aligned.
- Show Category Status Summary: The Category Status Summary are the colored numbers to the right of each category, indicating the number of objects in a particular state. This can visible at all times, or only when categories are collapsed.
- Empty Categories: Select this option to hide categories with no objects, giving the view a cleaner look.
- Update Interval: Change the time between AppStack refreshes.

Use filters for AppStack

You can filter AppStack data by status, display name, and applications to create different layouts for data to speed troubleshooting for servers and virtual servers.

You may need to log in with an administrator account to perform this action.

Filter options

By default, every object displays in the AppStack Environment view. To display only objects with a certain status, filter the view by selecting one or more statuses and then applying the filter. You can also filter objects based on various properties, as well as keywords.
The filtering sidebar includes the following options:

- **STATUS**
  - Down (44)
  - Critical (11)
  - Warning (11)
  - Unknown (22)
  - Unreachable (12)
  - Up (562)
  - External (1)

- **DISPLAY NAME**
  - Search
  - CUPS
  - Low Free Space
  - IBM DB2

**Hides/Shows the Status Filter.**
Expands/Collapses the options presented in the current listing.

**Number of objects with the listed status.**

**Toggles the displaying of less common statuses to select from.**

**Text box and button used for adding keywords.**
Keywords currently added. Uncheck each keyword to temporarily omit it from the search. To remove the keyword completely, click [X].

**Allows you to add various categorized properties to the filter. This opens a pop-up window in order for you to make your selections.**

**Clears all status selections, keywords, and property filters.**

**Applies the filter with the statuses, keywords, and properties selected.**

ℹ️ Statuses only become available for filtering when an object in your environment is in that state. For example, if no objects have a status of Down, the Down status will not be available when filtering.

**Creating filters**

Creating filters using the AppStack filter properties provides more options to refine the content displayed in the AppStack.

1. Click My Dashboards > Environment, and then click [+] Add Filter Properties.
2. In the pop-up menu, select one or more objects from the Orion Object drop down list.
3. Click Add column to display the new filters.

Create an AppStack layout

A Layout is a user-defined filter with configured settings you can save and use when viewing the AppStack. You can tailor layouts with filters to view specific objects of interest. A default layout is provided that contains all currently monitored objects visible in the AppStack Environment. You cannot modify the default layout. As you create and save layouts, load them via the Layout menu.

To create a layout:

1. Create a filter by selecting filter options in the sidebar or selecting filter properties.
2. Apply the filter to update the AppStack.
3. Under the Layout menu, click Save As New Layout. Enter a name and save.

You can modify the layout by making changes to the filter and clicking Save. If you want to save as a new layout, click Save as New Layout. To remove a layout, select the layout as currently used and select Delete this Layout.
For example, to create a Layout where only IIS objects in the AppStack Environment view are shown, add the IIS filter property. Apply the filter and save it as a new layout.

The layout is added to the Layout menu, where you can easily navigate to it. The dot next to a layout in the menu indicates the layout currently used.

Use alerts to monitor your environment with SAM

An alert is an automated notification that a network event occurred; for example, when a server doesn't respond. The network event that triggers an alert is determined by conditions you set up when you configure your alert. You can schedule alerts to monitor your network during a specific time period, and create alerts that notify different people based on when the alert is triggered.

The types of events you can create alerts for vary, depending on the Orion Platform products installed. For example, in NPM you can create an alert to notify you if a node in a specific location goes down or if the network response time is too slow. With SAM, you can receive alerts about application response times or when your Exchange mailbox database is almost full.

You can create alerts for any monitored object, and alert against volumes and nodes with most Orion Platform products. To get started with alerts in the Orion Web Console, click Settings > All Settings > Manage Alerts.
See the Orion Platform Administrator Guide to learn more about alerts, including:

- Alert preconfiguration tasks
- Best practices and tips
- Creating alerts

Use SAM alert variables

SAM supplements the alerting abilities of the Orion Platform with several SAM-specific configurable alerts and alert variables.

- Component monitor properties for alerts
- Component Monitor variables for alerts
- Application Monitor properties for alerts
- Application Monitor variables for alerts

To learn more about Orion Platform alerts, see the Orion Platform Administration Guide.

Component monitor properties for alerts

The following alert properties are available for component monitors.

**Component Name**

This allows you to base your alert criteria on component names.

**Component Type**

This allows you to base your alert criteria on component types. Specify the component monitor type by value using the following table.

<table>
<thead>
<tr>
<th>COMPONENT MONITOR TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP User Experience Monitor</td>
<td>35</td>
</tr>
<tr>
<td>Directory Size Monitor</td>
<td>38</td>
</tr>
<tr>
<td>DNS Monitor - TCP</td>
<td>4</td>
</tr>
<tr>
<td>DNS Monitor - UDP</td>
<td>5</td>
</tr>
<tr>
<td>DNS User Experience Monitor</td>
<td>15</td>
</tr>
<tr>
<td>Download Speed Monitor</td>
<td>25</td>
</tr>
<tr>
<td>Exchange Web Services User Experience Monitor</td>
<td>51</td>
</tr>
<tr>
<td>File Age Monitor</td>
<td>36</td>
</tr>
<tr>
<td>File Change Monitor</td>
<td>23</td>
</tr>
<tr>
<td>File Count Monitor</td>
<td>39</td>
</tr>
<tr>
<td>Component Monitor Type</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>File Existence Monitor</td>
<td>28</td>
</tr>
<tr>
<td>File Size Monitor</td>
<td>22</td>
</tr>
<tr>
<td>FTP Monitor</td>
<td>7</td>
</tr>
<tr>
<td>FTP User Experience Monitor</td>
<td>24</td>
</tr>
<tr>
<td>HTTP Form Login Monitor</td>
<td>27</td>
</tr>
<tr>
<td>HTTP Monitor</td>
<td>6</td>
</tr>
<tr>
<td>HTTPS Monitor</td>
<td>14</td>
</tr>
<tr>
<td>IMAP4 Monitor</td>
<td>13</td>
</tr>
<tr>
<td>IMAP4 User Experience Monitor</td>
<td>30</td>
</tr>
<tr>
<td>JMX Monitor</td>
<td>49</td>
</tr>
<tr>
<td>LDAP User Experience Monitor</td>
<td>34</td>
</tr>
<tr>
<td>Linux/Unix Script Monitor</td>
<td>21</td>
</tr>
<tr>
<td>Log Parsing Monitor</td>
<td>61</td>
</tr>
<tr>
<td>MAPI User Experience Monitor</td>
<td>31</td>
</tr>
<tr>
<td>Nagios Script Monitor</td>
<td>50</td>
</tr>
<tr>
<td>NNTP Monitor</td>
<td>11</td>
</tr>
<tr>
<td>ODBC User Experience Monitor</td>
<td>16</td>
</tr>
<tr>
<td>Oracle User Experience Monitor</td>
<td>18</td>
</tr>
<tr>
<td>Performance Counter Monitor</td>
<td>37</td>
</tr>
<tr>
<td>POP3 Monitor</td>
<td>12</td>
</tr>
<tr>
<td>POP3 User Experience Monitor</td>
<td>29</td>
</tr>
<tr>
<td>Process Monitor – SNMP</td>
<td>8</td>
</tr>
<tr>
<td>Process Monitor - WMI</td>
<td>1</td>
</tr>
<tr>
<td>RADIUS User Experience Monitor</td>
<td>40</td>
</tr>
<tr>
<td>SMTP Monitor</td>
<td>10</td>
</tr>
<tr>
<td>SOAP Monitor</td>
<td>58</td>
</tr>
<tr>
<td>SQL User Experience Monitor</td>
<td>17</td>
</tr>
<tr>
<td>COMPONENT MONITOR TYPE</td>
<td>VALUE</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>SNMP Monitor</td>
<td>32</td>
</tr>
<tr>
<td>SQL User Experience Monitor</td>
<td>17</td>
</tr>
<tr>
<td>TACACS+ User Experience Monitor</td>
<td>41</td>
</tr>
<tr>
<td>TCP Port Monitor</td>
<td>2</td>
</tr>
<tr>
<td>Tomcat Server Monitor</td>
<td>33</td>
</tr>
<tr>
<td>Web Link Monitor</td>
<td>26</td>
</tr>
<tr>
<td>Windows Event Monitor</td>
<td>42</td>
</tr>
<tr>
<td>Windows PowerShell Monitor</td>
<td>45</td>
</tr>
<tr>
<td>Windows Script Monitor</td>
<td>20</td>
</tr>
<tr>
<td>Windows Service Monitor</td>
<td>9</td>
</tr>
<tr>
<td>WMI Monitor</td>
<td>19</td>
</tr>
</tbody>
</table>

Component Status
This allows you to alert on Critical, Down, Unknown, Up, and Warning status.

Response Time
This allows you to alert on response time.

Statistic Data
This allows you to alert on statistic data.

Process (Service) Name
This allows you to alert on the process or service name. For example: dns.exe, or AlertingEngine.

Process Instance Count
This allows you to alert on the instance count of a process.

Percent CPU
This allows you to alert on the percentage of CPU in use of a monitored process or service.

Percent Physical Memory
This allows you to alert on the percentage of physical memory in use of a monitored process or service.

Percent Memory Used
This allows you to alert on the percentage of total memory in use of a monitored process or service.
**Percent Virtual Memory**

This allows you to alert on the percentage of virtual memory in use of a monitored process or service.

**Virtual Memory Used**

This allows you to alert on the amount of virtual memory in use, in bytes, of a monitored process or service.

**Component Monitor variables for alerts**

The following variables are available when selecting APM-Component as the property type.

> "APM" was the original product name for SAM.

<table>
<thead>
<tr>
<th>COMPONENT MONITOR VARIABLE</th>
<th>MACRO</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>${(APM: ComponentDetailsURL)}</code></td>
<td><code>${(N=SwisEntity;M=DetailsUrl)}</code></td>
<td>Hyperlink to the Component Details page that triggered the alert.</td>
</tr>
<tr>
<td><code>${ApplicationId}</code></td>
<td><code>${(N=SwisEntity;M=Application.ApplicationID)}</code></td>
<td>Provides the unique numeric identifier of the application. This value is analogous to the node ID.</td>
</tr>
<tr>
<td><code>${ApplicationName}</code></td>
<td><code>${(N=SwisEntity;M=Application.ApplicationAlert.ApplicationName)}</code></td>
<td>Provides the name of the monitored application.</td>
</tr>
<tr>
<td><code>${ApplicationStatus}</code></td>
<td><code>${(N=SwisEntity;M=Application.Status)}</code></td>
<td>Provides the status of the application.</td>
</tr>
<tr>
<td><code>${ComponentId}</code></td>
<td><code>${(N=SwisEntity;M=ComponentAlert.ComponentID)}</code></td>
<td>Provides the numeric component ID of the specific application.</td>
</tr>
<tr>
<td><code>${ComponentMessage}</code></td>
<td><code>${(N=SwisEntity;M=ComponentAlert.ComponentMessage)}</code></td>
<td>Message sent when alerting on component status.</td>
</tr>
<tr>
<td><code>${ComponentName}</code></td>
<td><code>${(N=SwisEntity;M=ComponentAlert.ComponentName)}</code></td>
<td>Provides the name of the component, for example, SW Module Engine.</td>
</tr>
<tr>
<td>COMPONENT MONITOR VARIABLE</td>
<td>MACRO</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>${ComponentStatus}</td>
<td>${N=SwisEntity;M=Status}</td>
<td>Provides the status of the specific component.</td>
</tr>
<tr>
<td>${ComponentType}</td>
<td>${N=SwisEntity;M=ComponentAlert.ComponentType}</td>
<td>Provides the numeric component type.</td>
</tr>
<tr>
<td>${DisplayType}</td>
<td>${N=SwisEntity;M=ComponentAlert.DisplayType}</td>
<td>Provides the display type for the specific monitor. For example, Windows Service.</td>
</tr>
<tr>
<td>${LastTimeUp}</td>
<td>${N=SwisEntity;M=ComponentAlert.LastTimeUp}</td>
<td>Provides the date and time the component was last seen in the Up state.</td>
</tr>
<tr>
<td>${MemoryUsed}</td>
<td>${N=SwisEntity;M=ComponentAlert.MemoryUsed}</td>
<td>Provides the memory used by a component, in bytes.</td>
</tr>
<tr>
<td>${MultiValueMessages}</td>
<td>${N=SwisEntity;M=ComponentAlert.MultiValueMessages}</td>
<td>Message sent when alerting on the Multiple Value Chart.</td>
</tr>
<tr>
<td>${MultiValueStatistics}</td>
<td>${N=SwisEntity;M=ComponentAlert.MultiValueStatistics}</td>
<td>Statistics sent when alerting on the Multiple Value Chart.</td>
</tr>
<tr>
<td>${NodeID}</td>
<td>${N=SwisEntity;M=ComponentAlert.NodeId}</td>
<td>Provides the numeric node ID of the server on which the application is monitored.</td>
</tr>
<tr>
<td>${Percent ApplicationAvailability}</td>
<td>${N=SwisEntity;M=ComponentAlert.PercentApplicationAvailability}</td>
<td>Provides the availability of an application as a percentage.</td>
</tr>
<tr>
<td>COMPONENT MONITOR VARIABLE</td>
<td>MACRO</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>${Percent ComponentAvailability}</td>
<td>${N=SwisEntity;M=ComponentAlert. PercentComponentAvailability}</td>
<td>Provides the availability of a component as a percentage.</td>
</tr>
<tr>
<td>${PercentCPU}</td>
<td>${N=SwisEntity;M=ComponentAlert.PercentCPU}</td>
<td>Provides the amount of CPU used by a component as a percentage.</td>
</tr>
<tr>
<td>${PercentMemory}</td>
<td>${N=SwisEntity;M=ComponentAlert. PercentMemory}</td>
<td>Provides the memory used by a component as a percentage.</td>
</tr>
<tr>
<td>${PercentVirtualMemory}</td>
<td>${N=SwisEntity;M=ComponentAlert. PercentVirtualMemory}</td>
<td>Provides the virtual memory used by a component as a percentage.</td>
</tr>
<tr>
<td>${ProcessInstanceCount}</td>
<td>${N=SwisEntity;M=ComponentAlert. ProcessInstanceCount}</td>
<td>Provides the instance count of a process.</td>
</tr>
<tr>
<td>${ProcessName}</td>
<td>${N=SwisEntity;M=ComponentAlert.ProcessName}</td>
<td>Provides the process name.</td>
</tr>
<tr>
<td>${ResponseTime}</td>
<td>${N=SwisEntity;M=ComponentAlert.ResponseTime}</td>
<td>Provides the response time of a component.</td>
</tr>
<tr>
<td>${StatisticData}</td>
<td>${N=SwisEntity;M=ComponentAlert.StatisticData}</td>
<td>Provides the statistics data value of a component.</td>
</tr>
<tr>
<td>${StatusOrError Description}</td>
<td>${N=SwisEntity;M=ComponentAlert. StatusOrErrorDescription}</td>
<td>Provides the status of the component, including the full text of any error messages.</td>
</tr>
<tr>
<td>${Threshold-CPU-Critical}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds. ThresholdCPUCritical}</td>
<td>Provides the critical threshold for CPU.</td>
</tr>
<tr>
<td>${Threshold-CPU-Warning}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds. ThresholdCPUWarning}</td>
<td>Provides the Warning threshold for the CPU.</td>
</tr>
<tr>
<td>COMPONENT MONITOR VARIABLE</td>
<td>MACRO</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>${Threshold-PhysicalMemory-Critical}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdPhysicalMemoryCritical}</td>
<td>Provides the Critical threshold for physical memory.</td>
</tr>
<tr>
<td>${Threshold-PhysicalMemory-Warning}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdPhysicalMemoryWarning}</td>
<td>Provides the Warning threshold for physical memory.</td>
</tr>
<tr>
<td>${Threshold-ResponseTime-Critical}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdResponseTimeCritical}</td>
<td>Provides the Critical threshold for response time.</td>
</tr>
<tr>
<td>${Threshold-ResponseTime-Warning}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdResponseTimeWarning}</td>
<td>Provides the Warning threshold for response time.</td>
</tr>
<tr>
<td>${Threshold-Statistic-Critical}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdStatisticCritical}</td>
<td>Provides the Critical threshold for statistics.</td>
</tr>
<tr>
<td>${Threshold-Statistic-Warning}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdStatisticWarning}</td>
<td>Provides the Warning threshold for statistics.</td>
</tr>
<tr>
<td>${Threshold-VirtualMemory-Critical}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdVirtualMemoryCritical}</td>
<td>Provides the Critical threshold for virtual memory.</td>
</tr>
<tr>
<td>${Threshold-VirtualMemory-Warning}</td>
<td>${N=SwisEntity;M=ComponentAlertThresholds.ThresholdVirtualMemoryWarning}</td>
<td>Provides the Warning threshold for virtual memory.</td>
</tr>
<tr>
<td>${TimeStamp}</td>
<td>${N=SwisEntity;M=CurrentStatus.ObservationTimestamp}</td>
<td>Provides the last polling date and time of a component.</td>
</tr>
<tr>
<td>COMPONENT MONITOR VARIABLE</td>
<td>MACRO</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>${UserDescription}</td>
<td>${N=SwisEntity;M=ComponentAlert.UserDescription}</td>
<td>Provides a description of the component. Note: A default description is given by SolarWinds. Any changes you make will override the default description and be automatically saved.</td>
</tr>
<tr>
<td>${UserNotes}</td>
<td>${N=SwisEntity;M=ComponentAlert.UserNotes}</td>
<td>Provides notes from the user about a component. Changes you make will automatically be saved.</td>
</tr>
<tr>
<td>${WindowsEventMessages}</td>
<td>${N=SwisEntity;M=ComponentAlert.WindowsEventMessages}</td>
<td>Full details of the corresponding event.</td>
</tr>
<tr>
<td>${VirtualMemoryUsed}</td>
<td>${N=SwisEntity;M=ComponentAlert.VirtualMemoryUsed}</td>
<td>Provides the virtual memory used by a component, in bytes.</td>
</tr>
</tbody>
</table>

### Application Monitor properties for alerts

You can use the following properties in alerts for application monitors.

**Application Name**

This allows you to select the names of currently configured application templates as values.

**Application Status**

This allows you to select whether the application is in a Critical, Down, Unknown, Up, or Warning status.

### Application Monitor variables for alerts

The following variables are available when selecting APM-Application as the property type.
<table>
<thead>
<tr>
<th>Application Variable</th>
<th>Macro</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>${APM:ApplicationDetailsURL}</td>
<td>${N=SwisEntity;M=DetailsUrl}</td>
<td>Hyperlink to the Application Details page that triggered the alert, or the Details page of the application that triggered the component.</td>
</tr>
<tr>
<td>${Availability}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ApplicationAvailability}</td>
<td>Provides the status of the application.</td>
</tr>
<tr>
<td>${ComponentsWithProblems}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithProblems}</td>
<td>Provides a comma-delimited list of components in a Down, Unknown, Warning, or Critical state.</td>
</tr>
<tr>
<td>${ComponentsWithProblems Formatted}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithProblemsFormatted}</td>
<td>List of components that are not Up along with the component status, formatted with HTML tags. HTML formatting is used for the Send e-mail action to provide an improved appearance of the listed components.</td>
</tr>
<tr>
<td>${ComponentsWithProblems FormattedHtml}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithProblemsFormattedHtml}</td>
<td>List of components that are not Up along with the component status. Formatted with HTML tags for events that appear in the Orion Web Console.</td>
</tr>
<tr>
<td>${ComponentsWithStatus}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithStatus}</td>
<td>Provides a comma-delimited list of all components and their current status.</td>
</tr>
<tr>
<td>APPLICATION VARIABLE</td>
<td>MACRO</td>
<td>DEFINITION</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>${ComponentsWithStatusFormatted}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithStatusFormatted}</td>
<td>List of components with component status included formatted with html tags. Html formatting is used for send e-mail action to provide improved appearance of listed components.</td>
</tr>
<tr>
<td>${ComponentsWithStatusFormattedHtml}</td>
<td>${N=SwisEntity;M=ApplicationAlert. ComponentsWithStatusFormattedHtml}</td>
<td>List of components with the component status. Formatted with HTML tags for events that appear in the web console.</td>
</tr>
<tr>
<td>${ID}</td>
<td>${N=SwisEntity;M=Application. ApplicationID}</td>
<td>Provides the numeric application ID of the specific application.</td>
</tr>
<tr>
<td>${LastTimeUp}</td>
<td>${N=SwisEntity;M=CurrentStatus.LastTimeUp}</td>
<td>Provides the date and time the application was last seen in an Up state.</td>
</tr>
<tr>
<td>${Name}</td>
<td>${N=SwisEntity;M=Application. ApplicationAlert.ApplicationName}</td>
<td>Provides the name of the application that is triggering the alert.</td>
</tr>
<tr>
<td>${NodeID}</td>
<td>${N=SwisEntity;M=Application. Node.NodeID}</td>
<td>Provides the numeric node ID of the server on which the application is monitored.</td>
</tr>
<tr>
<td>${SystemSummaryFormatted}</td>
<td>${N=SwisEntity;M=Application. ApplicationAlert.SystemSummaryFormatted}</td>
<td>System summary. HTML formatting is used for the Send e-mail action to provide an improved appearance of the listed components.</td>
</tr>
</tbody>
</table>
Monitor application dependencies in SAM

SAM’s Application Dependencies feature provides a holistic view of application and server connections, expanding monitoring by detecting how applications and nodes interact with each other.

You can use data gathered during Application Dependencies polling to:

- Understand which applications, application processes, and nodes connect with each other.
- Ensure that the most important data for specific applications is monitored.
- Identify unmonitored applications and processes that require attention.
- Leverage latency and packet loss metrics to determine if an issue is caused by an application or the network.

This contextual visibility of relationships between applications and physical/virtual servers also reduces troubleshooting time. For example, instead of searching through many applications, nodes, and component monitors to determine why an application is slow, you can navigate to the Application Connections widget and analyze application dependencies to pinpoint the source of the issue.
To provide a more granular picture of application dependencies, the Connection Details page shows processes and ports for connections, plus node, application, process status, and (if Connection Quality polling is enabled) latency and packet loss statistics. The Connection Details page shows the entire communication stack from one node to another, which makes it a unique troubleshooting tool.

With the Application Dependencies feature, you can see where and how servers communicate with each other without contacting various teams to get information. You can use this feature to identify established connections and provide perspectives into the connections themselves.
To learn more, watch this video about discovering, monitoring and troubleshooting Application Dependencies.

For troubleshooting, Application Dependencies can help you determine if performance issues in an application are due to a server on one side or the other, or if issues are related to the actual communication between the two servers.

Application Dependencies polling overview

SAM uses two types of polling to collect application dependency data:

- Application Dependency polling: Discovers and monitors the following types of connections:
  - Application to application, in a typical client/server process monitored by SAM
    ![Application to application](image1)
  - Application to node, with a server process not currently monitored by SAM
    ![Application to node](image2)
  - Node to application, with a client application process not monitored by SAM
    ![Node to application](image3)

- Connection Quality polling: Tracks TCP communication traveling from client nodes hosting applications to target nodes. This synthetic polling collects latency and packet loss statistics for connections without intercepting network traffic, also known as "packet sniffing."

To learn more, see Manage polling for application dependencies.
How SAM uses Orion agents to monitor application dependencies

SAM uses Orion agents to flag nodes for Application Dependency polling. An agent must exist on at least one of the two nodes for which you want to display dependencies. If a node does not host an agent, the Application Connections widget displays sample data, as shown here.

Agent plug-in overview

The Orion Platform deploys and removes agent plug-ins as you enable and disable features in the Orion Web Console. A single agent-managed node may include multiple plug-ins to handle different tasks such as polling.

To support the Application Dependencies feature, SAM deploys agent plug-ins to nodes to monitor connections and network communications, then displays data on the Application Connections widget and Connection Details page.

To avoid having to update agent-plug-ins manually, make sure the Allow Automatic Agent Updates option is enabled on the Agent Settings page.

For Application Dependency polling:

- SAM deploys agent plug-ins to nodes if Application Dependency polling detects application-to-application or application-to-node connections.
- Agent plug-ins collect data about dependencies between applications (application-to-application connections) and/or nodes (application-to-node connections). It is available in Linux x64, Linux x86, and Windows versions.

To avoid performance issues, SAM does not deploy Application Dependencies plug-ins to the Main Polling Engine (that is, the Orion server).
For Connection Quality polling, if enabled:

- SAM deploys additional agent plug-ins to collect TCP latency and packet loss metrics.
- For Windows nodes connected to clients that host applications and application processes, TCP agent plug-ins include an Npcap driver to support Nping.

If you enable Connection Quality polling and disable it later, SAM removes the TCP agent plug-in but not the Npcap driver. See Remove an Npcap driver after disabling Connection Quality Polling in the SolarWinds Success Center.

SAM relies on server-initiated communications to detect "from" or "to" nodes, also called "passive agents" or "agentless" nodes. Only one node in a pair requires an agent plug-in. However, note that data gathered by polling depends on communication settings for both nodes, as described here:

- If target and client nodes both host agent plug-ins, SAM collects data via Application Dependency and Connection Quality polling for both nodes.
- If only the target node has an agent plug-in, SAM collects IP address and port data for the client node but not application details, process names, or connection statistics.
- If only the client node has an agent plug-in, SAM collects IP address and port data for the server node. If Connection Quality polling is enabled and SAM deployed a TCP agent plug-in to the connection source node, polling can capture latency and packet loss statistics.

Use the Manage Agents page to check the status of agent plug-ins.

To start monitoring connections between applications and nodes, configure the Application Dependencies feature, as described next.

See also:

- Access application dependency data in SAM
- Manage polling for application dependencies
- Troubleshoot application dependency issues in SAM

Want an application-centric perspective of database performance? Consider integrating SAM with SolarWinds Database Performance Analyzer (DPA).

Configure the Application Dependencies feature

To get started detecting and monitoring connections for applications and application processes in SAM, complete these tasks:

1. Enable Application Dependency polling
2. Deploy Orion agents to nodes that host applications
3. Assign application monitors to nodes
4. (Optional) Customize polling settings
5. (Optional) Set up Application Dependencies alerts
Before you begin

Before configuring the Application Dependencies feature, review these details about supported environments, settings, and monitoring limits.

Supported environments

SAM deploys agents and agent plug-ins to flag nodes for Application Dependency polling so machines must support Orion agent requirements. In addition, remote computers hosting applications listening on a specific port must support the following exceptions.

- On the destination node, allow inbound TCP connections for the port, plus an inbound rule from any random remote port.
- On the source node, allow an outbound TCP connection for the port from an Nping application on any random port.

ℹ️ The Applications Dependencies feature does not support Raspbian 8.0.

Note the following details about Connection Quality polling:

- Desktop operating systems such as Windows 8 are not supported.
- Windows 2008 R2 and R2 SP1 with driver installation protection enabled are not supported.
- Do not use on IPv6 or link-local addresses to communicate within the network segment (link) or the broadcast domain to which the host is connected.
- Each dependency must involve two separate nodes.

Required Orion Platform settings

Users with the Administrator role and the following Node Management rights can update Application Dependency polling settings:

- Execute Application Dependency polling.
- Deploy agents to nodes.

💡 Enable the Allow Automatic Agent Updates option on the Agent Settings page. Otherwise, you'll need to manually update agents for individual nodes on the Manage Agents page.

Recommended monitoring limits

For optimal performance, use the Application Dependencies feature to monitor up to 500 nodes. Recommended limits per Orion instance for monitoring dependencies include:

- Application to application: 500
- Application to node: 200
- Node to application: 200
- Average number of TCP connections per single dependency: 5

💡 An Orion agent hosted on a minimally provisioned server can consume high CPU usage during polling. If that happens, try reducing the quantity of monitored elements or increasing server resources.
Enable Application Dependency polling

To enable the Application Dependencies feature:

1. Click Settings > Application Connection Settings.
2. On the Application Connection Settings page, enable the Enable Application Dependency Polling option.
3. (Optional) Enable Connection Quality Polling.
4. Click Save changes.

If Application Dependency polling fails immediately after it is enabled, navigate to the Manage Agents page to check if an agent or agent plug-in is currently being deployed. Wait ten minutes and try polling again.
Deploy Orion agents to nodes that host applications you want to monitor

SAM uses Orion agents and agent plug-ins to flag nodes for Application Dependency polling. An agent must exist on at least one of the two nodes where you want to display application dependencies. If Orion agents do not yet exist on nodes, the Incoming Connections widget on the Node and Application Details pages displays sample data, as shown here:

Click Deploy Agent to Monitor Connections to add Orion agents and agent plug-ins for the Application Dependencies feature to a node.

You can deploy agents to multiple nodes at the same time on the Manage Agents page.

Assign application monitors to nodes

Assign Application Monitors (either out-of-the-box or custom templates) to nodes with the Add New Application Monitors Wizard. See Application Discovery.

You can also assign application monitors to nodes on the Manage Templates page.

Customize polling settings

To fine-tune application dependency polling and thresholds for your environment, use the Application Connection Settings page to:

- Enable Connection Quality polling to display network communication statistics about application dependencies.
- Set polling intervals.
- Indicate when SAM should remove a “down” connection and its dependencies from the Orion.
You can also disable Application Dependency polling for specific nodes, if necessary.

Note the following details about the Application Dependencies feature:

- Application Dependency polling does not monitor the Main Polling Engine (that is, the Orion server). Otherwise, polling would impact performance due to the multitude of connections involved.
- You do not need to configure nodes for polling in advance. SAM deploys required agent plug-ins during Application Dependency polling and (if enabled) Connection Quality polling.

Set up Application Dependencies alerts

You can set critical and warning thresholds between applications for TCP connection packet loss and latency, both at a global and individual connection level.

To configure Application Dependencies alerts at the global level.

1. Click Settings > All Settings > Application Connection Settings.
2. On the Application Connection Settings page, scroll down to Threshold Settings.
3. Select Critical and Warning values for Connection packet loss and Connection latency.
4. Click Save Changes.

To configure Application Dependencies alerts for individual connections, navigate to a specific node, click Edit Node, adjust Alerting Thresholds, and click Submit. You can also navigate to a node's Connection Details page and select Thresholds from the Commands menu.

Use the Application Dependencies feature with Windows Failover Clusters (WFCs)

When used with WFCs, the Application Dependencies feature creates dependencies between connected clients and listening servers on the server side for a cluster Virtual IP (VIP) instead of the active cluster member.
Here is the required configuration for this scenario:

- Only one agentless Orion node has an IP address that matches the virtual IP address of the clustered role.

  Roles were called "Services and Applications" in SQL Server 2012 and earlier.

- Each VIP node has a unique IP address to support the Application Dependencies feature's cluster-matching algorithm.
- A SAM process monitor such as AppInsight for SQL uses an agentless node.
- Application Dependency polling deploys agent plug-ins to agent-monitored cluster member machines so they can be assigned to non-cluster VIP addresses.

The following diagram illustrates an example of MSSQL running on a WFC:

Note these details about this figure:

- An agentless node has the same IP address, 10.140.126.20, as the SQL Cluster VIP role and AppInsight for SQL is assigned to the node.
- The cluster has two members with unique IP addresses monitored as Orion agent nodes.
- The Orion Server is monitored by an Orion Server template.
- The Orion Server instance uses the cluster VIP address, 10.140.126.20, for the SQL Server data store.

SAM can detect the database connection from the Orion Server to the SQL database as a connection between an application (the Orion SQL Server, as monitored by a template) and AppInsight for SQL (MSSQLSERVER) even though the target of the database connection is SQL running on an active cluster member.
Access application dependency data in SAM

As Application Dependency polling and Connection Quality polling (if enabled) occur, the latest data populates these parts of the Orion Web Console:

- The Application Connections widget on Node Details and Application Details pages.
- The Connection Details page that you can access via the Application Connections widget.

Application Connections widget

The Node Details and Application Details pages display the Application Connections widget if Application Dependency polling detects these types of dependencies:

- Application-to-application, including application processes
- Application-to-node
- Node-to-application

![Application Connections screenshot]

The Application Connections widget does not appear automatically on cloned, customized versions of the Node Details and Application Details pages. To add it, click Settings > All Settings > Manage Views. Select the page you want to add it to, click Edit, and select the Application Connections widget. Click Done to save your changes.

To learn more about customizing widgets, see the Orion Platform Administrator Guide.
Sample data appears in the Application Connections widget if a node does not host an application dependency **agent plug-in**, or if polling does not find dependencies. See [Deploy Orion agents to nodes with applications you want to monitor](#).

If Connection Quality polling is enabled, the Application Connections widget displays the following data about TCP connections:

- **Latency**: Network latency (also called response time) is the time required for a packet to travel across a network path from a sender to a receiver. The higher the latency, the greater the impact on application performance as perceived by users. To troubleshoot latency issues, see [Troubleshooting environmental issues with Performance Analysis (PerfStack) dashboards](#).
- **Packet Loss**: A percentage of packets lost with respect to packets sent, usually caused by network congestion. If this value exceeds the Orion general threshold, navigate to the Nodes with High Packet Loss widget to open the custom chart for the node.

Similar to [Spotlight functionality in AppStack](#), you can click connection types and status indicators at the top of the widget to filter data, as shown here:
Interpret colors and symbols in the Application Connections widget

Orion Platform products use icons as a visual language to describe the status of items such as nodes, interfaces, events, or alerts. See Status Indicators for details. Values that exceed Orion Platform thresholds appear bold on a colored background, as shown next.

Gray indicates that either an application is unknown or data is not displayed due to database credential account limitations stored in the SolarWinds Information Services (SWIS) business layer:

<table>
<thead>
<tr>
<th>TO DISPLAY...</th>
<th>YOU MUST HAVE PERMISSION TO VIEW...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node-to-node links</td>
<td>Both parent nodes</td>
</tr>
<tr>
<td>TCP connections and statistics about latency and packet loss</td>
<td>Both parent entities — either nodes or applications</td>
</tr>
<tr>
<td>Applications and application processes linked to a node</td>
<td>The parent node</td>
</tr>
<tr>
<td>Network connection thresholds for a node</td>
<td>The parent node</td>
</tr>
</tbody>
</table>
For a more granular picture of dependencies, view the Connection Details page, as described next.

**Connection Details page**

The Connection Details page shows the entire communication stack from one node to another, which makes it a unique troubleshooting tool.

Data displayed on the Connection Details page includes:

- The application name and detected process names, such as mysqld.exe.
- Monitored TCP metrics, including:
  - CPU: The percentage of CPU utilization.
  - P-MEM: The amount of physical memory used.
  - V-MEM: The amount of virtual memory used.
  - R-IOPS: Read input/output operations per second.
  - W-IOPS: Write input/output operations per second.
- The port used by the application process.
- Latency and packet loss metrics (if Connection Quality polling is enabled).
- Latest Events details.

The Connection Details page also includes a Commands menu that you can use to configure thresholds, initiate Application Dependency polling, or hide events.

To access the Connection Details page:

1. Access a Node Details page or Application Details page in the Orion Web Console.
2. Click the arrow next to a connection displayed in the Application Connections widget.
The Connection Details page opens, as shown in this example:

3. (Optional) Click an option on the Commands menu to:
   - Configure network connection thresholds.
   - Initiate Application Dependency polling immediately.
   - Show or hide events.

   Menu options vary based on account permissions.
Manage polling for application dependencies

SAM uses two types of polling to collect application dependency data:

- **Application Dependency polling**: Discovers and monitors connections between applications and application processes, plus connections between applications, application processes, and nodes.
- **Connection Quality polling**: Collects latency and packet loss for connections between client nodes hosting applications and target.

Options configured on the Application Connection Settings page impact how polling occurs across agent-monitored nodes with connections to applications.
Manage Application Dependency polling

After SAM deploys agents to monitored nodes, Application Dependency polling scans nodes twice in the first ten minutes and then every two hours to:

- Detect connections between applications, application processes, and nodes.
  
  SAM also detects "from" or "to" nodes that rely on server-initiated communications, known as "passive agents" or "agentless" nodes. See the Agent plug-in overview for details.

- Identify nodes as either targets or clients.
- Gather IP address and port data.
- Display data in the Application Connections widget on the Node and Application Details pages.

Note the following details about Application Dependency polling:

- Application Dependency polling occurs every two hours, so short-term connections are less likely to be detected due to the time between polling.
For Windows Failover Clusters (WFCs), SAM can create application dependencies between connected clients and listening servers on the server side for a cluster Virtual IP (VIP) instead of an active cluster member. See Monitor application dependencies for WFCs.

In addition to updating global polling settings on the Application Connection Settings page, you can disable the Application Dependencies feature and re-enable it again later if necessary, as described next.

Disable the Application Dependencies feature

To disable the Application Dependencies feature:

1. Navigate to the Application Connection Settings page.
2. Toggle the Clear the Application Dependency Mapping option.
3. Click Save changes.

SAM automatically removes related agent plug-ins from nodes and stops Application Dependency polling. Connection Quality polling, if enabled, also stops.

If Connection Quality polling was enabled, SAM removes the agent plug-in that delivered the Npcap driver but does not remove the driver. See Disable Connection Quality polling.

Manage Application Dependency polling for a specific node

You can manage Application Dependency polling for specific nodes on the Node Details page.

To disable Application Dependency polling for a specific node:

1. Click Edit Node on the Node Details page.
2. Clear the Application Dependency Polling Enabled check box.

To initiate Application Dependency polling for a single node:

1. Navigate to the Node Details page.
2. Click Poll Now.

To execute Application Dependency polling immediately across poll multiple nodes:

1. Navigate to the Manage Nodes page.
2. Select the nodes.
3. Click More Actions > Poll Now.

Manage Connection Quality polling

In addition to showing how nodes connect to applications and application processes, the Application Connections widget and Connection Details page can display TCP latency and packet loss metrics if Connection Quality polling is enabled, as shown here.
To gather TCP data, SAM deploys agent plug-ins to nodes to track communication traveling from clients that host applications and application processes to target nodes detected by Application Dependency polling.

Note the following details about Connection Quality polling:

- The default interval for Connection Quality polling is five minutes.
- Each dependency must involve two separate nodes.
- See Supported environments for additional Connection Quality polling requirements.
- Connection Quality polling captures TCP communications only. To track non-TCP communications in the Orion Platform, create a Windows Service monitor for the NetTcpPortSharing service.

Enable Connection Quality polling

To enable Connection Quality polling:

1. Navigate to the Application Connection Settings page.
2. Under Connection Quality Settings, toggle the Enable Connection Quality Polling option to On.
3. Click Save changes.
Disable Connection Quality polling

If you disable the Connection Quality Settings option on the Application Connection Settings page, SAM stops gathering latency and packet loss metrics but continues to gather application connection data. The status of connection entities appears as Unknown on the Connection Details page.

When you disable Connection Quality polling, SAM removes agent plug-ins that delivered Npcap drivers but does not remove the actual drivers. If remaining drivers present a security concern for your organization, visit the SolarWinds Success Center and see Removing an Npcap driver after disabling Connection Quality Polling.

Troubleshoot application dependency issues in SAM

If the Application Connections widget of a Node or Application Details page displays sample data with a "Why don’t I see any connections?" message, follow these steps:

1. Review your environment to ensure that:
   - Orion Platform settings are configured appropriately.
   - Application Dependency monitoring does not surpass recommended limits.
   - Polling is enabled on the Application Dependency Settings page.
   - Remote computers that host nodes use supported environments.
2. Make sure Orion agents were deployed to nodes that you want to monitor for dependencies.
3. Assign application monitors to nodes.
4. Review events displayed on the Connection Details page.
5. Explore Application Dependency log files.

If polling fails for Windows Server 2012 nodes, see Agent-related issues.

Review the following topics before contacting SolarWinds Customer Support:

- Troubleshoot agent-related issues for application dependencies in SAM
- Troubleshoot data-related issues for application dependencies in SAM
- Troubleshoot miscellaneous issues for application dependencies
- Locate Application Dependency log files

Troubleshoot agent-related issues for application dependencies in SAM

If you suspect agent issues are interfering with Application Dependencies and/or Connection Quality polling, here are some items to check:

- Review Configure the Application Dependencies feature to make sure that each node being polled:
  - Uses a supported environment.
  - Is monitored by an Orion agent.
  - Has an application monitor assigned to it.
• Navigate to the Manage Agents page to check if an agent or agent plug-in is currently being deployed. Wait five minutes for the next Connection Quality poll and check again.
• Check for issues with agents deployed by SAM to support application dependencies polling. See the Orion Platform Administrator Guide to learn about troubleshooting agents.

An Orion agent hosted on a minimally provisioned server can consume high CPU usage during polling. If that happens, try reducing the quantity of monitored elements or increasing server resources.

To learn about feature availability for agent vs. agentless nodes, see these Success Center articles:

• Comparison of Windows agent versus agentless
• Comparison of Linux agent versus agentless

Connection Quality polling fails for Windows Server 2012 nodes

If polling fails for Windows Server 2012 nodes, restart the nodes and wait for the next Connection Quality poll to occur. If the issue continues, visit the SolarWinds Success Center and see Connection Quality polling fails on Windows Server 2012 nodes.

Polling fails without errors

If application dependency data does not display as expected for a node:

1. Check if SAM deployed agent plug-ins to Orion agents on the node. Click Settings > Manage Agents > Select agent > More Actions > View installed agent plug-ins.
2. Navigate to the Node Details page to ensure that Application Dependency polling was not disabled for the node.

Polling can also fail if:

• A related component was removed or disabled.
• Agent plug-ins are currently being deployed, especially if the Application Dependencies feature or Connection Quality polling was disabled and then enabled again. Wait ten minutes and try again.

"Plug-in update required" notice

By default, the Allow Automatic Agent Updates option is enabled on the Settings > All Settings > Product Specific Settings > Agent Settings page. Application Dependency polling is also enabled by default so that SAM can poll agent-monitored nodes to detect connections between applications, application processes and nodes.

SAM deploys agent plug-ins to agent-monitored nodes when Application Dependency polling detects interaction between an application and/or application process and a node. Although most Orion Platform agents are deployed in advance, application dependency agent plug-ins are deployed immediately if an application-to-node connection is found.

If the Allow Automatic Agent Updates option is disabled on the Agent Settings page in the Orion Web Console, SAM cannot deploy agent plug-ins to server nodes and the status of the agent appears as "Plug-in update required" on the Manage Agents page.

If expected application dependencies do not appear after polling, navigate to the Manage Agents page. If a "Plug-in update required" notice appears for a node, you can either:
- Enable the Allow Automatic Agent Updates option on the Agent Settings page so SAM can deploy plug-ins automatically to all agent-managed nodes.
- Update agents individually on the Manage Agents page.

**Agent Issue warning appears in the Application Connections widget**

A red box in the Application Connections widget indicates that a deployed agent is not functioning properly.

Here are some items to check:

- Review Configure the Application Dependencies feature to check Orion Platform settings, deploy Orion agents, and assign application monitors to nodes, as necessary.
- Navigate to the Manage Agents page to check if the agent is currently being deployed or a plug-in is being installed. For Connection Quality polling, wait five minutes for the next poll and check again.
- Check for issues with agents and/or agent plug-ins. See the Orion Platform Administrator Guide to learn about troubleshooting agents.

**Uninstall Npcap drivers after disabling Connection Quality polling**

When Connection Quality polling is enabled on the Application Dependency Settings page, SAM deploys ADMConnectionQuality plug-ins with Npcap drivers to Windows nodes for the collection of latency and packet loss metrics.

If you disable Connection Quality polling, you can use a template to remove Npcap drivers, if necessary. See Remove Npcap driver after disabling Connection Quality Polling for ADM.

**Nping returns “Unable to start either npcap or npf service” message**

This message appears for nodes running Windows 2007 or if driver installation protection is enabled for Windows 2008 R2 or later. It is related to the Npcap driver deployed via an agent plug-in that supports the Nping tool which SAM uses to gather connection statistics on Windows nodes.

SolarWinds recommends upgrading nodes to Windows 2008 R2 or later. Otherwise, you will be prompted to install Npcap each time polling occurs for Windows nodes.

**Security warnings for Orion server**

If Connection Quality polling is enabled, Orion deploys agents and agent-plugins to Windows nodes connected to clients that host applications and application processes that may trigger warnings in third-party security software. Downloaded items include:

- An Orion agent.
- An agent plug-in that includes an Npcap driver to support Nping.
Troubleshoot data-related issues for application dependencies in SAM

Review this section if unexpected data appears in the Application Connections widget and/or the Connection Details page.

Stale data

The Application Dependencies feature is designed to group polls into batches for efficiency so different data may be polled at different times and the status of nodes, applications, and connections may not seem synchronized.

If you notice outdated data, check polling intervals on the Application Dependency Settings page, as well as intervals defined for individual nodes on the Node Details page.

Node-specific intervals override global polling intervals defined on the Application Settings page.

Unexpected nodes in Application Connections widget

Application Dependency polling and Connection Quality polling check agent-managed nodes to which Application Dependency plug-ins were deployed, but “to” and “from” connections can also be detected with agentless nodes, as described in the following scenarios:

- If only the target node hosts an Application Dependency agent plug-in:
  - Application Dependency polling gathers IP address and port data for the target node.
  - Data related to the client node (application, process name, etc.) and connection statistics is not gathered or displayed.

- If only the client node hosts an Application Dependency agent plug-in:
  - Application Dependency polling gathers IP address and port data for the client node.
  - Only IP address and port data are gathered from the server node.
  - Connection statistics are not gathered.

TCP connection metrics do not update after polling

Application Dependency polling identifies agent-monitored nodes connected to applications and deploys plug-ins to those nodes before Connection Quality polling occurs. There may be a delay in between the two types of polling, plus agents and agent plug-ins need time to deploy. Wait for the next Connection Quality polling cycle to begin.

Date discrepancies on Connection Details page

The Last Polled value on the Connection Details page shows the latest time of Application Dependency polling. If polling intervals were edited for individual nodes, that date may not reflect the date of the last poll across all nodes. For example, if Node1 was polled one hour ago, but the last large-scale Application Dependency poll occurred two hours ago, the Last Poll date reflects the most recent period — one hour.
Applications and application processes lack expected node dependencies

SAM removes application-to-node connections if a parent/child node or parent/child application is removed from the Orion Platform or is no longer monitored by an agent. For dependencies detected by Connection Quality polling, the last TCP connection is removed when the parent dependency is removed.

A TCP connection may also be removed when:

- You remove or disable a related component.
- The LastSeenTime Stamp for the connection is not updated for over eight hours. Consider changing the Remove down connections options on the Application Dependency Settings page. See Customize polling settings.
- You disable Application Dependency polling for a specific node on the Edit Node page.

If applications and application processes do not have expected dependencies with nodes on the Application Connections widget, confirm that SAM detected communication between nodes by checking the inventory log:

C:\ProgramData\SolarWinds\Logs\ADM\{NodeID}\{NodeIP}.log

Check the data processing logs for monitoring applications on each node:

C:\ProgramData\DataProcessingLogs\NodeId-{NodeID}\*.log

"Unknown" connection status

If you initiate polling on the Connection Details page, Application Dependency polling starts but an "Unknown" connection status may appear until the next Connection Quality poll occurs (every five minutes, by default).

"Loopback" connection status

TCP Loopback connections established internally on a node may appear on the Application Connections widget. A Loopback connection status indicates an internal connection on the node (localhost connection). If the destination and source nodes are the same, Connection Quality polling ignores the connection.

Hyperlinks to Orion components (process, port) missing

The application template does not contain any component monitoring that the system can refer to for the given process or port.

Troubleshoot miscellaneous issues for application dependencies

This section contains additional troubleshooting tips for the Application Dependencies feature.

Check Application Dependencies services

The following table describes where to check to ensure Application Dependencies services are functioning. See Application Dependencies log files for log file locations.
<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>GROUP STATUS</th>
<th>EXPECTED RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check nodes that will be polled</td>
<td>Collector Service log</td>
<td>See polling plans for nodes with active applications.</td>
</tr>
<tr>
<td>Check planned jobs</td>
<td>Polling Plan log</td>
<td>See jobs created for node active applications.</td>
</tr>
<tr>
<td>Check that a publish-subscribe pattern (pubsub) succeeded</td>
<td>Business Service log</td>
<td>See the cache that was created, along with inventory messages received for node with active applications.</td>
</tr>
</tbody>
</table>

**Handle HTTP listeners port sharing in Windows**

A networking Windows OS subsystem is implemented as a kernel-mode device driver called the HTTP protocol stack — also called HTTP.sys. This driver listens for HTTP requests from the network, passes requests to IIS or other applications for processing, and returns processed responses to client applications.

Detecting HTTP connections is not the main goal of the Application Dependencies feature. Typically, communication between clients and HTTP servers is not permanent, and Application Dependency polling occurs relatively infrequently so it will not detect short HTTP connections.

HTTP connections may appear in the Application Connections widget in the following circumstances:

- Random capture of common HTTP(s) occurred.
- Orion Server communication was captured during Orion server polling or via user action in the Orion Web Console that initiated a poll.
- HTTP communication persisted for a long time, perhaps due to tunneling of another kind of TCP communication over HTTP(s).
Monitor Windows Communication Foundation (WCF) communication

The Windows Communication Foundation (WCF) application uses the Net.TCP Port Sharing service to share ports across multiple processes to reduce the number of ports that need to be open on a firewall. That service listens on port 17777, which is the same port several Orion Platform services listen to so they can forward communication to the Orion Platform through an internal, non-TCP communication channel. The ability to track WCF communication with the Application Dependencies feature is planned for a future release.

Locate Application Dependency log files

The Application Dependency polling job searches for application and application process connections from nodes monitored by agents to which application monitors were assigned. The job also tracks the creation of application dependencies, as indicated by agent plug-ins that SAM deploys during polling.

SAM stores Application Dependency polling logs on the main Orion server and also on target machines, as described here:

- Default Orion server locations:
  - C:\ProgramData\SolarWinds\Logs\SAM.ADM\ADMPollingJob_{PID}.log
  - C:\ProgramData\SolarWinds\Logs\ADM\ADMPollingJob_{PID}.log
- Default target machine locations:
  - C:\ProgramData\SolarWinds\Logs\Agent\SolarWinds.ADM.AgentPlug-in.exe.1360.0001.log.txt

Connection Quality polling logs are stored in the following files on the main Orion server:

- C:\ProgramData\SolarWinds\Logs\ADM\ADMPollingJob_{PID}.log
- C:\ProgramData\SolarWinds\Logs\Agent\SolarWinds.ADM.AgentPlug-in.exe.1360.0001.log.txt

To free up disk space, move logs to the C:\Program Files (x86)\SolarWinds\Logs\SAM.ADM\ folder where files are automatically deleted after five days by default.

Monitor hardware health in SAM

You can use SAM to monitor the health of server hardware components including temperature, fan speed, power supply, CPU, memory, disk space, and more. SAM provides instant visibility of the status (up, critical or warning), allows you to set baseline values, and alerts you if levels fall below set thresholds.

Monitoring is achieved by polling via SNMP, WMI, or ICMP (for some Cisco devices), depending upon the node.

Hardware health monitoring does not count against your licenses.

For the following devices, SAM monitors hardware by polling nodes and utilizing agent software provided by hardware manufacturers:
- Dell servers with OpenManage Server Administrator Managed Node 7.2 or later
- HP servers with HP System Insight Manager 6.2 or higher
- HPE ProLiant Gen10 Servers with SIM 8.0 or later *(via SNMP protocol only)*
- IBM server with IBM Director (Common Agent, 6.3 or higher)

SAM can also monitor hardware health for Cisco UCS blade servers without additional software. See [Monitor Cisco UCS devices in the Orion Platform](https://thwack.solarwinds.com/community/threads/cisco-ucs-deployment.823224/).

Note the following details about monitoring hardware health:

- When you add nodes using **Discovery**, hardware health sensors are enabled for devices that support hardware health monitoring automatically.
- You can customize the hardware health monitoring interval for each device from the corresponding device snapshot page. To change the hardware monitoring interval for a particular device, go to the Hardware tab in the device snapshot page and edit the value for the Interval option. See the [Orion Platform Administrator Guide](https://thwack.solarwinds.com/community/threads/orion-platform-administrator-guide.828639/) for details.
- Certificate errors found during polling are ignored by default, but you can change that setting.
- For tips on monitoring HPE ProLiant servers, see [this THWACK post](https://thwack.solarwinds.com/community/threads/cisco-ucs-deployment.823224/).

**Configure hardware health monitoring**

To get started monitoring hardware health in SAM, you'll need to:

1. Review [hardware health monitoring requirements](https://thwack.solarwinds.com/community/threads/orion-platform-administrator-guide.828639/).
2. Download, install, and configure hardware monitoring agent software from third-party vendors. Click [here](https://thwack.solarwinds.com/community/threads/cisco-ucs-deployment.823224/) for details.

   - Extra software is not required to monitor hardware health for Cisco UCS devices.

3. Run **Discovery** to detect third-party agent software and enable hardware health monitoring automatically across multiple nodes.

   - You can also enable hardware health monitoring for individual nodes.

When SAM enables hardware health monitoring during Discovery, it also enables [Asset Inventory data collection](https://thwack.solarwinds.com/community/threads/cisco-ucs-deployment.823224/) to track each node's hardware and software. Polling jobs begin immediately, and are later scheduled to occur at random times over a 24-hour period. If necessary, you can adjust the default interval for Asset Inventory polling.

**Hardware health monitoring requirements in SAM**

The following systems support hardware health monitoring in SAM:

- Cisco UCS blade servers (See [Monitor Cisco UCS devices in the Orion Platform](https://thwack.solarwinds.com/community/threads/cisco-ucs-deployment.823224/) for details.)
- Dell servers with OpenManage Server Administrator Managed Node 7.2 or later
- HP servers with HP System Insight Manager 6.2 or higher
- HPE ProLiant Gen10 Servers with SIM 8.0 or later *(via SNMP protocol only)*
- IBM server with IBM Director (Common Agent, 6.3 or higher)
OS and protocol requirements

Hardware health monitoring supports the following operating systems and protocols:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>SNMP, WMI</td>
</tr>
<tr>
<td>Linux</td>
<td>SNMP</td>
</tr>
<tr>
<td>AIX v7 and higher</td>
<td>SNMP</td>
</tr>
<tr>
<td>VMware ESX/ESXi v4.x and v5.x</td>
<td>CIM, VMware API</td>
</tr>
<tr>
<td>VMware API for ESX/ESXi hosts polled via vCenter</td>
<td>HTTPS</td>
</tr>
</tbody>
</table>

Use SNMP for VMware nodes not polled via CIM or VMware API.

Download third-party software

SAM uses standard protocols such as SNMP and WMI to monitor hardware health data, but not all information is available natively from an OS without installing the hardware vendor’s required agent software. To download agent software for supported devices, visit the SolarWinds Success Center and see Third-party software required to monitor hardware health and collect asset inventory information in SAM.

Enable hardware health monitoring for individual nodes in SAM

When you add nodes during Discovery, hardware health sensors are automatically enabled for devices that support hardware health monitoring and polling begins. There are two ways to enable hardware health monitoring for individual nodes; through the Add Node wizard, and through the Node Details group of the Node Details page.

Add Node Wizard

From the Add Node wizard, the option to display Hardware Health of Servers is available after a node has been defined. Check this box to enable hardware health monitoring.
Manually add hardware monitoring

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click My Dashboards > Home.
2. In the All Nodes widget, click the node you want to monitor.
3. Click List Resources.
4. Select Hardware Health Sensors, and click Submit.

To disable hardware monitoring, navigate back to this screen and uncheck Hardware Health of Servers, then click Submit.

For information on troubleshooting hardware health polling, see Troubleshooting hardware health.

Configure how SAM handles certificate errors during Hardware Health polling

By default, SAM ignores certificate errors during Hardware Health polling. You can update that setting so users are prompted to verify untrusted connections before proceeding.

To configure SAM to prompt users to verify connections with invalid certificates during Hardware Health polling:

1. Log into the Orion Web Console as an administrator.
2. Navigate to the default folder, C:\Program Files (x86)\SolarWinds\Orion\HardwareHealth\SolarWinds.HardwareHealth.Collector.dll.config.
4. Scroll to <appSettings> and change "false" to "true" in the HardwareHealthVerifyServerCertificate line, as shown here:

   <add key="HardwareHealthVerifyServerCertificate" value="true" />

5. Save your changes.
6. Use the Orion Service Manager to restart the SolarWinds Collector Service.
Monitor with AppInsight applications

SolarWinds SAM's AppInsight applications provide a level of detail and expert knowledge far beyond what a simple template can provide. You can monitor virtually every aspect of key business tools, including Microsoft Active Directory, Exchange, IIS, and SQL, by getting relevant performance information from the server level, as well as drilling down into the data store layer for performance data.

AppInsight applications are considered templates until they're assigned to nodes during Discovery or via the Manage Application Monitor Template page. After being applied to nodes, AppInsight templates become monitored as applications. To learn more, see The template and application relationship.

Currently, SAM offers these AppInsight applications:

- AppInsight for Active Directory
- AppInsight for Exchange
- AppInsight for IIS
- AppInsight for SQL

To learn how AppInsight applications consume licenses, see the SAM licensing model.

To learn more about AppInsight applications, check out these videos:

- AppInsight for Exchange: Two Geeks and a Goddess
- How To: AppInsight for IIS
- Introducing AppInsight for SQL Server
- Monitoring Microsoft IIS
- Q & A: AppInsight for Exchange
- SQL Server Performance: AppInsight for SQL
The template and application relationship

The following illustration explains how templates and applications relate and is true for all templates, including ApplInsight applications.

If you change a configuration setting or component monitor at the template level, the applications based on that template are affected. Conversely, if you change a setting at the application level, only that application is affected. This inheritance relationship is beneficial if you need to make a many changes quickly.

For example, rather than change one item on 100 applications that are based on a single template (which requires 100 changes), you can change the one item on the template, then that one change trickles down to all applications based on the template.
AppInsight for Active Directory

Key aspects of monitoring Active Directory involve keeping a close watch on the application and service availability, and ensuring various performance metrics are checked against accepted thresholds.

AppInsight for Active Directory monitors physical and virtual Active Directory environments to identify issues about domain controllers, replication, and more. You can use it to track many key aspects of Active Directory by getting relevant performance data from the server level. You can also drill down into the data store layer for performance data.

Here are some ways you can use the status and metrics provided by AppInsight for Active Directory:

- **File replication service**: Identify failures on replications or issues with the network leading to slow replication rates between websites.

- **Directory services**: Watch critical directory services to ensure your email and phone contacts are always in-sync.

- **Service outages**: Monitor domain controllers continuously to prevent service outages. Diagnose domain controller performance issues by tracking CPU usage, connected users, failed logins, account lockouts, and more. Discover domain controllers on unmonitored nodes within your environment.

- **Dependencies**: Troubleshoot Active Directory dependencies with widgets that show in-depth details about issues impacting performance.
To quickly set up and use AppInsight for Active Directory:

- **AppInsight for Active Directory licensing.**
- **Configure AppInsight for Active Directory:**
  - Add a domain controller as a monitored node that's already running Active Directory Domain Services.
  - Assign the AppInsight for Active Directory template to the monitored node.

To further refine AppInsight for Active Directory monitoring, you can:

- **Customize widgets** that appear in views.
- Adjust component monitors in the AppInsight for Active Directory template.
- Configure **alerts** to receive notifications about domain controller usage, issues, and thresholds.

After you assign AppInsight for Active Directory to a node, it's considered to be an application on the node and reports data to SAM through a predefined set of component monitors, including:

- **Windows Event Log Monitor**, that uses WMI and RPC communication to scan Windows Event Logs for server events.
- **Performance Counter Monitor**, that uses RPC methods or WMI to collect Windows Performance Counter data.

For a list of component monitors included in AppInsight for Active Directory, see the [SAM Template Reference](#).

### AppInsight for Active Directory licensing

Per the [SAM licensing model](#), when using AppInsight for Active Directory, 50 component monitors count against your licensed number of component monitors per server.

AppInsight applications provide tremendous value within SAM while consuming a fixed number component monitor licenses. However, they cannot be partially unlicensed because the way they collect data differs significantly from traditional application templates.

For example, if you have an active license for 1,500 component monitors and use AppInsight for Active Directory to monitor 5 domain controllers, 250 component monitors count against your total license. The amount of licenses used is strictly per monitored AppInsight for Active Directory application. The number of Active Directory instances you have on domain controllers affects the number of elements.

\[
\text{50 component monitors} \times \text{5 domain controllers} = \text{250 component monitors used.}
\]

This leaves you with 1,000 component monitors available for use elsewhere.

\[
\text{1,500 component monitors} - \text{250 component monitors used for AppInsight for Active Directory} = \text{1,250 component monitors remaining.}
\]

If you choose not to use AppInsight for Active Directory, you are not penalized any number of component monitors.
AppInsight for Active Directory requirements and permissions

**Supported versions**

Domain controllers should already be running Active Directory Domain Services (AD DS) on:

- Windows Server 2012 R2, or
- Windows Server 2016

**Ports**

- WMI technology is based on DCOM / Remote Procedure Call (DCOM/RPC) communication that allocates the ports within a dynamic port range, typically between 1025 and 65536. Enable the Inbound Rules in the WMI group and create firewall exceptions to allow TCP/UDP traffic on ports 1024 — 65535 so monitored objects that use WMI can be mapped.
  - WMI TCP ports 1025 — 5000
  - TCP ports 49152 — 65535
- For LDAP, use the default port for TCP and UDP, 389.
- For LDAP over SSL (LDAPS), use port 636.

**Encryption**

Active Directory does not support encryption so the encryption method for connecting to domain controllers is set to None, by default. To use SSL or StartTLS, you can add an LDAP certificate to the server manually.

**Authentication**

By default, authentication is set to Negotiate so SAM can use Kerberos or NT LAN Manager (NTLM).

**Permissions**

- Local administrator permissions are required to add AppInsight to nodes, but are not needed for monitoring after configuration is complete.
- Application credentials must be from the domain of the monitored node with proper read/write permission for Active Directory services.
- Domain credentials used for monitoring must have read access to monitored Active Directory instances.

> SolarWinds recommends using Active Directory accounts with limited permissions (for example, read-only administrators) for AppInsight for Active Directory monitoring.
Configure AppInsight for Active Directory

After reviewing requirements and permissions, make sure you have key information and an Active Directory user account so you can add AppInsight for Active Directory to domain controller nodes being monitored in the Orion Platform. Information you'll need includes:

- Either the IP address or fully-qualified domain name (FQDN) of the domain controllers.
  
  To access FQDN details, open a Windows command prompt on a computer on the correct network and type nslookup.

- The port number, encryption method, and authentication method for each server.
- The domain credentials for an account that SAM can use to log in to Active Directory. The account does not need elevated privileges.

SolarWinds recommends using Active Directory accounts with limited permissions (for example, read-only administrators) to monitor AppInsight for Active Directory.

Add AppInsight for Active Directory to domain controller nodes

To configure and monitor domain controllers in SAM, add AppInsight for Active Directory to nodes that are already running Active Directory Domain Services. You can add AppInsight to nodes through Discovery, or manually via the Manage Templates or Node Details pages.

To ensure that node status appears in AppInsight for Active Directory widgets, SolarWinds recommends configuring nodes to support both WMI and ICMP polling so that AppInsight widgets can display node status and names properly via WMI. ICMP-only nodes cannot supply DNS or SysName values required to compute replications for destination domain controller FQDN names. See this article in the SolarWinds Success Center for details.

After adding AppInsight for Active Directory to nodes, you can edit the AppInsight for Active Directory template to enable total counters (for example, Total User Accounts and Total Computer Accounts). Those component monitors are initially disabled by design, to circumvent performance issues in environments with great quantities of user and computer accounts.

Add nodes with the Discovery Wizard

Before you begin, enable WMI on domain controllers so they can be detected by the Discovery Wizard.

Use the Discovery Wizard (also called Network Sonar Discovery) to add a new node and select AppInsight for Active Directory for monitoring. Credentials are inherited from the node automatically.

1. Click Settings > All Settings > Add Node.
2. When the Add Node wizard appears, enter information on the Define Node tab and click Next.
3. On the Choose Resources tab, select AppInsight Applications > Active Directory.

![List Resources - ENG-AUS-SAM](image)

4. Click Next and follow onscreen instructions to complete the wizard.

5. To confirm the node was added:
   a. Click My Dashboards > Applications > Active Directory.
   b. Navigate to the All Application widget, expand the tree, and click the Active Directory application.

**Add AppInsight for Active Directory to nodes via the Manage Templates page**

You can add AppInsight for Active Directory monitoring to a domain controller already being monitored as a node via the Manage Templates page.

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. On the Manage Templates page, switch to the Application Monitor Templates tab.
3. In the Template Name column, select the AppInsight for Active Directory check box.
4. Click Assign to Node.
5. Complete fields on the Set up AppInsight for Active Directory page and click Assign Application Monitor.

   > The default port to connect to domain controller LDAP services is 389.

**Add AppInsight for Active Directory to nodes via the Node Details page**

Follow these steps to add AppInsight for Active Directory to a domain controller already monitored as a WMI node in SAM.

1. Click My Dashboards > Home > Summary.
2. Expand and select the monitored domain controller node in the All Nodes - Tree View widget.
3. When the Node Details page appears, click List Resources in the Management widget.
   The list may take a few minutes to generate.
4. Select Microsoft Active Directory to enable AppInsight for Active Directory data collection. When done, click Submit.

5. Click My Dashboards > Applications > SAM Summary.

6. Locate the All Applications widget, and click the Microsoft Active Directory application on the specific node you modified.

7. When prompted, enter your Active Directory credentials and select the port used to communicate with the domain.

8. Click Test to verify the credentials and configured permissions.

9. Click Assign Credential to save the configuration.

After adding AppInsight for Active Directory to nodes, you can edit the AppInsight for Active Directory template to enable total counters, if desired.

Edit the AppInsight for Active Directory template

You may need to log in with an administrator account to edit templates.

The AppInsight for Active Directory template includes numerous component monitors to provide data about domain controllers and Active Directory services. If you modify these settings and configurations in the template, the details in AppInsight application monitors already assigned to servers update to match.

The primary reason to edit most SAM templates is to configure polling frequency, polling method, and thresholds for warning and critical states for monitored metrics before assigning templates to nodes. Like the other AppInsight templates, the AppInsight for Active Directory template includes several component monitors with default settings that cannot be modified due to dependencies. Also, you cannot add component monitors to this template.
When working with component monitors, note that AppInsight for Active Directory uses domain controller IP addresses instead of domain names for polling; LDAP components do not include the $DomainName parameter in configuration fields. This use of IP address enables different applications to get data from all monitored domain controllers in a single domain.

See the SAM Template Reference for a list of component monitors included in this AppInsight template. You can also select the template on the Manage Application Monitor Templates page and click Edit to display component monitor details, and then make your changes.

Use the "Enable total counters" feature

During the initial configuration of AppInsight for Active Directory, the following component monitors are disabled, by default, to avoid performance issues for domain controllers in environments with large quantities of users and computers:

- Total User Accounts
- Total Disabled User Accounts
- Total Computer Accounts
- Total Inactive Users
- Total Inactive Computers
- Total Expired Password User Accounts

To enable these counters for the main template or for individual AppInsight for Active Directory application monitors, select the check box for the "Enable total counters" option and wait ten minutes for polling to occur.

If you decide to disable the "Enable total counters" feature later, be sure to restart the Orion Collector service to clear its cache. Otherwise, AppInsight for Active Directory applications display as Down and warnings appear in SAM logs. To restart services, use the Orion Service Manager, accessible by clicking Settings > All Settings > Orion Service Manager.

AppInsight for Active Directory alerts

AppInsight for Active Directory includes the predefined alert described in the following table:

<table>
<thead>
<tr>
<th>ALERT</th>
<th>TRIGGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication from [SourceDomainControllerName] to [DestinationDomainControllerName] failed on [NamingContextName] - last success time [LastSuccessTime] consecutive number of failures [consecutiveNumberOfFailures] last attempt time [LastAttemptTime]&quot;</td>
<td>Replication of a domain controller fails</td>
</tr>
</tbody>
</table>

AppInsight for Active Directory alerts appear in the Active Alerts widget on the following pages:

- Active Directory Summary page
- Application Details page
- Domain Details page
Trigger conditions related to AppInsight for Active Directory that you can use in alerts include:

- AppInsight for Active Directory: Application
- AppInsight for Active Directory: Naming Context
- AppInsight for Active Directory: Site

To learn more about using alerts in the Orion Platform, see

- Managing Alerts (video)
- Orion Platform Administrator Guide

Troubleshoot AppInsight for Active Directory

AppInsight for Active Directory monitors physical and virtual Active Directory environments to identify issues about domain controllers, replication, and more. This topic describes about issues you may encounter when using AppInsight for Active Directory in SAM.

Be sure to review AppInsight for Active Directory requirements and permissions.

Issue: AppInsight for Active Directory widgets do not display polled data.

After configuring AppInsight for Active Directory, you may not see active data in widgets or receive alerts immediately because polling may occur at different intervals, ranging from minutes to hours. After 24 hours, widgets and alerts should start reporting data.

The default polling interval for AppInsight for Active Directory is 10 minutes.

Issue: Node status does not appear in AppInsight for Active Directory widgets, and Active Directory widgets display IP addresses instead of node names.

To ensure that node status appears in AppInsight for Active Directory widgets and that widget show proper node names, SolarWinds recommends configuring nodes to support both WMI and ICMP polling so that AppInsight widgets can display node status and names properly via WMI. ICMP-only nodes cannot supply DNS or SysName values required to compute replications for destination domain controller FQDN names. See this article in the SolarWinds Success Center for details.

Issue: Domain controllers display "Unknown" for several custom performance counters.

Check the server to make sure the Windows NT Directory Service (NTDS) category of performance counters is present. If the base set of performance counters libraries was corrupted, you may need to rebuild it.

Issue: Components display Unknown status after initial poll.

Generic application monitors like Windows Event Log Monitor and Performance Counter Monitor may have an Unknown status after the first round of polling. Wait 10 minutes for the next successful poll.

If the status of a domain controller remains Unknown after polling, make sure that a node exists for the domain controller and that AppInsight for Active Directory is assigned to the node. See Configure AppInsight for Active Directory.
Issue: After enabling Total counters in the AppInsight for Active Directory template, the AppInsight for Active Directory application appears Down and warnings occur in SAM logs.

Restart the Orion Collector service to clear its cache. See Enable total counters for details.

Issue: Domain Controllers display Unknown status in AppInsight for Active Directory widgets.

Make sure the node is being monitored and that the AppInsight for Active Directory application template is assigned to the node. See Configure AppInsight for Active Directory.

Issue: AppInsight for Active Directory applications appear Down and warnings occur in logs after enabling and later disabling the Total Counters feature in the template.

You can edit the AppInsight for Active Directory template to turn on the Total Counters feature. If you decide to disable the feature again later, AppInsight for Active Directory applications may incorrectly display as Down and warnings may appear in SAM logs. Enabling and disabling components for AppInsight applications can place the Orion Collector service in an unsynchronized state.

Restart the Orion Collector service to clear its cache. Then restart services with the Orion Service Manager, accessible by clicking Settings > All Settings > Orion Service Manager.

Issue: The Replication widget shows only a few domain controllers.

The Replication widget displays domain controllers (shown on the right, below) that replicate with the monitored domain controller (shown on the left) in the same domain, by design.
AppInsight for Exchange

Designed exclusively for the Exchange Mailbox role, AppInsight for Exchange provides visibility into storage issues, mail queues, mailbox database status and growth, events, and critical processes and services. With AppInsight for Exchange you can get a single view of all relevant current and historical Exchange performance metrics.

You can add AppInsight for Exchange automatically through Discovery or add it to nodes manually via the Node Details page. After it is applied to a node, AppInsight for Exchange is considered an application and reports data to SAM through a set of component monitors.

AppInsight uses the Exchange credentials you provide to directly access the servers, complete configuration, and collect data during polling. Before adding AppInsight, review requirements and account permissions.

Advanced manual configuration of AppInsight for Exchange provides additional configuration and usage options. Manual configuration is recommended for experienced Exchange administrators.

To further refine AppInsight for Exchange monitoring:

- Customize widgets that appear in views
- Add and remove component monitors in the AppInsight for Exchange template
- Enable, modify, and use alerts for receiving notifications of Exchange server usage, issues, and thresholds

To optimize Exchange server performance, see 5 Tips to Optimize Exchange Server for Improved Performance. You can also download the free Exchange Monitor tool to check the status and performance of Exchange servers.

Navigating AppInsight for Exchange

AppInsight for Exchange can be found in the All Applications widget on the Application Summary view.

You may need to log in with an administrator account to perform this action.

To view the AppInsight for Exchange Details page:

1. In the Orion Web Console, locate the All Applications widget by clicking My Dashboards > Applications > SAM Summary.
2. Expand the AppInsight for Exchange tree and click [+].
3. Expand the node tree, click [+], and then click the application.

The AppInsight for Exchange Details view provides a customizable view of widgets for monitoring your Exchange servers and services. Widgets on the view offer access to more Exchange details pages including performance counter, database, and mailbox details.
- Click any performance counter in a widget to view the Exchange Performance Counter Details page.

![Critical Processes and Services](image)

- Click any database within a Database widget to view the AppInsight for Exchange Database Details page.

![Mailbox Database Status](image)
Click any user name within a Mailbox widget to view the AppInsight for Exchange Mailbox Details page.

**Users By Mailbox Size**

<table>
<thead>
<tr>
<th>USER NAME</th>
<th>MAILBOX SIZE</th>
<th>% QUOTA USED</th>
<th>ATTACHMENT SIZE</th>
<th>ATTACHMENT COUNT</th>
<th>LAST ACCESSED</th>
<th>DATABASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>labadmin</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>null</td>
<td>System</td>
<td>Mailbox Database</td>
</tr>
<tr>
<td>Discovery Search Mailbox</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>null</td>
<td>System</td>
<td>Mailbox Database</td>
</tr>
<tr>
<td>Labuser</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>null</td>
<td>System</td>
<td>Mailbox Database</td>
</tr>
<tr>
<td>Incoming Mail</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>null</td>
<td>System</td>
<td>Mailbox Database</td>
</tr>
</tbody>
</table>

**Recommendations for AppInsight for Exchange**

When using and configuring AppInsight for Exchange, SolarWinds recommend the following best practices:

- After configuring AppInsight for Exchange, you may not see active data in widgets or receive alerts immediately because polling may occur at different intervals, ranging from minutes to hours. After 24 hours, widgets and alerts should start reporting data.
- When first testing alerts, only assign the alerts to your own or other tester email addresses.
- Watch and monitor the alerts for two weeks to generate stable baselines. You can configure to use the new baselines for refining monitoring and alert actions for the usage and performance in your specific environment. Your environment's baseline and performance expectations may vary, as compared to the default thresholds.
- Create custom views with different AppInsight for Exchange widgets for user groups in your organization.

**AppInsight For Exchange licensing**

AppInsight applications provide tremendous value within SAM while consuming a fixed number component monitor licenses. However, they cannot be partially unlicensed because the way they collect data differs significantly from traditional application templates. You can disable some components within AppInsight applications but they will not reclaim component monitor licenses because AppInsight applications typically monitor far greater than 50 components.

Per the [SAM licensing model](#), when using AppInsight for Exchange, 50 component monitors count against your licensed number of component monitors per Exchange Server.
For example, if you have an active license for 1,500 component monitors and use AppInsight for Exchange to monitor 88 mailboxes over ten Exchange servers, 500 component monitors count against your total license. The amount of licenses used is strictly per Exchange server instance. The number of mailboxes you have on these servers affects the number of elements. For details on scalability, see the [Scalability Engine Guidelines for SolarWinds Orion Products](#).

\[(50 \text{ component monitors } \times 10 \text{ Exchange Servers} = 500 \text{ component monitors used.}\]

This leaves you with 1,000 component monitors available for use elsewhere.

\[(1,500 \text{ component monitors } - 500 \text{ component monitors used for AppInsight for Exchange} = 1,000 \text{ component monitors remaining}].\]

Also note that if you choose not to use this application, you are not penalized any number of component monitors.

The example below illustrates a situation where you have 40 available component monitors available, but surpassed your allowed number of 300 monitors by 60. AppInsight applications are licensed as 50 monitors per application and cannot be partially licensed, as is the case with typical applications, because the way they collect data differs significantly from traditional application templates. You can disable some components within AppInsight applications but they will not reclaim component monitor licenses because AppInsight applications typically monitor far greater than 50 components.

<table>
<thead>
<tr>
<th>SAM</th>
<th>Server &amp; Application Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Name</td>
<td>Version</td>
</tr>
<tr>
<td>Service Pack</td>
<td>License</td>
</tr>
<tr>
<td>Allowed Number of Component Monitors</td>
<td>Total Number of Component Monitors</td>
</tr>
<tr>
<td>Licensed Component Monitors</td>
<td>Unlicensed Component Monitors</td>
</tr>
<tr>
<td>Available Component Monitors</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>360</td>
</tr>
<tr>
<td>260</td>
<td>100</td>
</tr>
<tr>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

**AppInsight for Exchange requirements and permissions**

AppInsight for Exchange works only with the Mailbox Role, which is used for data storage. All other Exchange servers running different roles should use the Exchange application templates included with SAM if you intend to monitor them. Data is collected at the same default five-minute polling interval as traditional application templates.

**Supported Microsoft Exchange Server versions**

- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
Microsoft Exchange Server 2016

Exchange hybrid versions are not supported.

To gain insight into metrics, services and Database Availability Group (DAG) status for Microsoft Exchange Server 2013 and 2016, see Monitor your Microsoft Exchange Server with a free tool, Exchange Monitor in the SolarWinds online IT community, THWACK.

AppInsight for Exchange permissions

The following permissions are required to modify Exchange and WinRM settings on the server, as well as poll performance counters:

- Local administrator permissions are needed for automatic configuration, but are not needed for monitoring after configuration is complete.
- To provide organization-wide capability, the service account (Domain User) must be a member of the View-Only Organization Management group. Membership to this group gives the user object read-only access to the entire Exchange environment, without providing any domain or local access on the Exchange server. It also prevents unauthorized access to the account.
- To gather information, the user object must be assigned the Mailbox Search management role within Exchange. The account must be a member of the Local Administrators group.
  Note: Users without this role can access Exchange, but the additional level of permission is required to modify Exchange and WinRM settings on the server, or poll performance counters.
- For Mailbox statistics, Hub Transport Servers need to be accessed via RPC.

PowerShell requirements

- PowerShell 2.0 or later must be installed. See Use PowerShell in SAM.
- Permissions must be granted for PowerShell to be accessed. See Set PowerShell permissions for Exchange.

Common configuration issues

- To add local administrative privileges to an Active Directory account, see Manually configure an Exchange server.
- To determine Exchange URL settings, see Find Exchange URL settings.
- To install PowerShell on the Exchange server, see Manually configure an Exchange server.

Configure AppInsight for Exchange on nodes

To configure and monitor Exchange servers in SAM, add AppInsight for Exchange to an existing node manually or to a new node through the Discovery Wizard.

Review the Exchange requirements and permissions first:

- Local administrator permissions are required for automatic configuration. You do not need these credentials for monitoring Exchange.
To provide organization-wide capability, the service account (Domain User) must be a member of the View-Only Organization Management group to give the user object read-only access to the entire Exchange environment, without providing any domain or local access on the Exchange server. It also prevents abuse by an unauthorized user who might modify Exchange environment configuration, creating or deleting users, and so on.

To gather information, the user object must be assigned the Mailbox Search management role within Exchange. To configure the account, it must be a member of the Local Administrators group.

Each target server requires IIS, as well as PowerShell 2.0 or later so SAM can run custom PowerShell scripts to configure target servers for Exchange monitoring.

To learn more about PowerShell requirements and security considerations, see Use PowerShell in SAM.

SolarWinds recommends adding AppInsight for Exchange on WMI-managed nodes. If you convert an SNMP-managed node to WMI, interface statistics are lost. See Promote a Node to WMI Monitoring.

Note the following details about AppInsight for Exchange nodes:

- SAM does not support multiple instances of Exchange on the same server.
- Nodes that are not added via WMI do not display in the List Resources dialog box.
- Monitoring an Exchange Database Availability Group (DAG) by way of a Virtual IP address (VIP) is not supported. AppInsight for Exchange should only be applied to the physical IP address of each mailbox server in the DAG individually.

Add through the Discovery Wizard

Use the Discovery Wizard to add a new node and select AppInsight for Exchange for monitoring.

To learn more about the Discover Wizard, see the Orion Platform Administrator Guide.

1. In the Orion Web Console, click Settings > All Settings > Add Node.
2. Complete the information on the Define Node step, and then click Next.
3. On the Choose Resources step in the Add Node Wizard, select AppInsight for AD.
4. Click Next and complete the wizard as instructed.

When you complete the wizard, you need to enter credentials for Exchange through another page.

5. Click My Dashboards > Applications > SAM Summary.
6. In the All Applications widget, expand the listed servers to locate the new node, and then click the added Active Directory application and node.
7. Enter Active Directory credentials when prompted, and click Configure Server.

**Add to an existing node**

You can add AppInsight for Exchange monitoring to an Exchange server already monitored as a node in SAM.

1. In the Orion Web Console, click My Dashboards > Home > Summary.
2. Expand and select the monitored Exchanger server node in the All Nodes - Tree View widget. The details page for the node displays.
3. From the Management widget, click List Resources. The list may take a few minutes to generate.
4. Select Microsoft Exchange to enable AppInsight for Exchange data collection. When done, click Submit.

5. Click My Dashboards > Applications > SAM Summary.

6. Locate the All Applications widget, and click the Microsoft Exchange application on the specific node you modified.

7. Enter Exchange credentials when prompted, and click Configure Server.
When you click Configure Server, SAM runs custom PowerShell scripts to:

- Add the Mailbox Search Role to the Exchange server with the credentials provided.
- Enable the WinRM service to provide the Orion server with remote access to target servers.

The custom PowerShell scripts make the following configuration changes:

- Enable Windows Authentication for PowerShell’s web site
- WinRM and Windows Authentication configurations are performed remotely from SAM

The process initiated when you click Configure Server is sometimes called "Zero Config" (for example, in the SolarWinds online IT community, THWACK).

Edit the AppInsight for Exchange template

The AppInsight for Exchange template includes numerous component monitors to provide data about Exchange servers in your environment. If you modify these settings and configurations in the template, the details in AppInsight application monitors already assigned to servers update to match.
The primary reason to edit most SAM templates is to set general configurations such as polling frequency, polling method, and thresholds for warning and critical states for monitored metrics before assigning templates to nodes. The AppInsight for Exchange template includes several component monitors with default settings that cannot be modified due to dependencies. Also, you cannot add component monitors to this template.

You may need to log in with an administrator account to edit templates.

See the SAM Template Reference for a list of component monitors included in this AppInsight template. You can also select the template on the Manage Application Monitor Templates page and click Edit to display component monitor details, and then make any necessary changes.

To begin gathering metrics, assign the template to Exchange servers.

Advanced manual configuration of AppInsight for Exchange

Manual configuration is only recommended for experienced Exchange administrators. To learn about the automated method that uses a customer PowerShell script to make configuration changes, see Configure AppInsight for Exchange on nodes.

If the automated configuration of AppInsight for Exchange via Discovery fails, follow these steps to manually configure AppInsight for Exchange:

1. Verify Microsoft Exchange credentials
2. Manually configure an Exchange server
3. Set PowerShell permissions for Exchange
4. Find Exchange URL settings
5. Edit the AppInsight for Exchange template

Verify Microsoft Exchange credentials

Here are requirements for accounts used to access Exchange:

- The account must be a domain account.
- To modify IIS and PowerShell settings on the Exchange server, the account must be a Local Administrator.
- SolarWinds recommends using accounts that are not part of the Domain Administrators group. You can add a custom domain security group to define a specific type of administrator with specific permissions to the Local Administrators group. In large environments, use Group Policy Objects (GPOs). For smaller environments, you can perform create security groups manually.

To verify Exchange credentials, run this PowerShell cmdlet in the Exchange Management Shell (EMS): Get-ManagementRoleAssignment -RoleAssignee “USER_IDENTIY”
Manually configure an Exchange server

Manual configuration is only recommended for experienced Exchange administrators. SAM includes an automated way to configure Exchange servers, as described in Configure AppInsight for Exchange on nodes.

Before manually configuring an Exchange server for AppInsight for Exchange:

- Review AppInsight for Exchange requirements and permissions.
- Make sure to have credentials and a proper Exchange account
- Review the configuration changes to enable AppInsight for Exchange

Use the following instructions to configure an Exchange server:

- Define Exchange credentials
- Grant Exchange access
- Set mailbox search access
- Install PowerShell 2.0 or later
- Set PSLanguageMode to FullLanguage for the PowerShell website
- Create a self-signed certificate
- Configure WinRM 2.0 on an Exchange server
- Create a firewall rule
- Configure IIS
- Test the application

Define Exchange credentials

Use domain accounts to access Exchange Management interfaces; AppInsight for Exchange does not support local accounts. Select an existing Active Directory account or create one to use with AppInsight for Exchange. See Verify Microsoft Exchange credentials.

1. On the server where you are granting local administrative privileges, open the Computer Management console.

   On Windows Server 2012, use the Active Directory console to manage administrative privileges.

2. Navigate to the Administrators group.

3. Add the type in the Active Directory user name of the account. (Ensure the location is set to either the domain where the account is located or Entire Directory.)

4. Save your changes.

   Alternatively, add an Active Directory group to the local administrators group and add Active Directory user accounts to that group.
To verify the account and local group membership was configured properly, run the following in a PowerShell session:

```powershell
$Recurse = $true
$GroupName = 'Administrators'
Add-Type -AssemblyName System.DirectoryServices.AccountManagement
$group = [System.DirectoryServices.AccountManagement.GroupPrincipal]:FindByIdentity($ct,$GroupName)
$LocalAdmin = $group.GetMembers($Recurse) | select @{N='Domain'; E={$_.Context.Name}}, samaccountName, @{N='ObjectType'; E={$_.StructuralObjectClass}} -Unique
$LocalAdmin = $LocalAdmin | Where-Object {$_:ObjectType -eq "user"}
```

<table>
<thead>
<tr>
<th>Domain</th>
<th>SamAccountName</th>
<th>ObjectType</th>
</tr>
</thead>
<tbody>
<tr>
<td>lab.excai.en03</td>
<td>labuser</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>test1</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>monitoruser</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>labadmin</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>tedsowards</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>EowenQA</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>Tsi23</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>ncssearch</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>noexchange</td>
<td>user</td>
</tr>
<tr>
<td>lab.excai.en03</td>
<td>discoveruser</td>
<td>user</td>
</tr>
</tbody>
</table>

Grant Exchange Access

To grant Least Privilege access to the Exchange Organization:

1. Open Active Directory Users and Computers (ADUC) and find the Microsoft Exchange Security Groups OU.
2. From the View-Only Organization Management group, add the user name of the account you want to grant access to the Exchange organization.

⚠️ See Microsoft.com for detailed instructions.
Set Mailbox Search Access

Mailbox Search access is required to determine attachment counts and sizes.

1. From the Start menu, open the Exchange Management Shell (EMS).
2. Type: `New-ManagementRoleAssignment -Role "Mailbox Search" -User <Username of account being granted access>` and then press Enter.
3. To verify the management role has been properly assigned, enter the following command:
   `Get-ManagementRoleAssignment -RoleAssignee <Username of account>`

Install PowerShell 2.0 or later

PowerShell 2.0 or later is usually installed with Microsoft Server. Install it, if necessary (see [Use PowerShell in SAM](#)). You may also need to [Set PowerShell permissions for Exchange](#).

Set `PSLanguageMode` to `FullLanguage` for the PowerShell website

Use IIS Manager on the Exchange server to configure application settings for the default website and PowerShell virtual directory, and then recycle the MSExchangePowerShellAppPool application pool.

See Microsoft.com for detailed instructions.

Create a self-signed certificate

You can download a PowerShell script to create a self-signed certificate suitable for AppInsight for Exchange from the SolarWinds Success Center. See [Create a self-signed certificate for AppInsight for Exchange with a PowerShell script](#).

Alternatively, follow these steps to create your own certificate.

1. Using PowerShell and CertEnroll, open PowerShell in the Run as Administrator context.
2. Enter the following code:

   ```powershell
   $name = new-object -com "X509Enrollment.CX500DistinguishedName.1"
   $name.Encode("CN=TestServer", 0)

   $key = new-object -com "X509Enrollment.CX509PrivateKey.1"
   $key.ProviderName = "Microsoft RSA SChannel Cryptographic Provider"
   $key.KeySpec = 1
   $key.Length = 1024
   $key.SecurityDescriptor = "D:PAI(A;;0xd01f01ff;;;SY)(A;;0xd01f01ff;;;BA)(A;;0x80120089;;;NS)"
   $key.MachineContext = 1
   $key.Create()
   ```
Configure WinRM 2.0 on an Exchange server

1. Open a command prompt in the Run as Administrator context.

2. Type: `winrm create winrm/config/listener?Address=*+Transport=HTTPS @ {Port="5986";CertificateThumbprint="<Thumbprint value of certificate>";Hostname="<IP Address of Server>_Solarwinds_Exchange_Zero_Configuration"}` and press Enter.

3. Verify the configuration by typing: `winrm get winrm/config/listener?Address=*+Transport=HTTPS`.
Create a firewall rule

1. Open PowerShell using Run as Administrator.
2. Create a function for adding firewall rules using the following code:

   ```powershell
   function Add-FirewallRule {
     param(
       $name,
       $tcpPorts,
       $appName = $null,
       $serviceName = $null
     )
     $fw = New-Object -ComObject hnetcfg.fwpolicy2
     $rule = New-Object -ComObject HNetCfg.FWRule
     $rule.Name = $name
     if ($appName -ne $null) { $rule.ApplicationName = $appName }
     if ($serviceName -ne $null) { $rule.serviceName = $serviceName }
     $rule.Protocol = 6 # NET_FW_IP_PROTOCOL_TCP
     $rule.LocalPorts = $tcpPorts
     $rule.Enabled = $true
     $rule.Grouping = "@firewallapi.dll,-23255"
     $rule.Profiles = 7 # all
     $rule.Action = 1 # NET_FW_ACTION_ALLOW
     $rule.EdgeTraversal = $false
     $fw.Rules.Add($rule)
   }
   ``

3. Run the function to create the firewall exception for WSMAN with this command:

   ```powershell
   Add-FirewallRule "Windows Remote Management" "5986" $null $null
   ```
4. Verify the rule was created.

Configure IIS

1. Open a command prompt in the Run as Administrator context.
2. Change to the C:\Windows\System32\Inetsrv directory.
4. Open PowerShell in the Run As Administrator context.
5. Type: Import-Module WebAdministration and press Enter.
6. Type: (Get-WebConfiguration system.webServer/security/authentication/windowsAuthentication 'IIS:\sites\Default Web Site\PowerShell').enabled and press Enter.
7. If the return value is True, Windows Authentication is configured. If the value returned is False, follow these steps:
   
   i. Type: Set-WebConfiguration system.webServer/security/authentication/windowsAuthentication 'IIS:\sites\Default Web Site\PowerShell' -value True and then press Enter.
   
   ii. Type: (Get-WebConfiguration system.webServer/security/authentication/windowsAuthentication 'IIS:\sites\Default Web Site\PowerShell').enabled to verify the setting changed.

   iii. Close PowerShell.

   iv. In the open command prompt, type: appcmd.exe lock config -section:system.webServer/security/authentication/windowsAuthentication and then press Enter.

   v. Close the command prompt.
Test the application

Navigate to the Application Edit page and click Test. Your screen should look like the following illustration.

Set PowerShell permissions for Exchange

Your organization should internally review and assess to what extent PowerShell is incorporated into your environment. This is especially important when importing scripts from third parties, including content posted by other customers in the SolarWinds online IT community, THWACK. To learn more, see Use PowerShell in SAM.

Certain PowerShell permissions are required for AppInsight for Exchange and the Exchange server.

1. On the remote computer, open the PowerShell console.
2. Execute this command: `Set-PSSessionConfiguration Microsoft.Powershell -ShowSecurityDescriptorUI -Force`
3. When the Permissions dialog box appears, enable Full Control under the Permissions for Everyone group, and select Allow.
4. Ensure that the group containing the polling user can access PowerShell, and click OK.
5. Verify all permissions are set and saved.
Resolving errors after a reinstall

If PowerShell was installed on Windows Server 2012 with Exchange 2013 and subsequently uninstalled, a Microsoft error removes the required registry key for remote PowerShell to work properly. Security patches or updates may also cause this issue. You can download PowerShell 2.0 or later from https://docs.microsoft.com/en-us/powershell (@2018 Microsoft Corp. Link obtained on July 23, 2018.)

SolarWinds strongly recommends that you back up your registry before editing it. Only edit the registry if you are experienced and confident in doing so. Using a registry editor incorrectly can cause serious issues with your operating system, which could require you to reinstall your OS to correct them. SolarWinds cannot guarantee resolutions to any damage resulting from making registry edits.

To recreate the required registry key:

1. In Notepad, copy and paste the following text:
   Windows Registry Editor Version 5.00
   
   [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\1\PowerShellEngine]
   "ApplicationBase"="C:\\Windows\\System32\\WindowsPowerShell\\v1.0"
   "PSCompatibleVersion"="1.0, 2.0"
   "RuntimeVersion"="v2.0.50727"
   "ConsoleHostAssemblyName"="Microsoft.PowerShell.ConsoleHost, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35, ProcessorArchitecture=msil"
   "ConsoleHostModuleName"="C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\Microsoft.PowerShell.ConsoleHost.dll"
   "PowerShellVersion"="X.0"

2. Update the PowerShellVersion variable in the last line.

3. Save the file as PowerShellv1.reg and then double-click it to add it to the registry. (A reboot may be required.)

Find Exchange URL settings

As part of the AppInsight for Exchange configurations, you should verify the PowerShell Exchange and WinRM URLs are correct. By default, AppInsight for Exchange uses the following URLs for the Exchange and WinRM sessions, where ${IP} is the IP address of the server node being added.

- Exchange: https://${IP}/powershell/
- WinRM: https://${IP}:5986/wsman/

Verify the PowerShell instance used by Exchange on a server:

1. Open the IIS Manager and navigate to the default website then to the PowerShell virtual directory.
2. Verify the Virtual Path value (typically in the Advance Settings).
Verify a server's WinRM PowerShell instance:

1. Open a command prompt using Run as Administrator.
2. Enter the command `winrm get winrm/config/listener?Address=*+Transport=HTTPS` to get the current configuration for the HTTPS protocol.

```
C:\Windows\system32>winrm get winrm/config/listener?Address=*+Transport=HTTPS
Listener
  Address = *  
  Transport = HTTPS 
  Port = 9886 
  Hostname = 10.199.3.122_Solarwinds_Exchange_Zero_Configuration 
  Enabled = true 
  URLPrefix = Custom_wsman 
  CertificateThumbprint = 78A4B5FFA3450BBAB20210F9564015CF1D2EC628 
  ListeningOn = 10.199.3.122, 127.0.0.1, 169.254.96.84, ::1
```

The two items of interest for the URL are Port and URLPrefix. If either of these have been modified, and do not match the default values, edit the AppInsight for Exchange application with the correct values:

- **PowerShell Exchange URL**: `https://${IP}/powershell/
- **PowerShell Windows URL**: `https://${IP}:9886/Custom_wsman/`
Additionally, the value of Hostname must match the CN of the certificate listed in the Certificate Thumbprint property:

For more information, see:
- [Manually configure an Exchange server](#)
- [Set PowerShell permissions for Exchange](#)

**Attachment extensions acknowledged by AppInsight for Exchange**

The following table lists the default attachment extensions that are acknowledged by AppInsight for Exchange:

<table>
<thead>
<tr>
<th>.accdb</th>
<th>.doc</th>
<th>.inf</th>
<th>.mmp</th>
<th>.ppt</th>
<th>.swf</th>
<th>.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td>.arj</td>
<td>.dot</td>
<td>.ini</td>
<td>.mobi</td>
<td>.ps1</td>
<td>.tar</td>
<td>.xps</td>
</tr>
<tr>
<td>.avi</td>
<td>.eml</td>
<td>.iso</td>
<td>.mov</td>
<td>.pst</td>
<td>.tmp</td>
<td>.zip</td>
</tr>
<tr>
<td>.bak</td>
<td>.epub</td>
<td>.jar</td>
<td>.mp3</td>
<td>.pub</td>
<td>.txt</td>
<td></td>
</tr>
<tr>
<td>.bat</td>
<td>.flv</td>
<td>.jpg</td>
<td>.mpeg</td>
<td>.psd</td>
<td>.vbs</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>.bin</td>
<td>.gif</td>
<td>.jpeg</td>
<td>.msg</td>
<td>.rar</td>
<td>.wav</td>
<td></td>
</tr>
<tr>
<td>.bmp</td>
<td>.gifz</td>
<td>.lnk</td>
<td>.pdf</td>
<td>.reg</td>
<td>.wks</td>
<td></td>
</tr>
<tr>
<td>.cab</td>
<td>.hta</td>
<td>.log</td>
<td>.png</td>
<td>.rtf</td>
<td>.wma</td>
<td></td>
</tr>
<tr>
<td>.cmd</td>
<td>.htm</td>
<td>.mdb</td>
<td>.pot</td>
<td>.sql</td>
<td>.wmv</td>
<td></td>
</tr>
<tr>
<td>.csv</td>
<td>.img</td>
<td>.mid</td>
<td>.pps</td>
<td>.svg</td>
<td>.xls</td>
<td></td>
</tr>
</tbody>
</table>
AppInsight for IIS

AppInsight for IIS automatically monitors your IIS environment to identify IIS server, website, and application pool performance issues. The dashboard provides ease of monitoring with at-a-glance performance metrics for sites and applications pools, updated through access to the IIS. As sites and application pools change through the IIS Manager, the information and connections update in SAM. As you determine issues, stop or restart servers and pools as required.

AppInsight for IIS leverages PowerShell to collect data about your IIS servers. You need PowerShell 2.0 or later installed on the Orion server. You also need WinRM installed and configured on the Orion server and monitored IIS servers. For full details, see AppInsight for IIS requirements and permissions.

To start using AppInsight for IIS, apply it to a monitored node through Discovery or manually by node. Enter access credentials to add and monitor the IIS servers.

To quickly set up and use AppInsight for IIS, see:

- AppInsight for IIS requirements and permissions
- Configure AppInsight for IIS on nodes
- Advanced manual configuration of AppInsight for IIS

For advanced managing of IIS sites, disable sites when unused instead of deleting them. To reduce alert noise, disable any alerts associated with those disabled sites. Enable sites anytime you need them through the Site Details pages in AppInsight for IIS. This option and others are available in the Management widget per ISS server, site, and application pool.

To display data for nodes integrated with SolarWinds APM, access the Use the APM subview from the left menu of the AppInsight for IIS Application Pool Details view.

The following AppInsight for IIS reports are pre-installed:

- IIS SSL Certificate Expiration Report
- Site Connections Report
- Site Log Size by File
- Site Size by File

To learn more about AppInsight for IIS, see:

- IIS Web Server Monitoring Tool: AppInsight for IIS (SolarWinds YouTube video)
- Geek Speak: IIS Web Server Monitoring: Best Practices (THWACK)

Navigate to the AppInsight for IIS details view

Access AppInsight for IIS through the All Applications widget in the SAM Summary page.

The default IIS Application name is Microsoft IIS. You may need to log in with an administrator account to perform these actions.
To view the AppInsight for IIS details view:

1. In the Orion Web Console, click My Dashboards > SAM Summary.
2. Locate the All Applications widget and expand the AppInsight for IIS tree by clicking [+].
3. Click an IIS application to view it.

![AppInsight for IIS](image)

The Sites widget lists the sites (front and backend) for the IIS server. Click a site to view additional site details including response time, connections, and requests.

<table>
<thead>
<tr>
<th>Sites</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>STATE</td>
<td>CONNECTIONS</td>
<td>RESPONSE TIME (AVG)</td>
<td>START MODE</td>
</tr>
<tr>
<td>Solar Winds NetPerfMon</td>
<td>Started</td>
<td>18</td>
<td>N/A</td>
<td>Auto</td>
</tr>
</tbody>
</table>

The Application Pools widget lists up to 5 instances of IIS application pools. You use application pools to separate out applications from one another. If an issue occurs with an application, it may only affect other applications in the pool, not all applications in your environment.

Click a pool to view additional worker process details.

![Application Pools](image)

<table>
<thead>
<tr>
<th>Application Pools</th>
<th></th>
<th>WORKER PROCESSES</th>
<th>CPU</th>
<th>PHYSICAL MEMORY</th>
<th>VIRTUAL MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solar Winds Orion Application Pool</td>
<td>Started</td>
<td>1</td>
<td>0.00%</td>
<td>6.49%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

To view the Performance Counter details view, click a performance monitor in any widget.

**AppInsight for IIS requirements and permissions**

Review the requirements and permissions before configuring AppInsight for IIS nodes in your environment. AppInsight for IIS data is collected at the same default 5-minute polling interval as other application templates.

💡 Use the free [Remote Execution Enabler for PowerShell tool](https://supportportal.solarwinds.com/kb.html?view_page=295) to configure secure WinRM across the Orion server and target servers. See this [THWACK article](https://community.solarwinds.com/article/295) for details.
Following are AppInsight for IIS requirements for nodes:

- PowerShell 2.0 or later must be installed. Ensure the WinRM service is running and the Startup Type is set to Automatic. See Use PowerShell in SAM for details.
- Administrator rights or equivalent credentials to the IIS server being monitored are needed for configuration. Non-administrative permissions for polling and monitoring is only achievable using the optional Orion Agent.
- IIS 7.0 or later must be installed
- AppInsight for IIS supports the following OS and IIS versions:

<table>
<thead>
<tr>
<th>MICROSOFT OS</th>
<th>SUPPORTED IIS VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Server 2008</td>
<td>IIS 7.0</td>
</tr>
<tr>
<td>Microsoft Server 2008 R2 and Windows 7</td>
<td>IIS 7.5</td>
</tr>
<tr>
<td>Microsoft Server 2012 and Windows 8</td>
<td>IIS 8.0</td>
</tr>
<tr>
<td>Windows Server 2012 R2 and Windows 8.1</td>
<td>IIS 8.5</td>
</tr>
<tr>
<td>Windows Server 2016 and Windows 10</td>
<td>IIS 10</td>
</tr>
</tbody>
</table>

If a prerequisite is missing, AppInsight for IIS goes into an Unknown state.

AppInsight for IIS technologies and ports

- The IIS server must have the ports open on the managed nodes.

<table>
<thead>
<tr>
<th>TECHNOLOGY</th>
<th>PORT</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPC Endpoint Mapper</td>
<td>TCP port 135</td>
<td>SAM uses this port to establish WMI/RPC connections to the remote computer. RPC is required to gather performance counter data via the ASP.NET resource.</td>
</tr>
<tr>
<td>WMI</td>
<td>TCP ports 1025 - 5000 or TCP ports 49152 - 65535</td>
<td>By default, Windows uses a random port from this range for WMI communications. The default port range differs based on the OS so you'll need to create a firewall exception on the remote computer.</td>
</tr>
<tr>
<td>PowerShell</td>
<td>TCP port 5986</td>
<td>A secure listener hosted in the WinRM service.</td>
</tr>
<tr>
<td>HTTP</td>
<td>TCP, At least one port mentioned in the bindings of a site.</td>
<td>If the connection is not allowed, the HTTP Monitor is hidden.</td>
</tr>
<tr>
<td>TECHNOLOGY</td>
<td>PORT</td>
<td>NOTES</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTTPS</td>
<td>TCP. At least one</td>
<td>If the connection is not allowed, the HTTPS Monitor is hidden.</td>
</tr>
<tr>
<td></td>
<td>port mentioned in the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>secure bindings of a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>site.</td>
<td></td>
</tr>
<tr>
<td>SSL</td>
<td>TCP. At least one</td>
<td>If the connection is not allowed, the SSL Certificate Expiration</td>
</tr>
<tr>
<td></td>
<td>port mentioned in the</td>
<td>Date Monitor is hidden.</td>
</tr>
<tr>
<td></td>
<td>secure bindings of a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>site.</td>
<td></td>
</tr>
<tr>
<td>SMB (Windows Shares)</td>
<td>TCP port 445</td>
<td>Used for Site Directory Information and Log Directory Information.</td>
</tr>
</tbody>
</table>

### AppInsight For IIS licensing

When using AppInsight for IIS, 30 component monitors count against your licensed number of component monitors, per monitored IIS server.

For example, if you have an active license for 1,500 component monitors and use AppInsight for IIS to monitor 10 IIS servers, 300 component monitors count against your total license. The number of sites and application pools you have on these servers is irrelevant.

\[
(30 \text{ component monitors} \times 10 \text{ IIS servers} = 300 \text{ component monitors used.})
\]

This leaves you with 1,200 component monitors available for use elsewhere.

\[
(1,500 \text{ component monitors} - 300 \text{ component monitors used for AppInsight for IIS} = 1,200 \text{ component monitors remaining}).
\]

The example below illustrates a situation where 40 component monitors are available, but surpassed the allowed number of 300 monitors by 60. The discrepancy is due to the fact that AppInsight applications cannot be partially licensed, as is the case with typical applications.

---

<table>
<thead>
<tr>
<th>SAM</th>
<th>Server &amp; Application Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Name</strong></td>
<td>6.1.0</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Service Pack</strong></td>
<td>Production</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Allowed Number of Component Monitors</strong></td>
<td>300</td>
</tr>
<tr>
<td><strong>Total Number of Component Monitors</strong></td>
<td>360</td>
</tr>
<tr>
<td><strong>Licensed Component Monitors</strong></td>
<td>260</td>
</tr>
<tr>
<td><strong>Unlicensed Component Monitors</strong></td>
<td>100</td>
</tr>
<tr>
<td><strong>Available Component Monitors</strong></td>
<td>40</td>
</tr>
</tbody>
</table>
For more information, see [SAM licensing model](#).

**Configure AppInsight for IIS on nodes**

SolarWinds SAM offers two automated ways to configure AppInsight for IIS on nodes:

- For new nodes, use the Add Node wizard.
- For an existing node, navigate to the Node Details page and access the Configure Server page to launch an automated process.

Before adding AppInsight for IIS to nodes, review [AppInsight for IIS requirements and permissions](#).

**Configure AppInsight for IIS on nodes with the Add Node wizard**

Check out [this video](#) on configuring AppInsight for IIS on nodes.

To quickly configure AppInsight for IIS with the Add Node wizard:

1. In the Orion Web Console, click Settings > All Settings > Add Node.
2. Complete the information on the Define Node step, and then click Next.
3. On the Choose Resources step in the Add Node Wizard, select AppInsight for IIS.
4. Click Next and complete the wizard as instructed.
5. Navigate to the SAM Summary page.
6. In the All Applications widget, expand the AppInsight for IIS tree view to display the newly added node.
7. Click Microsoft IIS to open the Configure IIS Server for Monitoring dialog box.

8. Enter Exchange credentials and click Configure Server to start the automated process. A confirmation message appears when the process is complete.

   If the node enters an Unknown state, check [AppInsight for IIS requirements and permissions](#) for the target server.

**Configure AppInsight for IIS via the Node Details page**

To add AppInsight for IIS via the Node Details page:

1. Navigate to the Node Details page for the node.
2. In the Management widget, click List Resources.
3. Select the Microsoft IIS check box and click Submit.

If the Microsoft IIS option does not appear, review [AppInsight for IIS requirements and permissions](#) on the target server.

4. Navigate to the SAM Summary page.

5. In the All Applications widget, expand the AppInsight for IIS tree view to display the newly added node.
6. Click Microsoft IIS to open the Configure IIS Server for Monitoring dialog box.

![Configure IIS Server For Monitoring dialog box]

7. Enter Exchange credentials when prompted and click Configure Server to start the automated process. A confirmation message appears when the process is complete. If the node enters an Unknown state, check [AppInsight for IIS requirements and permissions](#) for the target server.

For more information, see:
- [AppInsight for IIS requirements and permissions](#)
- [Edit the AppInsight for IIS template](#)

Integrate SolarWinds APM with IIS nodes

SolarWinds APM is a Software-as-a-Service (SaaS)-delivered product that extends the application monitoring capabilities of SolarWinds SAM. It delivers advanced performance metrics for IIS-based, .NET applications that are integrated into the SAM experience. By monitoring application performance, application dependencies, transaction times, and overall user experiences, you can better understand why transactions in your applications are slow or failing as fast as possible.
You can use SolarWinds APM to cross-reference application, server, and infrastructure metrics side-by-side in the same dashboard, and trace requests through your entire application and infrastructure. APM also includes heat maps to isolate outliers and trends, which you can use to drill down to traces for root cause.

SolarWinds APM adds an extra layer of information to nodes monitored by AppInsight for IIS that enables you to:

- Gain a code-level look into your AppInsight for IIS applications.
- Monitor the performance of custom IIS-based, .NET applications.
- Check the status of application stacks and see how tiers of applications interact with each other.
- Determine the databases and users that applications send information to.
- Identify if issues are network-, server-, or application-related so you can route them to the right team.

The following chart highlights the benefits of adding APM integration to IIS nodes monitored in SAM:

<table>
<thead>
<tr>
<th>Feature</th>
<th>SAM</th>
<th>APM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom application performance monitoring</td>
<td>☐</td>
<td>☑</td>
</tr>
<tr>
<td>Deep application performance visualization &amp; tracing</td>
<td>☐</td>
<td>☑</td>
</tr>
<tr>
<td>Application dependency mapping</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Hyper-V and ESX health and performance metrics</td>
<td>☑</td>
<td>☐</td>
</tr>
<tr>
<td>Server volume monitoring and capacity planning</td>
<td>☑</td>
<td>☐</td>
</tr>
<tr>
<td>Server hardware health and performance monitoring</td>
<td>☑</td>
<td>☐</td>
</tr>
<tr>
<td>Built-in alerts and reporting</td>
<td>☑</td>
<td>☐</td>
</tr>
</tbody>
</table>
After you integrate SolarWinds APM with an IIS node, visualizations of key metrics — such as a breakdown of response time into applications, remote calls, and database queries — appear on the subview of the IIS Application Pool details page.

Integrate APM with an IIS node in SAM

APM requires SAM 6.7 or later. Before using APM, review the latest SAM system requirements, along with AppOptics supported platforms and operating systems.

The first time you integrate SolarWinds APM with an IIS node monitored in the Orion Platform by SAM, you’re prompted to create an APM account or provide credentials for an existing APM account.

When you add APM to a node, IIS services restart on the node; related websites will go down.

To integrate SolarWinds APM with an IIS node for the first time, follow these steps:

1. In the Orion Web Console, navigate to Settings > All Settings.
2. Under Product Specific Settings, click APM Deployment Summary.
3. On the APM Deployment Summary page, click Add.

4. When the Add an Integration wizard appears, click Create a new account.

If you already have an APM account, click Go to APM Login. See the APM Administrator Guide to learn about retrieving an APM API token.
5. On the Log In page, provide account details and click Sign Up.
The Orion Platform generates a unique API token that provides access to APM data and adds it to the Orion database.
6. On the Select Nodes page, select the IIS node(s) you want to integrate with APM and click Deploy.

If no nodes appear, check that AppInsight for IIS is configured correctly on the target server. See Configure AppInsight for IIS on nodes and Troubleshooting AppInsight for IIS.

The Orion Platform deploys APM agents to nodes and restarts IIS services on remote machines. After a few minutes, APM performance metrics appear in widgets such as IIS Pools with APM Data and Top IIS Pools.

For more information about APM, see the APM Administrator Guide. To learn about SolarWinds AppOptics, where you can display in-depth details about APM data shown in the Orion Web Console, see AppOptics documentation.

SolarWinds SAM was formerly called APM so you may see legacy references to APM in certain folders, URLs, and the Orion database that are related to SAM.

Use the APM subview

For IIS pools on nodes integrated with SolarWinds APM, the SolarWinds APM subview is accessible from the AppInsight for IIS Application Pools Details view in SolarWinds SAM. Hover over the left menu and click the SolarWinds APM icon.

This view includes the following default widgets:

- Average Response Time
- HTTP (5xx) Error Rate
- HTTP Methods
- HTTP Status Codes
- Requests per second
Within these widgets, you can:

- Hover over points on graphs to display specific information.
- Click the time period hyperlink to select a different range (for example, Last 12 hours or Last 12 days) or configure a specific time and date range. Graphs show data for the last hour, by default.
Click Full Data View at the top of the APM subview to open a browser and log into SolarWinds AppOptics with your APM account credentials to display details about data shown on this subview, as shown in the following example:

To learn more about using AppOptics with your SolarWinds APM account, see AppOptics documentation.

**Edit the AppInsight for IIS template**

The AppInsight for IIS template includes numerous component monitors to provide data about IIS servers in your environment. If you modify these settings and configurations in the template, the details in AppInsight application monitors already assigned to servers update to match.

The primary reason to edit most SAM templates is to set general configurations such as polling frequency, polling method, and thresholds for warning and critical states for monitored metrics before assigning templates to nodes. The AppInsight for Exchange template includes several component monitors with default settings that cannot be modified due to dependencies. Also, you cannot add component monitors to this template.

You may need to log in with an administrator account to edit templates.

See the [SAM Template Reference](#) for a list of component monitors included in this AppInsight template. You can also select the template on the Manage Application Monitor Templates page and click Edit to display component monitor details, and then make any necessary changes.

To begin gathering metrics, assign the template to IIS servers.
Advanced manual configuration of AppInsight for IIS

If the automatic configuration of AppInsight for IIS via Discovery fails, follow the steps listed below to manually configure a remote computer.

For automatic AppInsight for IIS configuration see, Configure AppInsight for IIS on nodes.

1. Review AppInsight for IIS requirements and permissions.
2. Install PowerShell on target servers
3. Set the execution policy on target servers
4. Create a self-signed certificate
5. Create a firewall rule for AppInsight for IIS
6. Update WsMan limits
7. Create a WinRM listener for AppInsight for IIS

You can also edit the Edit the AppInsight for IIS template.

Install PowerShell on target servers

To support the ability of the Orion server to execute PowerShell commands against remote, target servers configured as managed nodes in SAM, you’ll need to verify that PowerShell 2.0 or later is installed on the target servers, along with other supporting software such as the WebAdministration snapin. WinRM must also be enabled.

SolarWinds provides customer support for PowerShell scripts and functionality that is built into SAM, but not for scripting languages or custom scripts. For scripting support from the SolarWinds online IT community, visit THWACK.

The easiest way to set up target servers for PowerShell is to configure AppInsight for IIS on each node.

Before proceeding, review AppInsight for IIS requirements and permissions.

There are several ways to configure target servers as AppInsight for IIS nodes in the Orion Web Console:

- Use the Discovery Wizard or Node Details page.
- Download the free Remote Execution Enabler for PowerShell tool to configure secure WinRM across one or more target servers. See this THWACK article for details.
- Navigate to the IIS application on a node to open the Application Details page. Click Edit Application Monitor to open the Edit Application page, and then click Configure Server, as shown here.
PowerShell 2.0 and later requires .NET Framework 3.5.x. If that version is not yet installed on target servers, use Server Manager's Add Roles and Features wizard to add it.

Set the execution policy on target servers
For AppInsight, the Execution Policy needs to be set to RemoteSigned on target servers.

1. Open a PowerShell session in the Administrator context (Right-click and select Run as Administrator).
2. Enter the following command: `Set-ExecutionPolicy RemoteSigned`

Create a self-signed certificate
You can download a PowerShell script to create a self-signed certificate suitable for AppInsight for IIS [here](#).

After downloading the file, execute the script by right-clicking it in Windows Explorer and selecting, Run with PowerShell. Use the following parameters:

- **IP address**: Mandatory
- **Certificate lifetime in days**: Optional

You can run this script with the default arguments from the PowerShell console or specify each one.

**For example:**

```
& '.\Create self-signed certificate script.ps1' 192.168.2.69 3650
```

where 192.168.2.69 is the IP address of the node to be monitored by AppInsight for IIS and 3650 is 3,650 days (10 years).

See also [Update WsMan limits](#).
Create a firewall rule for AppInsight for IIS

SolarWinds offers a PowerShell script that you can use to create firewall rules for IIS, available [here](https://www.solarwinds.com). Download, unpack, and execute the script by right-clicking it and selecting, Run with PowerShell. The following parameters apply:

- **Without parameters:** The rule is created with the default name, "Windows Remote Management HTTP/SSL" for port 5986.
- **With one parameter:** Non-default custom port.
- **With two parameters:** Non-default custom port and rule name

You can run this script with the default arguments from the PowerShell console or specify each one, as shown in this example:

```
& '.\Add firewall rule.psl' 5988 "My custom firewall rule name"
```

- **Important:** The default port for this rule is 5986 and does not need to be specified. Custom ports, as in the example above that uses port 5988, must be specified.

- **Important:** Your organization should internally review and assess to what extent PowerShell is incorporated into your environment. This is especially important when importing scripts from third parties, including content posted by other customers in the SolarWinds online IT community, THWACK. To learn more, see [PowerShell security considerations](https://www.solarwinds.com).

Update WsMan limits

The WSMan provider for Windows PowerShell provides methods and properties that can be used to create a session. SolarWinds provides a PowerShell script to update the WsMan limits suitable for AppInsight for IIS, available for download [here](https://www.solarwinds.com).

Once downloaded, execute the script by right-clicking it and selecting, Run with PowerShell. The following parameters apply:

- **maxConcurrentUsersDefaultValue** - Default value is 5
- **maxShellsPerUserDefaultValue** - Default value is 5
- **maxMemoryPerShellMBDefaultValue** - Default value is 150
- **serviceRestartRequired** - Default value is $false

You can run this script with the default arguments from the PowerShell console or specify each one.

**For example:**

```powershell
& '.\Update WsMan Limits.psl'
```
Create a WinRM listener for AppInsight for IIS

When AppInsight for IIS is configured for target servers, port 5968 is used as the WinRM HTTPS listening port, by default.

If you receive an "HTTPS listener currently exists on port 5986" message, follow these steps to switch the listening port to an available port.

1. To configure WinRM on an IIS server, open an elevated Windows PowerShell command prompt, to run PowerShell as an administrator.

2. Copy the following text to a text editor:
   ```powershell
   winrm create winrm/config/listener?Address=*+Transport=HTTPS @
    {Port="5986";CertificateThumbprint="<Thumbprint value of certificate>;Hostname="<IP Address of Server>_Solarwinds_Zero_Configuration"}
   ```

3. Change Port="5986" to match an available port.

4. Copy the text, paste it into the PowerShell command line, and press Enter. Results should look something like this:

5. To verify the configuration, type `winrm get winrm/config/listener?Address=*+Transport=HTTPS`.

6. Press Enter.
Results should look something like this:

```
C:\Windows\system32>winrm get winrm/config/listener?Address=*+Transport=HTTPS
Listener
  Address = *
  Transport = HTTPS
  Port = 5986
  Hostname = 10.199.3.122_Solarwinds_Exchange_Zero_Configuration
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint = 78a4b5ffe3450bba920210f9564015cf1d2ec628
  ListeningOn = 10.199.3.122, 127.0.0.1, 169.254.96.84, ::1
```

Find IIS URL settings

By default, AppInsight for IIS uses the following URL for the IIS and WinRM sessions, where $\{IP\}$ is the IP address of the server node being added.

https://$\{IP\}:5986/wsman/

Verify a server’s WinRM PowerShell instance

1. Open a command prompt in the Run as Administrator context.
2. Type: `winrm get winrm/config/listener?Address=\*+Transport=HTTPS` to get the current configuration for the HTTPS protocol.
Additionally, the value of Hostname must match the CN of the certificate listed in the Certificate Thumbprint property.
AppInsight for SQL

Identify performance and troubleshoot issues for SQL databases and queries with AppInsight for SQL. This feature provides consolidated resource views into over 100 performance metrics across all SQL servers monitored by SAM. These metrics include read and write latency, index fragmentation, expensive queries (based on CPU time), SQL agent job status with logs, capacity, and resource consumption for CPU, memory, and drive space. Through one page, monitor resource consumption, respond to alerts, and monitor expensive queries without digging through multiple pages and views.

Investigate issues and performance trends without hunting through numerous views into the SQL servers in your environment. AppInsight for SQL provides a level of detail and expert knowledge far beyond what a SQL template can provide, allowing you to monitor virtually every aspect of your SQL instances and databases. The feature polls and reports metrics without use of agents, directly accessing the SQL server using configured access permissions and credentials using SNMP and WMI.

You may need to log in with an administrator account to perform these actions.

To add a node and configure AppInsight for SQL, see the following:

- **Add AppInsight for SQL to a node**: to monitor SQL servers through the Discovery wizard or manually.
- **Monitor a SQL named instance with AppInsight for SQL**: to monitor a named SQL server using the server's Virtual IP (VIP).
- **Monitor clusters with AppInsight for SQL**: to monitor an SQL cluster (used for high availability).

When applied to a Microsoft SQL instance, the AppInsight for SQL page replaces the default node details page. It monitors as an application using multiple component monitors the AppInsight for SQL template. To modify the template or use other templates, see Edit the AppInsight for SQL template.

The views are accessible through one of the following:

- **SQL Server**: provides all monitored SQL applications. Select My Dashboards > Applications > SQL Server, locate and access a node.
- **AppInsight for SQL**: in the All Applications widget provides all monitored SQL server nodes. Select My Dashboards > Applications > SAM Summary, locate the All Applications widget. Expand AppInsight for SQL and select a node to view.
The SQL Server Application Summary view provides widgets you can use to monitor overall data for all SQL servers including alerts, events, and consumed resources, including:

- All Applications view of all currently managed SQL servers as nodes, expandable to locate specific nodes
- Active Application Alerts for specific alerts affecting SQL servers
- Top Processes by CPU Load, Physical Memory, Virtual Memory and more to gauge applications consuming resources
- Top Monitored Processes by I/O Total Operations, Reads, and Writers for highest bandwidth consumption, reads, and write latency
Further expanding the All Applications on the SQL Server Application Summary displays a quick view into performance counters and status. Each counter and metric can be expanded to review data.
To view all information for an application node, select the application from the view to open the details page for the AppInsight for SQL node. For example, MSSQLSERVER on lab-dem-sql-02.demo.lab in the Orion demo encountered a critical alert. To troubleshoot, access the SQL server through the All Applications widget.

This view consolidates all specific data for the selected SQL server:

- Summary of data including an AppStack view for troubleshooting, alerts, events, expensive queries, capacity usage, and other metrics
- Queries to filter expensive queries
- Transactions of WPM transaction monitors for that installed module
- DB performance for DPA IM data for that installed module
To review database data in the AppInsight for SQL Details view, locate and select a database in the All Databases widget. The Database Details view lists all databases on the node, alerts, AppStack, and more.

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATUS</th>
<th>DATABASE SIZE</th>
<th>TRANSACTION LOG SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>Online</td>
<td>4.00 MB</td>
<td>768.00 KB</td>
</tr>
<tr>
<td>model</td>
<td>Online</td>
<td>3.06 MB</td>
<td>768.00 KB</td>
</tr>
<tr>
<td>msdb</td>
<td>Online</td>
<td>18.38 MB</td>
<td>19.63 MB</td>
</tr>
<tr>
<td>SolarWindsOrion</td>
<td>Online</td>
<td>52.06 MB</td>
<td>18.13 MB</td>
</tr>
<tr>
<td>tempdb</td>
<td>Online</td>
<td>8.00 MB</td>
<td>768.00 KB</td>
</tr>
</tbody>
</table>

To see performance counter details, in the AppInsight for SQL Details view, locate, and select a performance counter in any widget.

Watch and learn about AppInsight for SQL.

Did you know that when SAM alerts you about an application failure, you can use SolarWinds IPAM to quickly determine if a broken or missing DNS record is at fault? You can then use SAM to troubleshoot the service further.

AppInsight For SQL licensing

Per the SAM licensing model, when using AppInsight for SQL, 50 component monitors count against your licensed number of component monitors per SQL instance.

AppInsight applications provide tremendous value within SAM while consuming a fixed number component monitor licenses. However, they cannot be partially unlicensed because the way they collect data differs significantly from traditional application templates. You can disable some components within AppInsight applications but they will not reclaim component monitor licenses because AppInsight applications typically monitor far greater than 50 components.

For example, AppInsight for SQL monitors over 120 individual metrics even if there is only one database running on the server. If you don't want to monitor certain metrics, remove the warning and critical thresholds for those components by editing the application. You will not be alerted or notified about those components again and the components will not appear in a warning or critical state in the Orion Web Console.

Add AppInsight for SQL to a node

With the credentials and permissions configured on the target SQL server, you can easily add AppInsight for SQL to SQL servers through the Discovery Wizard or manually. When added, you create a monitored node in SAM for the Microsoft SQL Server application. SAM polls the node for metrics including read and write latency, index fragmentation, expensive queries (based on CPU time), SQL agent job status with logs, capacity, and resource consumption for CPU, memory, and drive space.
To view monitored nodes, access the SQL Server page and specific SQL server details pages. See AppInsight for SQL for details.

The instance becomes enabled after the first poll, which may take a few minutes.

Add through the Discovery Wizard

Use the Discovery Wizard to add a new node and select AppInsight for SQL for monitoring.

1. Click Settings > Discovery Wizard.
2. In the Network panel, enter the IP addresses to scan and follow the onscreen instructions.
3. On the Applications panel, all detected and supported Microsoft SQL Servers display.
   Select the servers to add for monitoring as nodes. Only supported versions of Microsoft SQL are located and displayed for monitoring. To see all supported versions, see AppInsight for SQL requirements and permissions.
4. Click Next and complete the wizard as instructed.
5. Navigate to the All Applications widget and click your AppInsight for SQL application. The Enter Credentials page appears.

6. Enter your SQL credentials and select a port (or use default port).
7. Click Test to verify the credentials and configured permissions.
8. Click Assign Credential to save and complete configuration.

Add an SQL server as a node manually

When added, you need to select the appropriate Microsoft SQL server version from the AppInsight Applications list.

You may need to log in with an administrator account to perform this action.
1. In the Orion Web Console, click Settings > All Settings > Add Node.

2. Enter the IP address or hostname for the SQL server. Select the polling method and continue.

3. On the Choose Resources panel, check the appropriate AppInsight Application for the SQL server you will monitor.

4. Complete the steps, specifying metrics, custom polling engines, and other options, as desired.

5. Review and adjust any settings and click Add Node.

6. Navigate to the All Applications widget and click your AppInsight for SQL application.

   The Enter Credentials page appears.

7. Enter your SQL credentials and select a port (or use the default port).

8. Click Test to verify the credentials and configured permissions.

9. Click Assign Credential to save and complete configuration.
AppInsight for SQL requirements and permissions

AppInsight for SQL supports the following versions of Microsoft SQL Server:

- Microsoft SQL Server 2008, SP1, SP2, SP3
- Microsoft SQL Server 2008 R2, SP1, SP2 SP3
- Microsoft SQL Server 2012 SP1
- Microsoft SQL Server 2014
- Microsoft SQL Server 2014 SP1
- Microsoft SQL Server 2014 SP2
- Microsoft SQL Server 2016
- Microsoft SQL Server 2016 SP1
- Microsoft SQL Server 2017 (Windows only)

AppInsight for SQL does not require named-pipes. However, it does require TCP. For example, SAM uses TCP detection during discovery. You may receive an error message pertaining to "named-pipes." This is the result of the last client protocol that is tried during connection to the SQL server.

AppInsight for SQL permissions

Following are required permissions needed for AppInsight for SQL. See also SAM port requirements.

- Administrator permission at the host level.
- Be a member of the db_datareader role in the msdb database
- VIEW SERVER STATE permissions
- View any definition
- Connect permission to all databases, including Master and msdb
- Execute permission on the Xp_readerrorlog stored procedure

If utilizing a domain user for AppInsight for SQL, the domain user must be a member of the SQL server's local admin group.

Review the following information regarding monitoring SQL servers with AppInsight for SQL:

- AppInsight for SQL supports SNMP and WMI protocols and uses SQL to gather application data. Additional data is available for nodes managed via WMI.
- Agents do not work with AppInsight for SQL if the SQL server is monitored in a cluster.
- SQL clusters cannot be polled with domain credentials via the Orion agent because agents do not work with AppInsight for SQL if the SQL server is monitored in a cluster.

SQL account permissions

The following script configures permissions for a SQL account. You must connect to the SQL database server as "sa" or equivalent to create an account.

This script makes changes directly to the database. Create a database backup before running either script.
USE master
GRANT VIEW SERVER STATE TO AppInsightUser
GRANT VIEW ANY DEFINITION TO AppInsightUser
GRANT VIEW ANY DATABASE TO AppInsightUser
EXEC sp_adduser @loginame = 'AppInsightUser',@name_in_db = 'AppInsightUser'
GRANT EXECUTE ON xp_readerrorlog TO AppInsightUser
USE msdb
EXEC sp_adduser @loginame = 'AppInsightUser',@name_in_db = 'AppInsightUser'
EXEC sp_addrolemember N'db_datareader', N'AppInsightUser'

Windows Authentication

The following script configures permissions for a SQL account with Windows Authentication:

⚠️ This script makes changes directly to the database. **Create a database backup before running either script.**

USE master
GRANT VIEW SERVER STATE TO "Domain\AppInsightUser"
GRANT VIEW ANY DEFINITION TO "Domain\AppInsightUser"
EXEC sp_adduser @loginame = 'Domain\AppInsightUser',@name_in_db = 'Domain\AppInsightUser'
GRANT EXECUTE ON xp_readerrorlog TO "Domain\AppInsightUser"
USE msdb
EXEC sp_adduser @loginame = 'Domain\AppInsightUser',@name_in_db = 'Domain\AppInsightUser'
EXEC sp_addrolemember N'db_datareader', N'Domain\AppInsightUser'
EXECUTE sp_MSforeachdb 'USE []; EXEC sp_adduser @loginame = 'Domain\AppInsightUser', @name_in_db = 'Domain\AppInsightUser'''

Domain account with Orion agent

To use a domain account with an Orion agent, the domain account needs to have “**Log on as a batch job**” policy enabled for the default batch execution mode. Set this permission either locally on the monitored SQL server or as a domain policy (which enforces the policy to all machines within the domain). For details on batch mode, see [https://technet.microsoft.com/en-us/library/cc957131.aspx](https://technet.microsoft.com/en-us/library/cc957131.aspx) (© 2018 Microsoft Corp.)

This policy is only enabled for a LocalSystem account by default and explicitly needs to be added for the domain account.

This user right is defined in the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers. The location for the policy is **Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment.**

See also [Use a domain user to monitor AppInsight for SQL through an agent.](#)
AppInsight for SQL licensing

When using AppInsight for Exchange, 50 component monitors count against your licensed number of component monitors, per Exchange Server. See the SAM licensing model.

Monitor a SQL named instance with AppInsight for SQL

A named SQL instance is a SQL server given a name consisting of the network name of the server plus the instance name specified during installation. To monitor SQL named instances in SAM with AppInsight for SQL, you need the Virtual IP (VIP) of the SQL server and the full name.

Before adding the server as a node, verify the credentials and permissions are configured on the target SQL server.

<i>You may need to log in with an administrator account to perform this action.</i>

To view monitored nodes, access the SQL Server page and specific SQL server details pages. See AppInsight for SQL for details.

<i>The instance becomes enabled after the first poll, which may take a few minutes.</i>

Locate the VIP for the server

If you do not know the VIP, follow these steps to ping the server:

1. Open a command prompt.
2. Ping the server name: ping NAME.
3. Make note of the returned IP. You will use this IP when adding the server as a monitored node.

4. Close the command prompt.

Add the node manually with AppInsight for SQL

With the VIP, add the named SQL instance as a node for monitoring. These instructions manually add the node. You can also use the Discovery Wizard. For details, see Add AppInsight for SQL to a node.
1. In the Orion Web Console, click Settings > All Settings > Add Node.
2. Enter the VIP address for the SQL named instance for the Polling Hostname or IP Address.
3. Select a polling method, then select the polling engine for that node.
4. On the Choose Resources panel, check the appropriate AppInsight Application for the SQL server you will monitor.

![ADD NODE](image)

5. Complete the steps making selections as desired such as specific metrics or custom polling engines.
6. Review and adjust any settings and click Add Node.
7. Navigate to the All Applications widget and click your AppInsight for SQL application. The Enter Credentials screen displays.

![Enter Credentials](image)

8. Enter your SQL credentials and select a port (or use the default port).
9. Click Test to verify the credentials and configured permissions.
10. Click Assign Credential to save and complete configuration.
Monitor clusters with AppInsight for SQL

Your environment may include clustered SQL servers, perhaps for use with SolarWinds High Availability. These clustered SQL servers can be added and monitored with AppInsight for SQL through SAM.

Agents do not work to poll AppInsight for SQL when the SQL server being monitored is in a cluster.

Before adding the server as a node, verify the credentials and permissions are configured on the target SQL server.

You may need to log in with an administrator account to perform this action.

To view monitored nodes, access the SQL Server page and specific SQL server details pages. See AppInsight for SQL for details.

The instance becomes enabled after the first poll, which may take a few minutes.

First, you need the IP address of the cluster.

1. Connect to the instance that you want to monitor using SQL Management Studio.
2. Execute the following query to make sure you have the proper target node and instance name:
   ```sql
   SELECT SERVERPROPERTY('ServerName')
   ```
3. Determine the IP address of the target node.

   Open a command prompt and ping the server: `ping NAME`. For example, ping `P111SQLV23` in the following example. The IP address is determined to be `10.1.70.123`.

   ![Ping example output](image)

Next, add the node and assign AppInsight for SQL.

1. In the Orion Web Console, click Settings > All Settings > Add Node.
2. Enter the IP address for the SQL cluster for the Polling Hostname or IP Address.
If the IP address represents a cluster, the node name of the active cluster member populates. SolarWinds recommends changing the node name to something more easily understood during the final step of the Add Node Wizard. For example, for this cluster you could enter the name and (cluster): P111SQLV23 (cluster).

3. Select a polling method, then select the polling engine for that node.

4. In the Choose Resources step, select the AppInsight Application for the SQL server to monitor.

![ADD NODE]

5. Complete the wizard as instructed to add the node for monitoring.

6. Navigate to the All Applications widget and click your AppInsight for SQL application to display the Enter Credentials page.

![Enter Credentials]

7. Enter your SQL credentials and select a port (or use default port).

8. Click Test to verify the credentials and configured permissions.

9. Click Assign Credential to save and complete configuration.
Edit the AppInsight for SQL template

The AppInsight for SQL template includes numerous component monitors to provide data about monitored SQL instances. If you modify these settings and configurations in the template, the details in AppInsight application monitors already assigned to instances update to match.

The primary reason to edit most SAM templates is to set general configurations such as polling frequency, polling method, and thresholds for warning and critical states for monitored metrics before assigning templates to nodes. The AppInsight for SQL template includes several component monitors with default settings that cannot be modified due to dependencies. Also, you cannot add component monitors to this template.

To monitor specific instances without using AppInsight for SQL, SolarWinds recommends Monitor with other SQL application templates.

See the SAM Template Reference for a list of component monitors included in this AppInsight template. You can also select the template on the Manage Application Monitor Templates page and click Edit to display component monitor details, and then make any necessary changes.

You may need to log in with an administrator account to edit templates.

To begin gathering metrics, assign the template to a SQL instance.

Customize AppInsight for SQL monitoring

In addition to editing the AppInsight for SQL template, there are many ways to customize AppInsight for SQL monitoring, as described here.

- **Customize widgets on the AppInsight for SQL Details page**
- **Set custom properties for a node**
- **Remove a database from an active AppInsight for SQL node**
- **Monitor with other SQL application templates**

Customize widgets on the AppInsight for SQL Details page

With administrator privileges, you can change the widgets (formerly called "resources") that appear on an AppInsight Details page. Each widget provides data configuration and display options, consuming data provided by the template application and component monitors. See Add widgets and columns to views, and define subviews.

Set custom properties for a node

You can also set custom properties for the AppInsight for SQL application monitor assigned to a specific node.

1. Navigate to the AppInsight for Exchange Details page for a node.
2. Click Edit Application Monitor in the upper right-hand corner of the page.
3. Edit the Application Monitor as necessary and click Select resources as needed, then click Submit.
Consider adding the node ID in the Application Monitor Name field so you can easily identify this monitor later.

Remove a database from an active AppInsight for SQL node

If you need to remove a database from an active AppInsight for SQL node, one way to do it is to "unmanage" the node to halt polling and related alerts. To do so, navigate to the Node Details page, click Unmanage in the Application Details widget, and specify an interval for the assigned application monitor during which no statistics will be collected. For another workaround, see Hide Databases from AppInsight for SQL via the All Databases widget.

When a database, node, or interface is "Unmanaged" in SAM, the element is not polled and alerts are not triggered because the alert engine does not detect any change on the object. However, if you create a new alert or edit conditions for an existing alert for an unmanaged database/node/interface, the alert will be triggered and alert actions will occur because the Orion Platform does not check to see if the element is unmanaged before running the alert, by design.

To learn about continuing to gather statistics about a database, node, or interface while blocking a flood of alerts, visit the SolarWinds online IT community, THWACK, and read Tips & Tricks: Stop the Madness — Avoiding alerts but continuing to pull statistics.

Monitor with other SQL application templates

To monitor specific nodes without using AppInsight for SQL, SolarWinds recommends using SQL templates. For example, you may have a large amount of SQL database instances to monitor such as a service provider with SQL instances per customer. To avoid performance issues, you may not want an AppInsight for SQL running for each of these instances.

To monitor a SQL named instance, first add it as a standard node by navigating to the Manage Nodes page, clicking Add Node, and providing required details.

Do not select the option for AppInsight for SQL for the node yet — you can apply a SQL Server template on the Manage Templates page, as described next.

After adding the SQL named instance as a standard node, follow these steps to assign a SQL Server template to the node.

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Search or browse for SQL Server templates.
3. Select the SQL Server template you want to assign to SQL named instances to monitor. You may want to assign multiple templates to the node. Assign these one at a time.
4. Once assigned, you can edit one or more templates by checking the boxes and selecting MultiEdit.
5. Enter the SQL named instance for the SQL Server Instance.
6. You can further modify the templates and application monitors as needed. Data captured through the templates should display on the Node Details for those monitored nodes.
All component or application monitors in a template relate to the SQL named instance, not the server. In Microsoft, the SQL server is a series of instances. These instances are default, unnamed instances or specifically named SQL instances. An instance is an installed Microsoft SQL in a specific directory path.

For a list of available templates, see SAM Component Monitor Types and the SAM Template Reference. Refer to this THWACK article also, SQL Named Instance Monitoring.

Hide databases from AppInsight for SQL in the All Databases widget

These steps do not "unmanage" the database. They hide the database from appearing in the All Databases widget, but the Orion Platform will continue alerting on the database. To prevent a database from triggering alerts, unmanage the database on the Database Details page's Management widget instead.

Follow these steps to prevent databases from appearing in the All Databases widget that is part of AppInsight for SQL.

1. Navigate to the Management widget on the Database Details page and confirm that the database is currently being managed.
2. Navigate to the AppInsight for SQL Details page for your SQL instance.
3. In the URL of the browser, record the last few digits of the applicationID. You'll need it for a later step.

Example: http://Solarwinds/Orion/APM/SqlBlack...NetObject=ABSA:168

4. On the Orion server, open Database Manager by clicking Start > All Programs > SolarWinds Orion > Advanced Features > Database Manager
5. When the application loads, click Add Default Server
6. Expand the Orion Database to view tables for installed SolarWinds applications.
7. Locate the table 'APM_SqlBbDatabase'
8. Run the following query:

   SELECT TOP 1000 * FROM [dbo].[APM_SqlBbDatabase] where applicationId = 168

9. Update the ApplicationID to match your Application ID you recorded during step 3.
10. Click "Enable table editing" above the Query text box.
11. Clear the check box in the Visible column to hide the database from the All Databases widget on the AppInsight for SQL Details page. Repeat, as necessary, for other databases you want to hide.
12. Click 'Enable table editing' again to finish the editing session.

AppInsight for SQL alerts

AppInsight for SQL includes two predefined alerts:
- Alert me when my database file disk I/O latency is high
- Alert me when my database file is running low on space

**View Alerts**

> You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > Manage Alerts.

2. Select the alerts you want to view from the list, or type a keyword in the text box and click Search.
Manage SAM templates and component monitors

SAM includes over 250 out-of-the-box (OOTB) monitoring templates that you can assign to nodes and begin using immediately. Templates include component monitors for specific applications or for scripts, grouped by application categories.

The following diagram illustrates how you can assign different templates to nodes, and then display results in the Orion Web Console.

![Diagram showing assignment of templates to nodes and display of results in Orion Web Console]

Component Monitors

Component monitors are the building blocks of SolarWinds SAM. Each monitors the status and performance of a different aspect of an application. There are several different types of component monitors, each containing settings that define what is monitored and how to monitor it.

Some types of component monitors allow you to set threshold conditions on the monitored parameters. You can set separate thresholds to indicate warning and critical conditions. For example, if you are monitoring the percentage of free space remaining on a volume, you can set a warning threshold at 15%, and a critical condition at 5%.
As an analogy, pretend SolarWinds SAM is monitoring a car. You would have component monitors to check tire pressure, engine RPM, water temperature, battery voltage, and other important subsystems of that vehicle. You can set alerts to give notification if the water gets too hot, or if the battery voltage drops too low.

**Application Monitor Templates**

A template is a group of component monitors modeling the total availability and performance level of an application. A complicated application such as Windows Server may require dozens of component monitors to accurately assess its current status and performance.

Instead of creating component monitors one-by-one for every application server, you can assign a pre-made template. The template can either be one included with SolarWinds SAM, or a custom template you make yourself. For example, you can assign the included Microsoft Windows Server 2003-2008 template to your Windows 2003 and Windows 2008 computers and obtain vital statistics on all of them.

A template is only a blueprint and does not perform any monitoring on its own. Only after assigning the template to a server node are active assigned component monitors created.

To continue the car analogy, pretend you want to monitor a fleet of 50, 2010, blue Dodge Charger automobiles. Instead of defining the component monitors for 50 cars, you can define all the component monitors in a Dodge Charger template.

**Assigned Component Monitors**

Assigned component monitors are created by assigning Application Monitor templates to server nodes. Each actively monitors its assigned node according to its settings. Component monitors inherit these initial settings from the template. If you make a change to a template, that same change is rolled out to all assigned Application Monitors based on the template.

You can override the template settings at any time, breaking the inheritance relationship between the component monitor and its template. For example, the user name and password usually differ for each node, and you would select a different credential for each assigned Application Monitor, thus overriding the template setting for the Credentials field.

To restore the inheritance relationship between a component monitor and its template, click Inherit From Template next to the setting.

Continuing the car analogy, when you assign the Dodge Charger template to a Dodge Charger vehicle, you now have a set of assigned component monitors for monitoring the vehicle's tire pressures, engine RPM, and so forth.

**Assigned Application Monitors**

An assigned Application Monitor runs its assigned component monitors at regular intervals, and then uses the status results from the component monitors to determine an overall status for the application.

If some of the component monitors are up and others are down, the Application Monitor follows the Status Rollup Mode setting in the Orion Web Console Settings to show either the worst status of the group or a warning status.
The difference between an assigned Application Monitor and a template is that the template is only a blueprint and does not perform any monitoring on its own. Only after assigning the template to a server node does SolarWinds SAM conduct any actual monitoring on the node.

To complete the car example, you assign the Dodge Charger template to all the Dodge Charger vehicles to create the assigned Application Monitor for determining the overall status for your Dodge Charger fleet. For example, the fleet may be 95% available at a given time due to warnings for some of the cars.
The following diagram illustrates the workflow involved in creating an application to be monitored by SAM.

To begin using templates and component monitors, see Manage SAM templates and Work with component monitors in SAM templates. To learn about the component monitors includes in each OOTB template, see the SAM Template Reference.

See SAM port requirements to learn about template port requirements.
Learn more

- Understanding Application Templates (video)
- Creating a New Application Template (video)
- Managing Assigned Application Monitors (video)
- Building & Implementing Custom Application Templates (SolarWinds Academy)
- Expert Series: SAM - Alerting and troubleshooting templates (SolarWinds Academy)
- Using Automation to Apply SAM Templates (video)

Manage SAM templates

A template is the blueprint for an application. It is a collection of component monitors designed to monitor a server, application, or process. You can customize numerous templates using only the component monitors you need for a specific environment.

You can create a new template, copy a current template, or manage existing templates. You can access existing templates, assigned monitors, and import templates from THWACK via the All Settings > SAM Settings > Manage Templates page in the Orion Web Console.

ℹ️ You may need to log in with an administrator account to perform these actions.

💡 Watch and learn more about templates.

Templates include a number of component monitors you can add and then customize. When complete, assign the template to managed nodes to collect specific metrics at each polling.

SAM includes two options for creating new templates:

- (Recommended) Use the Component Monitor Template Wizard to create component monitors and add them to a new template. This option adds instances of a component monitor type with fewer steps to create a template.
- Create a fresh template without component monitors. You will need to add component monitors manually.

💡 Want to learn how to use Orion Platform groups, which can be tied to queries and alerts, to automatically apply and remove SAM templates? Check out the Using Automation to Apply SAM Templates video. See also Group monitored objects, in the Orion Platform Administrator Guide.

ℹ️ Component monitors may have prerequisites, configuration, and credentials requirements for target systems. See Component Monitor Types.

To manage templates, you can:

- Create a template
- Assign a template to a node
- Edit a template
- Copy a template
Create a template

The fastest method to create a template with a set of component monitors is using the Component Monitor Wizard. The wizard provides the most common and popular component monitors per platform. Using this wizard, you can create multiple instances of a selected component monitor to add to a new or existing template.

When done, you can edit the template to add additional component monitors as needed.

For example, you may need to monitor 20 services on servers and run custom PowerShell scripts. Creating a template using the wizard to easily generate 20 Process Monitors or Service Monitors (depending on your use case) for the services. After created, you can add additional PowerShell script monitors as needed.

You may need to log in with an administrator account to perform this action.

When you enter credentials for component monitors, a connection test runs. Depending on the components, you may need to enter additional server and credentials data.

1. Select a component monitor type to add to your application or template. These include monitors based on application category: Windows Systems, Linux - Unix Systems, VMWare Systems, and Applications.

2. Select a Target server by browsing to supported systems in your environment. You may also need to select a platform type and enter credentials for the target system.

   Select and enter all required information.

   A list of available processes, services, and performance counters displays for the server. Select options to create process monitors per service.

3. Customize the component monitor and click Next.

4. Create a new template using the New Application Monitor Template. Enter a name when prompted.
5. Assign the template to nodes in your environment according to server type (such as Windows or VMware).
   Expand and select one or more servers to assign the template.

6. Review the configured component monitors prepared to push to nodes. When ready, confirm the template creation.
   The template creates, populating with the component monitors and configured settings, assigned to the selected nodes.

If you do not want to use the wizard, you can also create a new, empty template and add component monitors as needed. For example, you may only want to create a template with PowerShell script monitors to gather data on your systems and applications.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings and select Manage Templates.
2. Click New Template.
3. Add general information and settings for the template including the name, description, and tags.

4. Depending on the load of the calls for the component monitors, set the polling frequency and timeout.

Continue by adding component monitors.

Create a custom template (Example)

The following procedure creates a SolarWinds SAM Application Monitor template that monitors a locally installed SQL Server instance. The template is simplified by using the Windows Service component monitors, a TCP port monitor for your SQL Server, and an HTTP monitor for the local Web Console.

Create an application template with the following component monitors:

- TCP port component monitor to monitor port 1433, the port through which SolarWinds communicates with the SQL Server.
- Service component monitors for the following windows services:
  - SolarWinds Alerting Engine
  - SolarWinds Network Performance Monitor
  - SolarWinds Job Engine
  - SolarWinds Job Scheduler
  - SolarWinds Module Engine
  - SolarWinds Syslog Service
  - SolarWinds Trap Service
HTTP component monitor to monitor port 80, the port through which you access the Orion Web Console.

Create a SolarWinds Windows service application template

You can create templates specifically for the following SolarWinds Windows services that support the Orion Platform:

- SolarWinds Network Performance Monitor
- SolarWinds Job Engine v2
- SolarWinds Job Scheduler
- SolarWinds Module Engine
- SolarWinds Syslog Service
- SolarWinds Trap Service

When creating the templates, use these instructions with variations based on the services you are monitoring:

1. Click Settings > All Settings > SAM Settings > Create New Template.
2. Add general information to the template including a name, description, and tags. Tags are used for searching or opening lists of templates.
3. Click Add Component Monitor, expand the Network Protocol Component Monitors list, and select TCP Port Monitor. Click Submit to add.
4. Click Rename and name the TCP port monitor, and click OK.
5. Ensure the Port Number field corresponds to the port used to communicate with the SolarWinds SQL Server instance. By default, this is port 1433.
6. Click Add Component Monitor, expand the Process and Service Component Monitors, and check Windows Service Monitor and click Submit.
7. Click Rename, name the SolarWinds Alerting Engine monitor, and click OK.
8. Enter or select the credential set to use when accessing the Windows service information.
9. Enter the name of the SolarWinds Alerting Engine service in the Net Service Name field.
10. Click Add Component Monitor then expand the User Experience Component Monitors list, and then check HTTP Monitor.
11. Click Rename, name the HTTP port monitor, then click OK.
12. Ensure the Port Number field corresponds to the port you use for the Orion Web Console port, then click Submit.

Assign a template to a node

To begin monitoring with templates, you need to assign the application templates and monitors to a node. When assigned and enabled, the template collects and reports on polling data to the node according to the application monitors and configuration settings.

💡 Use the Discovery Wizard to add nodes, if necessary.
To assign templates to nodes:

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Check the template(s) to assign, and click Assign to Node.
3. Specify the node(s) to monitor and click Next.
4. Enter or select the appropriate credentials.
5. Click Assign Application Monitors.

Edit a template

Every template, including default SAM templates, can be modified. You can change the name, description, general settings, and application monitors. You can also create a copy of a template and modify the copy.

1. In the Orion Web Console, click Settings > All Settings > Settings > Manage Templates.
2. Check an application template and click Edit.
3. Specify the values for the Polling Frequency and Polling Timeout fields.

   Setting a polling frequency below 30 seconds can result in erratic monitor behavior.

4. To add a monitor, click Add Component Monitor. Expand and filter through the options to add one or more monitors to the template.
5. You can edit one or more monitors, update settings per monitor, and more. See Modify templates, application monitors, and component monitors in SAM for details. To review help information per monitor, use the help option in the Orion Web Console.
6. When done, click Submit to save the changes.

Add Component Monitors

When editing a template, you can add one or more component monitors.

1. You have two options for adding component monitors. This scenario uses the Manually Add Component Monitor option.

   - Manually Add Component Monitors allows you to select and add multiple types of component monitors to the template
   - Browse for Component Monitors opens the Component Monitor Wizard to add multiple instances of a selected component monitor

For details on each monitor, see Component Monitor Types.
2. When added, you can modify the configurations and custom settings per monitor.

3. Click the checkbox for a component monitor, click in the Quantity field to add multiples, and click Add.

4. Expand each component monitor to configure settings, add scripts, and more.

5. When completed, click Submit.

   Click Save and Continue Working as you add and complete component monitors.

Copy a template

You can modify current templates, or create a copy. With copies, you can use a base template from the default templates, imported templates, or templates you created. Using a copy for a new template can make the process much faster than starting a new template without pre-filled monitors and configurations.

The copied templates use the same name of the original name with "- Copy" appended.

   If you want to completely modify a current template, use a copy to keep the original.

1. Click Settings > All Settings > SAM Settings > Manage Templates.

2. Select the application template you want to copy, and click Copy. A new template is added with all of the application and component monitors, including any configured settings.

Export and import templates

Templates can be exported and imported to share and use between all Orion users. Exporting templates creates a file you can share on thwack with other users.
Ensure the imported file format is XML with .apm-template extension. Unzip templates before importing.

To export a template to THWACK:

> Templates can take about 5 minutes to export to THWACK. If exporting multiple templates, give SAM time to complete the export.

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Browse or select for a template in the list.
3. Click Import Export > Export to THWACK.
4. Enter your THWACK account credentials when prompted.
   If you need an account, click Create Account and follow the steps.

To export to a file:

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Browse or select for a template in the list.
3. Click Import Export > Export as File, and click Save.
   The template packages into a file with an extension .apm-template. You can also post a zip file of templates if needed.
4. To share the template, visit the Application Monitor Templates folder on THWACK.
5. Click Write a document, add a title and description for the template, and attach the .apm-template file.
6. Add apm_monitoring_template in the Tags field.
7. Select Application Monitor Templates in Categories.
8. Click Publish to post the information and template.
To Import templates from THWACK:

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Click the Shared Templates on THWACK tab. A page opens and populates with available templates from THWACK.
3. Search or browse the templates. Select a template file and click Import. To view information per template from the authors, click the link for [content exchange area on THWACK](#).
4. Enter your THWACK account credentials when prompted.

5. The template imports into the Orion Web Console. Search or browse to locate and edit the template as needed.

   If you import a template with the same name as one of your existing templates, the name of the imported template is modified by appending (n) to the name, where n is an integer.

   If you receive an invalid file error, check the file format. The template should be XML code with .apm-template extension. If you are importing a ZIP, unzip the contents then import the template file.

**Import templates from a file**

You can download templates and import templates from your local computer using the Import option. You can locate these [templates on THWACK](#). If the template is in a .zip file, extract the zip and verify it is XML and has an extension of .apm-template. If not, you will receive an invalid file error.

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Click Import/Export, and click Import. A page opens to browse and select the file.
3. Browse and select the template file from your local drive.
4. Click Submit. SAM checks the file type and format.
5. The template imports into the Orion Web Console. Search or browse to locate and edit the template as needed.

If you import a template with the same name as one of your existing templates, the name of the imported template is modified by appending (n) to the name, where n is an integer.

Delete templates

Deleting a template also deletes all of its assigned applications, both modified and unmodified. To speed up user interface interaction, data is not immediately removed from the database, but systematically updated every few minutes in the background.

Instead of deleting a template, consider exporting it to a file so you can restore it later, if necessary.

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Check the template(s) to delete and then click Delete.
3. Confirm deletion by clicking Yes.

Tag a template

Tags are descriptive labels that help you classify and sort your application templates on the Manage Application Monitor Templates page. The application templates included in SolarWinds SAM have already been tagged with several descriptive labels you can modify as you see fit.

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Select the templates you want to tag, then click Tags.
3. Click Add existing tag(s) or select the tags from the list.
4. Type the tags, separating multiple tag entries with commas, then click Submit.

To remove tags:

1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Select the templates you want to tag, then click Tags > Remove Tags.
3. Select the tags from the list, then click Submit.

Change between 32-bit and 64-bit polling

You should use 64-bit polling on 64-bit OS systems. Using AppInsight applications with 32-bit polling on 64-bit computers via an agent prevents certain performance counters from collecting information.

You may need to log in with an administrator account to perform these changes.

To change to 64-bit polling at the application resource level:

1. Click My Dashboards > Applications > SAM Summary.
2. Select an AppInsight Application (Exchange, SQL, IIS) and then click Edit Application Monitor.
3. Expand Advanced, and then click Override Template.
4. In the Platform to run polling job field, change the value to x64.
5. Click Submit.

To change to 64-bit polling at the template level:
1. Click Settings > All Settings > SAM Settings > Manage Templates.
2. Select an ApplInsight application and click Edit.
3. Expand Advanced and change the Platform setting to x64.

Best practices for SAM templates

SAM template creation includes more than just adding and configuring component monitors. Use these best practices, tips, and tricks about performance enhancements, testing, and scripting to create, customize, and monitor component monitors and templates.

Performance enhancements

**Modify the polling frequency for performance**

Depending on the length of calls and amount of data pulled for a monitor, you may want to modify the frequency. For script monitors you may need to only run the script once per day or once per week. For example, to compare MIBs using the SolarWinds MIB Database template, you may only need to run the comparison once a day or week.
Extend the polling timeout for long calls

For scripts with lengthy calls for large amounts of data, extend the polling timeout. The default 300 seconds may not be long enough for script processing to complete. If the call may take more time, especially during peak times, increase the timeout to give the system time to complete the call. For example, for MIB database comparison scripts using the SolarWinds MIB Database template, multiple files are called, downloaded, and compared to return status messages and complete specific actions.

Enhance latency and performance by pulling multiple metrics per template

When executing script component monitors in a template, SAM affects performance and latency making calls to a target server. Complete calls for up to 10 metrics per script to reduce the number of calls, increasing performance. Depending on the size and processing of scripts, balance scripts and lengthy calls across multiple instances of a script monitor.

Script, monitor, and template testing

Check credentials and server permissions for scripts

Verify you have the correct credentials with assigned account permissions to execute scripts on the Orion Web Console and target server. Issues with scripts tend to be with credentials. The script monitors may provide fields for credentials, or you may need to provide credentials in the script code, arguments, or command line. Test the script in SAM prior to verify credentials and access.

Test scripts before monitoring

When adding and configuring script component monitors, you need to test the script. When the test completes, SAM registers each returned metric as a numbered output in the Orion SQL Database. You can configure the display of collected metrics and values through the component monitor. Each script monitor supports up to 10 different outputs.

Receive accurate node status

Until tested, scripts and component monitors return an initial unknown status. After testing, polling returns accurate application status.

Script best practices

Use code comments

Code comments help document the intent for code, decisions made, and to track changes. SolarWinds recommends using code comments to keep detailed steps and responses in your code. If additional administrators need to work in the script monitors, the comments provide context for the code.

```
# for a comment per line.

<#
For lengthy comments per code section.
#>
```
Do not use positional parametrization

In the command line for executing scripts, always add the parameter per value. Do not assume the position of data in the command dictates the parameter. For example, use -h for hostname.

Use a header for writing multiple scripts

Create a header in your code to reuse throughout your scripts. The header could include example code and code comments for:

- A listed of exit codes
- Set variables for return metrics commonly used in your scripts
- Use code to determine if you are testing code on the target server or the Orion system

For example, the following PowerShell code returns a message identifying if the server is a test system or the Orion server:

```powershell
if ( $env:COMPUTERNAME -like "^SWO=" )
{
    Write-Host "Message: Running on an Orion Server"
    Write-Host "Statistic: 1"
    $OrionServer = $true
    $ServerIP = '${IP}'
}
else
{
    Write-Host "Message: NOT running on an Orion Server"
    Write-Host "Statistic: 0"
    $OrionServer = $false
    $ServerIP = '192.168.0.1'
}
if ( $OrionServer )
{
    exit 0
}
```

Additionally, you could add a step to save the code if not on the Orion server.

Use SolarWinds macros

When using SolarWinds macros, consider assigning them to named variables in your scripts.

The following SolarWinds macros are available for Linux/Unix, Nagios, Windows Script, and PowerShell script monitors:

- `${USER}`
- `${PASSWORD}`
- `${PORT}`
- `${Node.SysName}`
- `${Node.Caption}`
- `${Node.DNS}` - Use this instead of `${IP}`.
- `${Node.ID}`
- `${Component.ID}`
- `${Component.Name}`
- `${Application.Id}`
- `${Application.Name}`
For agent monitored nodes, use the macros \${Node.SysName} and \${Node.DNS}. The \${IP} may return a loopback IP before polling starts.

**Report status through exit codes**

Scripts must report their status by exiting with the appropriate exit code. The exit code is used to report the status of the monitor, which is seen by the user through the interface.

A script should return an exit code which results in an Up (0), Warning (2), or Critical (3) status. When one of these exit codes is received the appropriate dynamic evidence table structure is created and all further exit codes are handled correctly. If the component only returns Down (1) or Unknown (4) on first use, the appropriate dynamic evidence table structure is not created appropriately.

⚠️ You must test the component monitor after entering the script to properly calibrate the monitor, generate tables, and verify correct communication between the target node, SAM, and the template.

- 0 - Up
- 1 - Down
- 2 - Warning
- 3 - Critical
- Any other value - Unknown, for example 4

**Multiple options for returning exit code and message**

You can return one of multiple options for exit codes and messages using IF/ELSE or case statements for your scripts.

**Use error trapping to capture issues**

Using error trapping code such as try/catch blocks help capture and report errors. These blocks provide better reporting of an error with detailed information for the issue.

**Import and export SAM templates**

SAM integrates with the SolarWinds online IT community site, THWACK, so SAM customers can share templates. You can export custom templates created for applications and custom scripts to THWACK to help other customers. Exporting templates creates an XML file with an .apm-template extension and adds the template to a repository accessed by SAM.

SAM automatically connects with THWACK to list all available templates for import. You need a THWACK account to download and install templates into SAM.
Please note, content posted by other community members in THWACK is not part of the SolarWinds software or documentation that you purchased from SolarWinds. Your organization should internally review and assess to what extent, if any, such custom scripts or recommendations will be incorporated into your environment. You elect to use third party content at your own risk, and you will be solely responsible for the incorporation of the same, if any.

Export a template

Templates can take about 5 minutes to export to THWACK. If exporting multiple templates, allow time for SAM to complete the export.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Browse or select for a template in the list.
3. Click Import Export > Export to THWACK.

4. Enter your THWACK account credentials when prompted.

If you need an account, click Create Account and follow the steps.

A message displays with the template exporting to THWACK. The template packages into an XML file with an extension .apm-template.
Import a template

1. In the Orion Web Console, click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Click the Shared Templates on THWACK tab.
3. Search or browse the templates. Select a template file and click Import.

To view details about a template, click the link for the content exchange area on THWACK.

4. Enter your THWACK account credentials when prompted.
   If you need an account, click Create Account and follow the steps.

5. The template imports into the Orion Web Console. In the dialog box, click View Imported Templates. 
   A list displays of all imported templates.
Understand the Credentials Library

You typically need to associate credentials with component monitors so they can retrieve data from target servers. For example, to use a WMI monitor, you must provide valid domain or computer credentials. Or, if your web server requires credentials, you must provide the appropriate credentials to access the protected sections of your site. Those credentials can be stored in the Credentials Library.

If each component in an application monitor requires a separate credential, you cannot add them in Add New Application Monitor wizard. Instead, edit the application monitor after creating it to configure the credentials of each component separately.

Note the following details about credentials:

- Credentials may be used several times if you Use Application Discovery to scan nodes and apply templates automatically. If credentials are incorrect, the account may become locked out. To avoid lockouts that affect actual users, consider using service accounts for monitoring.
- If SNMP credentials were provided for a node during Discovery, you do not need to specify additional credentials for SNMP operations. To learn more, see SNMP credentials for the Orion Platform.
- Credentials in this library can also be used with the "Inherit credentials" option for nodes and templates.

The Credentials Library differs from the Certificate Credentials Library, which stores certificate details for SSH keys, including user name, private key, and key type. See Store SSH keys in the Certificate Credentials Library.

Manage credentials in the Credentials Library

If you have administrator rights, you can use SAM's Credentials Library to manage credentials that are associated and used with component monitors to retrieve data from servers and services.

Add credentials to the Credentials Library

For a video overview, see Setting Credentials in SAM.

1. Click Settings > All Settings > SAM Settings > Credentials Library > Add New Credential.
2. Provide a name for the credential set. SAM will display this name in the Credential for Monitoring field of monitors that accept credentials.
3. Provide the user name and password, and then confirm the password and click Submit. If providing Windows credentials to access information through WMI, be sure to provide the account name in the following syntax: domainOrComputerName\user name for domain level authentication or User Name for workgroup level authentication.

You can assign credentials to all the associated components of a template or application monitor.
**Edit credentials in the Credentials Library**

1. Click Settings > All Settings > SAM Settings > Credentials Library.
2. Click Edit for the desired credential.
3. Modify the information as needed and then click Submit.

**Delete credentials from the Credentials Library**

1. Click Settings > All Settings > SAM Settings > Credentials Library.
2. Click Delete for the desired credential.
3. Click OK to confirm the deletion.

If you delete a credential set, be sure to update any monitors that were using the credentials to use a different credential set.

**Store SSH keys in the Certificate Credentials Library**

Typically, you must associate credentials with component monitors and templates to enable them to retrieve application data. For added security, SAM also includes a Certificate Credential Library where you can store certificate details for Secure Shell (SSH) keys required for script monitoring on SAM nodes, including:

- User Name
- Private Key: Upload a private key file or paste the private key in PEM format.
- Key Type: RSA or DSA
- Password (optional)

Certificates can be used for authentication with Linux devices monitored in SAM. Linux, Unix, and Nagios script monitors also support certificate based authentication.

The Certificate Credentials Library differs from the Credentials Library that stores standard authentication credentials for component monitors. For example, a WMI component monitor may need to run as a particular user (or service account) to collect information. See [Understand the Credentials Library](#) and the [Setting Credentials in SAM](#) video.

**To access the Certificate Credentials Library:**

1. Click Settings > All Settings.
2. Under Product Specific Settings, click SAM Settings.
3. Click Certificate Credentials Library.

**Assign certificate credentials**

There are two ways to assign Certificate Credentials — when assigning a template to a node or when editing a template directly. Before you begin, choose the right method

- If every node uses unique private keys, then editing the application after it is assigned is the best option.
If most or all of your nodes use the same private key, then you should edit the credentials directly in the template.

You'll be prompted to provide the following details for each certificate credentials:

- **Credential Name:** User-defined text that identifies the credential for later use in templates.
- **User Name:** The user who is associated with the public key certificate on the target computer.
- **Key:** Text content of the private certificate file in Privacy Enhanced Mail (PEM) format.
- **Key type:** The algorithm the certificate used to generate the certificate pair. (This can be found as part of the header. For example, e.g. "--- BEGIN RSA PRIVATE KEY---.")
- **Key password:** The password used to protect the certificate file

**To assign certificate credentials when assigning a template to a node:**

1. Assign a template to a node.
2. When asked to choose credentials, select the "Inherit credentials" from template option.
3. Click Assign Application Monitors and then edit the template by clicking Edit next to the template name.

4. From here, you can select one or more Linux/Unix/Nagios script component monitors to edit by checking the boxes to the left of each monitor and then clicking Multi-Edit.
5. Check the Authentication Type box and select User name and PrivateKey from the drop-down menu.
6. Check the Credential for Monitoring box and select the credentials you want to use from the drop-down menu, and then click Save.

**To assign certificate credentials when editing a template directly:**

1. Select the template you want to edit.
2. Change the Authentication Type option to User name and PrivateKey for each component monitor that should use this authentication type.
3. From the Credential for Monitoring field drop-down menu, select the credentials you want to use with your monitors, and then click Submit.

**Share application monitor templates on THWACK**

If you create a SAM template that you think might be useful to other SolarWinds users, you can share it via our online IT community, THWACK. You can also import templates planned for upcoming versions of SAM, or shared by other SAM administrators.
Want THWACK points? Upload a template to earn 50 THWACK points with an additional five points awarded every time you download a template. The THWACK Store offers lots of nifty swag.

Note the following details about THWACK templates:

- A THWACK account is required to export and import templates into SAM. To create one now, navigate to https://thwack.solarwinds.com.
- SolarWinds often posts SAM templates being developed for a future version of SAM to get customer feedback. While SolarWinds Support does not provide direct support for in-development templates, our product owners monitor these posts to answer questions and read your recommendations.
- SolarWinds does not provide direct support for templates posted by third parties. If you have questions or suggestions for material posted by a member of the THWACK community, use the forum's built-in messaging functionality.
- To learn about accessing THWACK templates via a proxy, click here.

Check your organization's policy about sharing company information publicly before posting to THWACK.

**Disclaimer:** Please note, any custom scripts or other content posted herein are provided as a suggestion or recommendation to you for your internal use. This is not part of the SolarWinds software that you have purchased from SolarWinds, and the information set forth herein may come from third party customers. Your organization should internally review and assess to what extent, if any, such custom scripts or recommendations will be incorporated into your environment. Any custom scripts obtained herein are provided to you “AS IS” without indemnification, support, or warranty of any kind, express or implied. You elect to utilize the custom scripts at your own risk, and you will be solely responsible for the incorporation of the same, if any.

Export a template

You can export custom templates you create for applications and custom scripts to THWACK to help other customers. Exporting templates creates an XML file with an .apm-template extension and adds the template to a repository accessed by SAM.

Templates can take about 5 minutes to export to THWACK. If exporting multiple templates, give SAM time to complete the export.

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Browse or select for a template in the list.
3. Click Import Export > Export to THWACK.
4. Enter your THWACK account credentials when prompted. SAM exports the template to THWACK as an XML file with an .apm extension.

Import a template

SAM automatically connects with THWACK to list all available templates for import. Browse and review the descriptions of available templates to find exactly the ones you need for your monitored applications.

1. Click Settings > All Settings > SAM Settings, and click Manage Templates.
2. Click the Shared Templates on THWACK tab.
   A page opens and populates with templates posted in THWACK.
3. Search or browse the templates. Select a template file and click Import.
   To view any details provided by the author, click the link for content exchange area on THWACK.

4. Enter your THWACK account credentials and click Log In.
5. After the template is uploaded, click View Imported Templates.

   If you import a template with the same name as an existing template, the name of the imported template is modified by appending (n) to the name, where n is an integer.
Manage templates and groups

You can assign templates and application monitors to groups of nodes. As you add nodes to the group, templates and applications linked to the group are automatically added to the new nodes. Likewise, if you add a new template or application monitor to the group, all node members receive the new additions.

The group assignments have the following logic:

- Added nodes receive the assigned templates and application monitors, regardless of template type. For example, if a template assigned to a group monitors a specific OS version such as Windows Server 2012, all nodes in the group are assigned the Windows Server 2012 template, regardless of the OS used by each node.
- Nested groups are not supported. AppInsight applications cannot be assigned to groups. Application assignment to a group happens in the background and can take some time to be created depending on the size of the application.
- If an OS-specific template, such as the Apache (Windows) template, is assigned to a group with Linux nodes, the template is assigned and reports a status of Unknown due to OS mismatch.
- If a node added to the group already has the template assigned to it, no changes occur.
- If a node is part of two groups with shared templates, the node only receives one copy of each assigned template and application monitors.
- If you delete a group, the assigned templates may remain assigned depending on the Advanced section configuration. In the Auto Delete Application Monitor option, select Yes to remove the templates if the group is deleted or nodes are removed from the group.

![Advanced settings for template assignment](image)

- You cannot assign credentials to a node and have all templates assigned to the group use those credentials. Credentials are not associated by group. You can select Inherit Credentials from Node when using WMI or an agent. The new application created from the template will use credentials from the node that is assigned to.

See also:

- Assign an application template to a group
- Unassign application monitors from a group

Assign an application template to a group

Assigning an application template to a group is more efficient than applying the template to individual nodes one at a time. Before you begin, create at least one group, as described in the Orion Platform Administrator Guide.
1. Click Settings > All Settings > SAM Settings > Manage Templates > Application Monitor Templates.

2. Select an application template and then click, Assign to Group.

3. Select a group from the Available Groups column, click the green arrow, and then click Next.

   By default, the template is assigned to those nodes in the group that are running a server operating system. This can be changed in the Advanced section at the bottom of the page. When a node is removed from a group, the application is deleted from the node by default. You can override this by selecting No under the Advanced section.

4. Select your credentials and then click, Assign Groups.

5. Click Done.

   It may take 10 minutes or so for assignments to be processed. If it takes longer, you may need to restart services.

Your application templates are now assigned to a group.
To view the applications within a group, click My Dashboards > SAM Summary and then click a group in the Application Templates Assigned to Group widget.

You can create dynamic groups and assign templates to these groups. For example, create a dynamic group for all Windows computers and the Windows Update template will be applied to the nodes of the group dynamically.

Create a template for an application

SolarWinds SAM provides pre-built templates that allow you to monitor over 200 applications. You can use the pre-built templates as is, you can modify them, or you can create a template.

Creating a template allows you to monitor uncommon components or applications that are not included in the pre-built application templates provided with SolarWinds SAM.

You have two options for creating templates:

- Create a new template and add component monitors as needed
- Create a copy of an existing template

The following example shows how to create and assign a template that contains the file size component monitor. You can add as many component monitors to a template as you need.

Create a template to monitor file size

1. In the Orion Web Console, click Settings > All Settings.
2. In the Product specific settings group, click SAM Settings.
3. Click Create a New Template, and enter a name.
4. Click Add Component Monitors > Manually add Component Monitors.
5. Select the File Size Monitor component, and click Add.

6. Enter the path and the file to monitor.

7. Enter warning and critical threshold values, and click Submit.

Apply the template to a node

1. In the Orion Web Console, click Settings > All Settings.
2. In the Product specific settings group, click SAM Settings.
3. Click Manage Templates, select the template, and click Assign to Node.

![Create New Template](image1)

4. Select the node from the left pane, click the green arrow to move it to the right pane, and click Next.

![Available Nodes](image2)

5. Choose the credentials, click Assign Application Monitor, and click Done.

Unassign application monitors from a group

1. Click Settings > All Settings > SAM Settings > Manage Templates > Application Monitor Templates.
2. Select the application monitor you want to unassign from the group, and then click the link in the Assigned To column.

3. On the Template Assignments page, click the Group tab and then click Unassign > Yes, Unassign.

4. Click Close.

Work with component monitors in SAM templates

Component monitors are the building blocks of SAM templates. Each template consists of multiple component monitors designed to poll for specific metrics about an application, process, or event.

For example, here are two component monitors included in the Active Directory 2016 Domain Controller Security template:

- The "Locked out users" monitor utilizes a PowerShell script to pull metrics for currently locked out users.
The "User Account: User account was created" monitor scans Windows event logs for events that match certain criteria (for example, events that contain a specific keyword).

Component monitors like these two are designed to poll for application data — in this case, Active Directory — so you may also hear them called "application monitors."

Some component monitors pull hardware-related metrics. For example, the OOTB Apache template includes a "ServerUptime" monitor that uses SSH to upload a script to a Linux/Unix server and retrieve server uptime metrics for an Apache server.

For details on application templates and monitors, see SAM Component Monitor Types and

- Modify templates, application monitors, and component monitors in SAM
- Linux/Unix system configurations for component monitors
- Assign application monitors to nodes
- Create component monitors in the Component Monitor Wizard
- Unmanage and manage assigned application monitors
- Manage templates and groups
- Create custom component script monitors
  - Use Windows Script monitors
  - Linux/Unix Script monitor
  - Nagios Script monitor
  - Create a Windows PowerShell monitor
- Understand application monitor thresholds
- Example tasks for application monitors

For details on application templates and monitors, see SAM Component Monitor Types and the SAM Template Reference.

Component Monitors

Component monitors are the building blocks of SolarWinds SAM. Each monitors the status and performance of a different aspect of an application. There are several different types of component monitors, each containing settings that define what is monitored and how to monitor it.

Some types of component monitors allow you to set threshold conditions on the monitored parameters. You can set separate thresholds to indicate warning and critical conditions. For example, if you are monitoring the percentage of free space remaining on a volume, you can set a warning threshold at 15%, and a critical condition at 5%.

As an analogy, pretend SolarWinds SAM is monitoring a car. You would have component monitors to check tire pressure, engine RPM, water temperature, battery voltage, and other important subsystems of that vehicle. You can set alerts to give notification if the water gets too hot, or if the battery voltage drops too low.
Application Monitor Templates

A template is a group of component monitors modeling the total availability and performance level of an application. A complicated application such as Windows Server may require dozens of component monitors to accurately assess its current status and performance.

Instead of creating component monitors one-by-one for every application server, you can assign a pre-made template. The template can either be one included with SolarWinds SAM, or a custom template you make yourself. For example, you can assign the included Microsoft Windows Server 2003-2008 template to your Windows 2003 and Windows 2008 computers and obtain vital statistics on all of them.

A template is only a blueprint and does not perform any monitoring on its own. Only after assigning the template to a server node are active assigned component monitors created.

To continue the car analogy, pretend you want to monitor a fleet of 50, 2010, blue Dodge Charger automobiles. Instead of defining the component monitors for 50 cars, you can define all the component monitors in a Dodge Charger template.

Assigned Component Monitors

Assigned component monitors are created by assigning Application Monitor templates to server nodes. Each actively monitors its assigned node according to its settings. Component monitors inherit these initial settings from the template. If you make a change to a template, that same change is rolled out to all assigned Application Monitors based on the template.

You can override the template settings at any time, breaking the inheritance relationship between the component monitor and its template. For example, the user name and password usually differ for each node, and you would select a different credential for each assigned Application Monitor, thus overriding the template setting for the Credentials field.

To restore the inheritance relationship between a component monitor and its template, click Inherit From Template next to the setting.

Continuing the car analogy, when you assign the Dodge Charger template to a Dodge Charger vehicle, you now have a set of assigned component monitors for monitoring the vehicle's tire pressures, engine RPM, and so forth.

Assigned Application Monitors

An assigned Application Monitor runs its assigned component monitors at regular intervals, and then uses the status results from the component monitors to determine an overall status for the application.

If some of the component monitors are up and others are down, the Application Monitor follows the Status Rollup Mode setting in the Orion Web Console Settings to show either the worst status of the group or a warning status.

The difference between an assigned Application Monitor and a template is that the template is only a blueprint and does not perform any monitoring on its own. Only after assigning the template to a server node does SolarWinds SAM conduct any actual monitoring on the node.
To complete the car example, you assign the Dodge Charger template to all the Dodge Charger vehicles to create the assigned Application Monitor for determining the overall status for your Dodge Charger fleet. For example, the fleet may be 95% available at a given time due to warnings for some of the cars.

The following diagram illustrates the work flow involved in creating an application to be monitored by SAM.
Report status through exit codes

Scripts in component monitors must report their status by exiting with the appropriate exit code. The exit code is used to report the status of the monitor, which is seen by the user through the interface.

A script should return an exit code which results in an Up (0), Warning (2), or Critical (3) status. When one of these exit codes is received the appropriate dynamic evidence table structure is created and all further exit codes are handled correctly. If the component only returns Down (1) or Unknown (4) on first use, the appropriate dynamic evidence table structure is not created appropriately.

<table>
<thead>
<tr>
<th>EXIT CODE</th>
<th>SERVICE STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Up</td>
</tr>
<tr>
<td>1</td>
<td>Down</td>
</tr>
<tr>
<td>2</td>
<td>Warning</td>
</tr>
<tr>
<td>3</td>
<td>Critical</td>
</tr>
<tr>
<td>Any other value</td>
<td>Unknown, for example 4</td>
</tr>
</tbody>
</table>

The following code snippet highlights proper usage of exit codes.

```powershell
if ($Error.Count -eq 0) {
    Write-Host "Message: User $address received: $stat items during last month";
    Write-Host "Statistic: $stat";
    Exit 0;
}
Write-Host "Message: $($Error[0])";
Exit 1;
```

The two exit codes in this example are conditional, meaning either one or the other will be triggered based on a certain outcome.

- When Exit 0; (status of Up) is reported, the message and statistic are displayed and the monitor shows a status of Up.
- When Exit 1; (status of Down) is reported, the message and statistic are not displayed and a status of Down is reported.

For example, if you want to inform SolarWinds SAM that a PowerShell script reports an Up status, you would exit the script using Exit 0.

Assign application monitors to nodes

To begin collecting data in SAM, you need to assign the application monitors or templates to nodes managed by the Orion Platform. The quickest way to assign Application Monitors to nodes is through the Add New Application Monitors Wizard. You can also assign them through the Manage Templates page.

ℹ️ You may need to log in with an administrator account to perform this action.
To assign a template using the Add New Application Monitors Wizard:

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manually Assign Application Monitors.
2. Select the Application Monitor template to apply, and then click Next.
3. Select the server nodes to which you want to apply the Application Monitor template, and then click Next.
4. Select existing credentials or create new credentials, then click Assign Application Monitors.

To assign a template through the Manage Application Monitor Templates page:

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Browse and select the template you want to assign.
3. Click Assign to Node, then select the server nodes to which you want to apply the Application Monitor template, and then click Next.
4. Select existing credentials or create new credentials, then click Assign Application Monitors.
5. Review the information for the assigned Application Monitor and then click Done.

To remove application monitors:

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Select a template and click Edit. The template opens displaying all currently assigned monitors.
3. Select the monitors you want to remove and click Delete.

Scan nodes for applications to apply templates automatically in SAM

SolarWinds SAM can scan nodes for you and automatically assign the application monitors it deems suitable for each scanned node. You select the nodes to be scanned, the application templates to use in the scan, and the scanning parameters to determine a match. When the Application Discovery wizard runs, SAM compares applications located on nodes with the parameters to automatically assign monitors.

You may need to log in with an administrator account to perform these actions.

You cannot scan for user experience (UX) monitors, but you can assign them to nodes manually. Adding monitors from this page does not affect your scan.

To scan nodes, discover applications, and assign monitors:

1. Click Settings > All Settings > SAM Settings > Click Scan Nodes for Applications.
2. Browse or filter to select nodes to scan. You can select all or pick and choose from lists. The number of nodes selected lists on the page. Click Next.
3. Browse, filter, and select the applications you want to scan for. These applications have associated monitors. Expand the Advanced Scan Settings to set the exactness for matches. Click Next.

To minimize scan time, limit the number of application templates in your first scan. To see more application templates, select a different template group from the Show Only list.
• Exact Match: All the components must match to assign the template.
• Strong Match: Most of the components must match to assign the template.
• Partial Match: Some of the components must match to assign the template.
• Minimal Match: At least one component must match to assign the template.

4. Some application templates require credentials either to access restricted resources, or to run within the context of a specific user. To scan for these templates, add the necessary credentials to the list, moving them into an order for using. If a template you are scanning for requires credentials, the credentials in this list are tried in the order in which they appear. You can add credentials or allow credentials to inherit from the node’s local credentials. Click Next.

If you have domains sharing user names with different passwords, we recommend you run separate application discoveries for each domain.

Credentials are tried several times over the course of a scan. If the credentials are incorrect, the account can become locked due to the amount of failures. To avoid potential account lockouts that affect actual users, use service accounts created specifically for use during monitoring. With service accounts, no actual user is affected by an account lockout if a password is entered incorrectly.

5. Review the selected options before scanning. If the automatic discovery matches templates already assigned to the node, the template is not automatically assigned a second time. To assign duplicate templates, select Yes, Assign Anyway from the Do you want to assign duplicates list.

6. Click Start Scan to start the discovery and assignments.

7. The scan runs in the background. A message appears at the top of the Orion Web Console when scanning is complete. Click View results to see the results of the scan.

You can modify the assigned applications monitors via the Manage Application Monitors page, accessible by clicking Settings > All Settings > SAM Settings > Manage Application Monitors.

Unmanage and manage assigned application monitors

You may need to unmanage or manage assigned application monitors. When a template of application monitors is unmanaged, you no longer receive data in SAM. To resume viewing status and polled data, you need to manage the application monitors template.

You cannot unmanage specific application monitors within a template. Monitors have an enable/disable option.

You may need to log in with an administrator account to perform this action.

1. Click Settings > All Settings > SAM Settings > Manage Application Monitors.
2. Locate the application monitor template you want to manage or unmanage.
3. Select an option: Manage or Unmanage.
   - If managing, the application monitor template immediately begins polling per the configured intervals.
   - If unmanaging, a scheduler opens to set the date and time to begin and end the unmanaged state. Click OK to save.

Modify templates, application monitors, and component monitors in SAM

Through the Manage Application Templates page, you can modify default templates and create new templates that consist of one or more application or component monitors. To see a list of the monitors in a template, select the template on the Manage Application Monitor Templates page and click Edit.

For a list of templates included with SAM, see the SAM Template Reference.

You can edit monitors individually, edit multiple monitors of the same type simultaneously, as well as disable or delete monitors in a template. To modify the specific configuration of a monitor, edit it individually.

You may need to log in with an administrator account to perform these actions.

Multi-edit components

You can simultaneously edit multiple component monitors within a template. You can select multiple monitors using the checkboxes to modify, without needing to be of the same type. To reorder the monitors, drag and drop them in the table view. You can perform the following types of multi-edit options:

- Multi-Edit: A prompt displays with editing options specific to the type of monitors selected. For example, the Fetching Method for the three Services selected can be changed from RPC to WMI.

  Different types of monitors have different options available for editing.

- Assign Credentials: Modify the credentials as needed for the selected monitors.
- Test: Run a communication test on the monitors using agent or agentless communication as configured.
- Set Test Node: Modify the test node for communication tests.
- Disable/Enable: Start and stop polling for the monitor.
- Delete: Permanently removes the monitor.

When editing a template, the green arrows to the right of each component monitor, highlighted below, lets you to change the order of each monitor. This new order is respected only on the All Applications widget and the Application Details widget.
Delete or disable a monitor

You can disable or delete application monitors as needed. When you disable, the monitor remains in the SAM without communicating and collecting data. You can disable one or monitors.

- If you want to stop all monitoring using the monitor, click Disable.
- If you want to monitor using the monitor again, select and click Enable.

If you no longer want the monitor in the template, you can delete to permanently remove it from SAM.

1. When editing a template, select the monitor you want to remove.
2. Confirm by clicking Yes. The monitor is permanently removed from the template.

Create component monitors in the Component Monitor Wizard

The Component Monitor Wizard provides a starting point where you can select component monitors based on a specific process, performance counter, or service. You can add these monitors with modifications to application monitors and templates for assigned node monitoring. You can then discover and monitor nodes that match the selected process.

Component monitors are a part of the larger whole of application monitors and templates used to monitor applications. Templates include multiple application monitors, which include multiple component monitors, for tracking and providing data on nodes and applications. The application template can then be applied to nodes running the application the template was designed for. To learn how to create an application template, see Assign application monitors to nodes.

You may need to log in with an administrator account to perform this action.

1. Click > Settings > All Settings > SAM Settings > Component Monitor Wizard.
2. Select a component monitor type to add to your application or template. These include monitors based on operating system: Windows Systems, Linux - Unix Systems, VMWare Systems, and Applications.

When using the Component Monitor Wizard, the only component monitor supported for Orion Agent for Linux is the Linux - Unix System: Process Monitor. To create and assign component monitors and templates for Orion Agent for Linux, see the list in Monitor with Orion agents in SAM.
3. Depending on the component monitor you select, a new set of steps display. For example, if you create a Process Monitor, you should select a Target system by browsing to supported systems in your environment. You may also need to select a platform type and enter credentials for the target system. Select and enter all required information.

4. When you click next, if you entered credentials, a connection test runs. If it passes, continue adding component monitors and properties. Depending on the components, you may need to enter additional server and credentials data.

5. With your component monitors set, you can add the component monitor to other application monitors and templates.

6. Finally, assign the application monitor or template to nodes in your environment. When confirmed, the node details page will include these new monitors and resulting data.

Linux/Unix system configurations for component monitors

To effectively monitor Linux/Unix systems with the Orion agent for Linux/Unix, consider and complete the following configurations for component monitor and templates. You may need to reference vendor documentation, as well as work with your NOC and server management groups to complete modifications.

- Apache configurations
- Tomcat configurations
- JMX
- Squid
- Nagios
- ODBC configurations
- PostgreSQL
- MySQL
- IBM DB2 with ODBC

Apache configurations

You should have the following packages installed:

- Apache httpd
- LWP::UserAgent (Perl)

To install and configure:

1. Log into your Apache server using SSH or telnet client. Grant yourself root permissions (so root).
2. Use the command: `yum install perl-libwww-perl`
3. Locate `httpd.conf`: `find /etc -iname httpd.conf`. The file is typically located in `/etc/httpd/conf/httpd.conf`. 
4. Locate the following section:

```html
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from .example.com
</Location>
```

5. Uncomment the code. Edit the Allow from rule, substituting the IP address or hostname of your Apache server for localhost. Use localhost only if the Apache server services the loopback interface. Also make sure the ExtendedStatus is On.

```html
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from localhost
</Location>
ExtendedStatus On
```

6. Restart the Apache httpd: service httpd restart
   or
   Reload the configuration: service httpd reload

Tomcat configurations

Apache Tomcat is an open-source web server, implementing several Java EE specifications.

Follow these steps to install and configure Tomcat servers to work with SAM component monitors. If Apache Tomcat is already installed, you can skip to step 4 to create an account used for polling.

1. Determine the version of Tomcat you want to install. You may have multiple versions available to install. To locate all versions, use this command:

   ```bash
   yum search tomcat
   ```

2. After determining the version, you can install the tomcat package and the admin webapps package. Locate the versions to install from the search results. For example:

   ```bash
   tomcat5.i386 : Apache Servlet/JSP Engine, RI for Servlet 2.4/JSP 2.0 API
   tomcat5-admin-webapps.i386 : The administrative web applications for Jakarta Tomcat
   ```
3. Install the versions using the located versions. To install the example above:

```
yum install tomcat5 tomcat5-admin-webapps
```

4. Next, you need to modify the tomcat-users.xml file to create a user account to access the web manager (‘manager’,‘manager-gui’). To locate, use this command:

```
find /etc -iname tomcat-users.xml
```

5. Edit the file and locate the `<tomcat-users>` section. If commented out, uncomment the configuration. Edit to add a user of `tomcat` with the password of `tomcat`. The template uses this account to access data.

```
<tomcat-users>
  <user name="tomcat" password="tomcat" roles="tomcat,manager,manager-gui" />
  <user name="role1" password="tomcat" roles="role1" />
  <user name="both" password="tomcat" roles="tomcat,role1" />
</tomcat-users>
```

6. Restart the service. The command depends on the version you installed. For example, this command restarts the installed service above:

```
service tomcat5 restart
```
JMX

For JMX configurations, install and configure WebSphere per your Linux/Unix distribution. See the IBM Knowledge Center for details. The overall steps include:

1. Install WebSphere and note the following details:
   - You may need to create an administrator account and a new package group.
   - During installation, turn off the secure storage option as this is not needed.
   - Make sure to select the Application Server during installation.

2. Start and stop the WebSphere AppServer.

3. Use a browser to access the WebSphere AppServer console at http://websphere-host:9060/ibm/console
   - You may need to open a port to 9060 on the Websphere host.
   - Use the credentials for the account created during installation.

4. In the console, expose the JMX platform mbeans in the WebSphere AppServer by:
   a. Clicking Servers > Server Types > WebSphere application servers.
   b. Selecting the listed application server.

5. Locate the Java and Process Management > Process definition. In the Generic JVM arguments dialog box, enter the following code:

   Djavax.management.builder.initial=

   Note: There is nothing after the = sign.

   Dcom.sun.management.jmxremote=true
   Dcom.sun.management.jmxremote.port=8686
   Dcom.sun.management.jmxremote.ssl=false
   Dcom.sun.management.jmxremote.authenticate=false
   Djava.rmi.server.hostname=localhost

6. Restart the AppServer.

Squid

Squid is a caching and forwarding web proxy. It has a wide variety of uses, from speeding up a web server by caching repeated requests; to caching web, DNS and other computer network lookups for a group of people sharing network resources, to aiding security by filtering traffic.

To configure Squid for agent polling:

- SNMP is required - Enable the SNMP feature for Squid. If using Squid-3, this feature is available to enable by adding a configuration option to squid.conf. If using Squid-2, you will need to enable the feature using a series of scripts then configure.
- perl-switch is required - Add per your Linux distribution vendor's documentation.

Review your vendor documentation for details at http://www.squid-cache.org/.
Nagios

Nagios monitors systems, networks, and infrastructure. It offers monitoring and alerting servers, switches, applications, and services. Refer to your vendor’s documentation to properly install and configure Nagios according to your Linux distribution.

You can use the [Nagios Script monitor](#) and [Linux/Unix Script monitor](#) for monitoring Nagios.

If you are using this component monitor with Nagios and polled by Orion agent for Linux, you need to enter a script as follows. The argument would be `python $(SCRIPT)` with a body of:

```python
from random import randint
STAT_PAIRS_COUNT = 10

for stat_id in range(STAT_PAIRS_COUNT):
    print 'Statistic.random%d: %d
Message.random%d: %s' % (stat_id, randint(1,100), stat_id, 'random number: ' + str(randint(1,100)))
```

**ODBC configurations**

To monitor a database with an ODBC connector, install an ODBC driver for your installed database type, create a user, and test access. Use the new account credentials in your SAM template and component monitors. For a full list of available templates, see the [SAM Template Reference](#).

For the following instructions, you should have a database installed such as MySQL or PostgreSQL.

1. (Required) To begin, you should have [unixODBC](#) installed to support Microsoft Windows ODBC. If not, download and install the unixODBC driver to the Linux/Unix-based computer.
2. Install the [DBMS](#) driver.
3. Download and install an ODBC driver for your installed database type.
4. Create a user account for the database.
5. Configure the user access privileges with administrative privileges.
6. Test access to the database using the newly created account credentials.

Configure and assign a SAM template with the credentials.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Search for a template based on the database type such as ODBC or Oracle. SolarWinds recommends creating a copy of the template. Select a template and click Copy.
3. Modify the settings of the template and component monitors based on the metrics you want to monitor.
4. To enter credentials, select the component monitor checkboxes and click Assign Credentials.
5. Enter the credentials you created for the database and click OK.
6. Test the access for the template by assigning it to a node. Click Test Node to assign and test the access. If you encounter issues, verify the credentials and privileges for the account.
PostgreSQL configurations

To monitor a PostgreSQL database server with component monitors, you need to install an ODBC driver, create a user, and test access. Use the new account credentials in your SAM template and component monitors.

SAM provides a PostgreSQL template and component monitors for PostgreSQL 9.

For the following instructions, you should have PostgreSQL installed on a Linux-based computer.

1. Login to the Linux-based computer with administrator privileges.
   You will need to create a user account on the server and in the PostgreSQL database.

2. (Required) To begin, you should have unixODBC installed to support Microsoft Windows ODBC. If not, download and install the unixODBC driver to the Linux-based computer.
   Command: `yum install unixODBC`

3. Download and install the PostgreSQL ODBC driver on the target Linux-based computer.
   Command: `yum install postgresql-odbc`

4. Create a user account for the database.
   The example of commands adds a user account dbuser with the password Password1 to the Linux-based computer. The next commands login as the database superuser to the PostgreSQL server and modify the created user.

   ```
   # adduser dbuser
   # passwd Password1
   # su - postgres
   $ psql -d template1 -U postgres
   ```

   At the prompt, create the dbuser account with the Password1 password:

   ```
   template1=# CREATE USER dbuser WITH PASSWORD 'Password1';
   ```

   Next, grant privileges for your PostgreSQL database (TestDatabase) to the user account:

   ```
   template1=# GRANT ALL PRIVILEGES ON DATABASE TestDatabase to dbuser;
   ```

   Close the template:

   ```
   template1=# \q
   ```

5. Test access to the database using the newly created account credentials.

   ```
   $ su - dbuser
   $ psql -d TestDatabase -U dbuser
   ```

   Configure and assign a SAM template with the credentials.
In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.

Search for the PostgreSQL template. SolarWinds recommends creating a copy of the template. Select the template and click Copy.

Modify the settings of the template and component monitors based on the metrics you want to monitor.

To enter credentials, select the component monitor checkboxes and click Assign Credentials.

Enter the credentials you created for the database and click OK.

Test the access for the template by assigning it to a node. Click Test Node to assign and test the access. If you encounter issues, verify the credentials and privileges for the account.

**Troubleshooting**

If you receive the error "FATAL: Peer authentication method failed for user 'postgresuser'", change the authentication method to md5.

**MySQL configurations**

To configure MySQL, you need to install and configure the appropriate MySQL repository, create a user account, and test access. Use the new account credentials in your SAM template and component monitors.

SAM templates include two options for MySQL 5.7.9 and earlier and 5.7.9 and later.

For the following instructions, you should have MySQL installed on a Linux-based computer.

1. Login to the Linux-based computer with administrator privileges.
   
   You will need to create a user account on the server and in the MySQL database.

2. (Required) To begin, you should have unixODBC installed to support Microsoft Windows ODBC. If not, download and install the unixODBC driver to the Linux-based computer.

3. Install the MySQL ODBC driver.
   
   Command: `yum install mysql-connector-odbc`

4. Create a user account for the database.
   
   The following commands create a MySQL account of dbuser with the password Password1 for the database dbtest. The commands also grant all access to the user account.

   ```
   create database testdb;
   create user 'dbuser'@'localhost' identified by 'Password1';
   grant all on dbtest.* to 'dbuser' identified by 'Password1';
   ```

5. (Required) Grant the following access for local access to monitor through SAM and the Orion Platform.
Enter the IP address of this MySQL Linux-based computer, replacing the example value of 10.100.100.100.

```bash
mysql -uroot -p
CREATE USER 'dbuser'@'%' IDENTIFIED BY 'Password1';
GRANT ALL ON testdb.* TO dbuser@'%' IDENTIFIED BY 'Password1';
GRANT SELECT ON performance_schema.* TO dbuser@'10.100.100.100'
IDENTIFIED BY 'Password1';
```

6. Test access to the database using the newly created account credentials.

   Command: `mysql -u testuser -p`

Configure and assign a SAM template with the credentials.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Search for the MySQL templates. SolarWinds recommends creating a copy of the template. Select a template based on version and click Copy.
3. Modify the settings of the template and component monitors based on the metrics you want to monitor.
4. To enter credentials, select the component monitor checkboxes and click Assign Credentials.
5. Enter the credentials you created for the database and click OK.
6. Test the access for the template by assigning it to a node. Click Test Node to assign and test the access. If you encounter issues, verify the credentials and privileges for the account.

**Troubleshooting**

**Error:** Data source name not found and no default driver specified. This error occurs on a CentOS system.

**Solution:** Verify the CentOS system has the mysql-connector-odbc installed. Use the following command to receive a package and version name on the target Linux-based computer: `rpm -q mysql-connector-odbc`

Verify you have the appropriate credentials assigned for each component monitor.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Search for the SAM template. Select and click Edit.
3. Select the component monitor to check assigned credentials.
4. To modify, click Assign Credentials, enter credentials, and click OK.
5. Test the template.

If you continue to encounter issues, verify the appropriate driver is defined on the CentOS system to match the connection string, as shown in the next example.

```bash
cat /etc/odbcinst.ini
[MySQL]
```
IBM DB2 with ODBC configurations

For IBM DB2, you need to download, install, and configure DB2. (See the [IBM DB2 documentation site](https://www.ibm.com).) SAM provides two IBM DB2 templates: [IBM DB2](https://www.ibm.com) and [IBM DB2 HADR Health](https://www.ibm.com).

Linux-based computers

Here are prerequisites for the following steps:

- IBM DB2 is installed on a Linux-based computer.
- A user account exists on the server and in the IBM DB2 database.

To run commands, you will need the instance users for the database. The instance user connects to the DB and should have the sqllib directory set up.

1. Log into the Linux-based computer with administrator privileges.
2. Use the following command to install the `unixODBC` package:
   
yum -y install unixODBC
3. Update the ODBC config files by adding the following settings to `/etc/odbcinst.ini`:
   
   ```
   [DB2]
   Description = DB2 Driver
   Driver = /opt/ibm/db2/V10.5/lib64/libdb2.so
   FileUsage = 1
   DontDLClose = 1
   ```

   ![Verify that libdb2.so has the correct file path.]

   4. Add the following settings to `/etc/odbc.ini`:
      
      ```
      [dbname]
      Driver = DB2
      ```

      ![The drivers should match in both files.]

   5. Connect locally to the database with the instance user credentials. For example:
      
isql -v dbname db2inst1 Password1
   6. Log into the DB2 database with the instance user credentials and open the DB2 command line tool.
7. Enable the required following setting used by the SAM IBM DB2 template:
   UPDATE DBM CFG USING DFT_MON_BUFPOOL ON
   UPDATE DBM CFG USING DFT_MON_LOCK ON
   UPDATE DBM CFG USING DFT_MON_STMT ON
   UPDATE DBM CFG USING DFT_MON_UOW on

8. Use the GRANT EXECUTE ON FUNCTION statement to grant privileges for the SYSPROC.MON_GET_TABLE function to user, where user is the username that SAM as shown in this example:
   GRANT EXECUTE ON FUNCTION SYSPROC.MON_GET_TABLE TO user
Configure DB2 on AIX devices

Prerequisites

Important: Do not install the 64-bit ODBC drivers; use the 32-bit drivers instead to support Python distributed on the Orion agent.

  The commands use the following driver bundle: v11.1.1fp1_aix64_odbc_cli_32.tar.gz.
- Install the ODBC driver manager package and its dependencies from www.perzl.org/aix (Obtained from www.perzl.org/aix, on February 1, 2018). Dependencies include:
  - info-6.4-1.aix5.1.ppc.rpm
  - libiconv-1.15-1.aix5.1.ppc.rpm
  - libtool-ltdl-1.5.26-2.aix5.1.ppc.rpm
  - readline-7.0-3.aix5.1.ppc.rpm
  - libffi-3.1.1.aix5.1.ppc.rpm (not a direct dependency, but needed by the SAM monitor plugin)
- The gunzip utility, which is part of the AIX 5L™ default installation setup. You can also download gzip for AIX 5L from http://www.ibm.com/servers/aix/products/aixos/linux/rpmgroups.html (© 2018 IBM Corp.; https://www-01.ibm.com; link acquired on August 2, 2018)
- Verify that the IBM rpm.rte package exists in the /usr/bin/rpm directory

Configure ODBC for AIX

1. Log into the AIX-based computer with administrator privileges.
2. Install the RPM packages.
   Run the following command as root: `rpm -ivh unixODBC-2.3.4-1.aix5.1.ppc.rpm`
   To install all dependencies, run the following command that expects the dependencies are in the same directory:
   `rpm -ivh *.rpm`
3. Install the DB2 ODBC driver. The example code creates a directory odbc_cli_32 in the /opt directory.
   `gunzip v11.1.1fp1_aix64_odbc_cli_32.tar.gz`
   `tar -x -f v11.1.1fp1_aix64_odbc_cli_32.tar -C /opt`
4. Add the following lines to /opt/freeware/etc/odbcinst.ini:
   ```
   [DB2]
   Description = DB2 Driver
   Driver = /opt/odbc_cli_32/clidriver/lib/libdb2.so
   FileUsage = 1
   DontDLClose = 1
   ```
   Verify libdb2 has the correct file path.
5. Add the following settings to `/opt/freeware/etc/odbc.ini`:

```
[dbname]
Description = IBM DB2 ODBC data source
Driver = DB2
DMEnvAttr = SQL_ATTR_UNIXODBC_ENVATTR={/opt/odbc_cli_32/clidriver}
```

- In `[dbname]`, enter the name of the database to be monitored.
- `DMEnvAttr` value depends on the installation directory for the DB2 ODBC CLI driver.

6. Run the following command as root. It allows the unixODBC manager driver to find the DB2 ODBC file.

```
ar -X32 x libdb2.a shr.o
mv shr.o libdb2.so
```

7. Verify that the SAM python plugin can connect to the DB2 database to monitor it. Save the following script as `verifyodbc.py`. The script is written for Orion agents in the default location `/opt/SolarWinds/Agent`.

```
Important: If you are not using the default port, add server hostname and port number to the connection string.

```python
import sys
sys.path.insert(0, "/opt/SolarWinds/Agent/bin/Plugins/SharedPythonLibs")
import pypyodbc

# Set Database, Uid, and Pwd to the database name, username, and user's password to access said database respectively.
# In this example, database name, username, and the user's password are TEST, db2inst1, and Password1 respectively.

conn_str = 'Driver={DB2};Database=TEST;Uid=db2inst1;Pwd=Password1;'

You can also use the following script to verify that the SAM python plugin can connect to the DB2 database:

```python
conn = pypyodbc.connect(conn_str, ansi=True)
conn.close()
except Exception, ex:
    print ex
    print "Failure"
```

8. Run the script using the following command:

```
/opt/SolarWinds/Agent/bin/python2 verifyodbc.py
```
9. SolarWinds recommends that you configure the utilities bundled in the DB2 ODBC CLI bundle. Create the db2cli.ini configuration file in /opt/odbc_cli_32/clidriver/cfg/db2cli.ini:

```
TEST
Database=TEST
dbalias=TEST
Protocol=TCPIP
Hostname=127.0.0.1
ServiceName=50000
```

Configure and assign the SAM IBM DB2 template to your Linux/Unix-based computer with the instance user credentials.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Search for a IBM DB2 template. SolarWinds recommends creating a copy of the template. Select a template based on version and click Copy.
3. Modify the settings of the template and component monitors based on the metrics you want to monitor.
4. To enter credentials, select the component monitor check boxes and click Assign Credentials.
5. Enter the credentials for the instance user account and click OK.
6. Modify the connection strings for monitors to match your DB. Use one of the following methods:

   - **Specify the actual DB and driver:**
     ```
     Driver=
     {DB2};Database=TESTDB;Hostname=${IP};Port=50000;Protocol=TCPIP;Uid=${USER};Pwd=${PASSWORD};
     ```
   - **Specify the existing data source in odbc.ini:**
     ```
     DSN=testdb;Uid=${USER};Pwd=${PASSWORD};
     ```

**Scripts with text output**

Scripts can report additional details by sending text to the script's standard output. SAM supports multiple values returned by a script using the following format:

```
Statistic.Name1: xMessage.Name1: abc
Statistic.Name2: yMessage.Name2: abc
```
### Use Windows Script monitors

This component monitor runs a Windows script on the Orion server and then processes the script's exit code and text output. This monitor can return up to ten pairs — 10 statistic values and 10 optional messages. This is best used in conjunction with the Multiple Statistic Chart. To create this monitor, see Use Windows Script monitors.

A maximum of 10 output pairs can be returned. If you exceed the maximum allowed, remove the excess output pairs or they are ignored.

#### Statistic

The statistic for this component monitor is the value returned by the script.

#### Script Monitor Formatting

At least one message and statistic is required. The statistic must be a valid integer and be able to be converted to double, otherwise it is handled as Not as Number (NaN). There is no maximum length for the message; however, only alphanumeric characters and the underscore are allowed.

---

<table>
<thead>
<tr>
<th>DETAIL TYPE</th>
<th>REQUIRED</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>Yes</td>
<td>A numeric value used to determine how the monitor compares to its set thresholds. This must be an integer value, (negative numbers are supported). Statistic.Name1: 123 Statistic.Name2: 456</td>
</tr>
<tr>
<td>Message</td>
<td>No</td>
<td>An error or information message to be displayed in the monitor status details. Note: Multi-line messages are supported. To use this functionality print each line using a separate command. For example: Message.Name1: abc Message.Name2: def</td>
</tr>
</tbody>
</table>

There is a limit of ten Statistic and Message pairs for the script. These can be placed anywhere in the script output. The Statistic and Message names you give must contain valid letters and/or numbers.

Sample output:

```
# Script comment: This shows two pairs. Ten pairs are possible.
Statistic.CPU: 31.08 Message.CPU: svchost.exe cpu usage
Statistic.RAM: 1234.56 Message.RAM: svchost.exe ram usage
```
Field descriptions

Description
This field provides a default description of the monitor. You can override the default description by adding to or replacing the text, which will then be automatically saved. The variable to access this field is `${UserDescription}`.

Component Type
This describes the type of monitor you are using.

Enable Component
This option determines whether or not the component is enabled. Disabling this component leaves it in the application as deactivated and does not influence application availability or status.

Credential for Monitoring
Select a Windows credential that is both a user who can log on to the Orion server, and has sufficient rights on the target node (which may be the Orion server itself, depending upon your application) to do whatever the script needs to do. For example, if the script does something with WMI, the credentials also need WMI rights on the target node.

Click a credential in the list, or use the `<Inherit credential from node>` option. If the credential you need is not in the credentials list, add it in the Credentials Library. See [Understand the Credentials Library](#) for details.

Script Arguments
This field is in the script editing window and allows you to specify arguments to pass to the script. You may include the variables `${IP}`, `${USER}`, and `${PASSWORD}`, which are replaced respectively by the IP address of the target node, the credential user name, and the credential password.

Script Engine
This field allows you to specify the scripting language to be used. The default value is vbscript. Below is a list of scripting engines that the Windows Script Host supports:

<table>
<thead>
<tr>
<th>Name</th>
<th>File Extensions</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBScript</td>
<td>.vbs</td>
<td>Installed by default</td>
</tr>
<tr>
<td>JScript</td>
<td>.js</td>
<td>Installed by default</td>
</tr>
<tr>
<td>PerlScript</td>
<td>.pls</td>
<td>Freeware</td>
</tr>
<tr>
<td>ooRexxScript</td>
<td>.rxs</td>
<td>Freeware</td>
</tr>
<tr>
<td>PythonScript</td>
<td>.pys</td>
<td>Freeware</td>
</tr>
<tr>
<td>TclScript</td>
<td>.tcls</td>
<td>Freeware</td>
</tr>
<tr>
<td>ActivePHPScript</td>
<td>.phps</td>
<td>Freeware</td>
</tr>
<tr>
<td>NAME</td>
<td>FILE EXTENSIONS</td>
<td>AVAILABILITY</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>RubyScript</td>
<td>.rbs</td>
<td>Freeware</td>
</tr>
<tr>
<td>Object Rexx engine</td>
<td></td>
<td>Commercial</td>
</tr>
<tr>
<td>Delphi scripting engine</td>
<td></td>
<td>Commercial</td>
</tr>
</tbody>
</table>

Note the following details:

- If the application monitor is set to Agentless, the VBscript runs on the main polling engine.
- If the application monitor is set to Agent the script runs on the target node.
- PowerShell uses WINRM that runs over TCP ports 5985 and 5986.
- All Linux, Unix, and Nagios script monitors use SSH over TCP Port 22.

**Script Body**

Click Edit to open the script editing window where you can type or paste script text.

**Status Roll-Up**

This option allows you to choose how you would like the monitor to report the returned results based on the output provided by the script. With the ability to have multiple values returned, selecting how your scripts report back to you offers more flexibility. The default selection is “Show worst status.”

**User Notes**

This field allows you to add notes for easy reference. You can access this field by using the variable, `${UserNotes}`.

See also [Create a Windows Script monitor](#).

**Create custom Windows scripts in SAM**

The SAM [Windows Script monitor](#) supports custom scripts on Windows-based systems. You can create multiple instances of this component monitor per template to run custom scripts.

This information details the format and usage of Windows Scripts for SAM, including script arguments and output formats. This information does not provide information on coding in programming languages such as those listed in the Script Engine field.

After configuring Windows and reviewing the components of a script, [create Windows scripts](#) using the component monitor per template.

**Windows script support in SAM**

SAM supports using Windows scripting on target Windows-based computers.

To locate a list of out-of-the-box Windows Script supporting templates, navigate to the Manage Templates page. Click Settings > All Settings > SAM Settings and click Manage Templates. In the search field, enter Windows to filter the list. These templates use customized instances of the Windows Script Monitor.
Use different scripting languages

On Windows-based computers, you can use one of several scripting languages or install a non-native scripting language.

1. Install the scripting language engine or software on the Orion server.
2. When installed, verify that the install location for the scripting language is in the system environment path variable.
3. In the SAM template Windows script monitor, enter the scripting language in the Script Engine.

How the Windows script works

The Windows Script Monitor executes the script on the Orion server connecting to the target server to collect metrics:

1. Executes the script using the script arguments on the Orion server.
2. The monitor connects to the target server using entered credentials for collecting metrics.
3. SAM parses the text output, saves data, and reports the values using the output formats from the component monitor.

Components of a Windows script

SolarWinds recommends writing scripts in the Windows Script monitor to verify correct access between the main polling engine, SAM, and the target server(s). You can write and test scripts in your preferred Integrated Scripting Environment (ISE), connecting and testing against the target server to verify if the code functions. Test the script through the Windows Script monitor to ensure entered credentials and target server access, and to generate metrics output.

The script in the monitor should include:

- Parameters to run the script in the Scripts Arguments field
- Script code added in the Script Body field with credentials
- The script engine to use for the supported programming language
- Exit codes to report status in the script
- Formatted output after running the script

You must test the component monitor and script before assigning and using the component monitor within a template as is. Testing the script generates the required database tables for metrics output and verifies output processes.

Script Arguments field

Use this field in the Edit Script window to specify arguments to pass to the script. You may include the variables \${IP} (or \${Node.DNS}), \${USER}, and \${PASSWORD}, which are replaced respectively by the IP address of the target node, the credential user name, and the credential password.
Script Body field

Use the Script Body field to enter your script code. You can write and compile the script in a language you prefer then copy and paste the code into this field. For example, customers have used Python. Depending on the language, you may need to take additional steps for the script and command line to execute.

Scripts must report their status by exiting with the appropriate exit code. The exit code is used to report the status of the monitor, displayed in the Orion Web Console.

Script Engine

This field allows you to specify the scripting language to be used. The default value is `vbscript`. Below is a list of scripting engines that the Windows Script Host supports:

<table>
<thead>
<tr>
<th>NAME</th>
<th>FILE EXTENSIONS</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBScript</td>
<td>.vbs</td>
<td>Installed by default</td>
</tr>
<tr>
<td>JScript</td>
<td>.js</td>
<td>Installed by default</td>
</tr>
<tr>
<td>PerlScript</td>
<td>.pls</td>
<td>Freeware</td>
</tr>
<tr>
<td>ooRexxScript</td>
<td>.rxs</td>
<td>Freeware</td>
</tr>
<tr>
<td>PythonScript</td>
<td>.pys</td>
<td>Freeware</td>
</tr>
<tr>
<td>TclScript</td>
<td>.tcls</td>
<td>Freeware</td>
</tr>
<tr>
<td>ActivePHPScript</td>
<td>.phps</td>
<td>Freeware</td>
</tr>
<tr>
<td>RubyScript</td>
<td>.rbs</td>
<td>Freeware</td>
</tr>
<tr>
<td>Object Rexx engine</td>
<td></td>
<td>Commercial</td>
</tr>
<tr>
<td>Delphi scripting engine</td>
<td></td>
<td>Commercial</td>
</tr>
</tbody>
</table>

₁ VBScript runs locally on the Orion server only. PowerShell uses WINRM that runs over TCP ports 5985 and 5986. All Linux, Unix, and Nagios script monitors use SSH over TCP Port 22.

Create custom component script monitors

The following sections provide guidance to help you create some of the more advanced types of component monitors.

- Use Windows Script monitors
- Linux/Unix Script monitor
- Nagios Script monitor
- Create a Windows PowerShell monitor
To learn about settings for each component monitor, click the More Information link in the SolarWinds SAM component monitor description.

SolarWinds fully supports scripts written and provided by the company; however, we do not provide customer support for custom scripts written by outside sources. SolarWinds does provide sample scripts that we do support located at: C:\Program Files\SolarWinds\Orion\APM\Sample-Script Monitors.

Ten output pairs can be returned when using script monitors. A usage example using the PowerShell script monitor might go like this:

Imagine you have an Exchange PowerShell script. With multiple values returned, you can get a mail traffic report broken down by day, hour, message size, and number of recipients.

If you exceed the maximum number of allowed output pairs of ten, the remainder above the tenth output pair is ignored.

Create a Windows Script monitor

The Windows Script monitor allows you to run custom scripts in SAM to monitor Windows-based computers. Before coding and testing a script, review the Windows script information and components and best practices for SAM.

You can create a Windows Script monitor to have SolarWinds SAM run a script using Windows Script Host. Windows Script Host comes with VBScript and Jscript, but can be extended with other scripting languages.

Scripts run on the Orion server and use the credentials you specify. The script must both return an exit code and output a text string containing a statistic value conforming to the specifications described later in this section.

Format the Statistic value to use the same decimal separator as the Orion server. The Orion server uses the decimal separator set by its Microsoft Windows regional settings. You may need to log in with an administrator account to perform this action.

1. Click Settings > All Settings > SAM Settings and click Manage Templates.
   The Manage Application Monitor Templates page opens. All available out-of-the-box and custom templates display.
2. Search to locate, select, and edit an existing template or click Manage SAM templates.
3. Click Add Component Monitor(s) and choose Manually Add Component Monitors.
5. Enter Script Arguments to run for the script. Include the variables \${IP} (or \${Node.DNS}), \${USER}, and \${PASSWORD}, which are replaced with the IP address of the target node, the credential user name, and the credential password.

- Depending on the script type, the command line may require additional information and parameters such as the file path or hostname.

7. Click Edit Script to enter and test the script.

8. Test the script by selecting settings for the output, specified node, and specified credentials. Click Get Script Output.

Output results display using the script code with success or failure and any additional notes.

9. Click Get Script Output to generate the output and click Save.

The metrics save to outputs to configure as part of the component monitor. These outputs display at polling intervals for the template.

10. When complete, click Submit to save the template.

When tested, output generates into customizable sections. Each output section is named by the Unique ID with a Display Name you can edit.

For each output, you can optionally:

- Convert the value using selected formulas.
- Configure the statistic thresholds to refine alerting for the component monitor. Set the warning and critical thresholds with specific values or baseline data.
- Select a roll-up status sort option to display the best or worst status up through the template and component monitor.
- Add notes to describe the script output.
Adapt an existing Visual Basic script to a Windows Script Monitor in a new template

1. In the Orion Web Console, click Settings > SAM Settings > Create a New Template, then name the template.
2. Click Add Component Monitor, then expand the Custom Component Monitors group, and then check Windows Script Monitor.
3. Click Submit, and then select credentials with the appropriate permissions to run the script on the Orion server, and that also has appropriate permissions to do whatever else the script requires.
4. Copy the Visual Basic script into the Script Body field.
5. Type any script arguments into the Script Arguments field.
6. Specify the critical and warning thresholds, then click Submit.

Macros for Script Arguments

Specify script arguments in the Script Arguments field if needed. You can use the following variables as script arguments:

`$IP`

This is replaced with the target node's IP Address.

`$USER`

This is replaced with the user name from the credential set.

`$PASSWORD`

This is replaced with the password from the credential set.

Script exit codes and text output

Scripts must report their status by exiting with the appropriate exit code, which is then used to report the status of the monitor in the Orion Web Console, as described in Report status through exit codes. For example, to inform SolarWinds SAM that a VBScript reports Up status, you would exit the script using code similar to the following, where 0 reports Up: 

```wscript.quit(0)```
Scripts report additional details by sending text to the script's standard output. See Scripts with text output for details.

Example scripts

Below is a sample vbscript that returns two values:

- The total number of files in a folder
- Twice the total number of files in the same folder

To use this script, copy and paste the following code into the Script Body field. In the Scripts Arguments field, type in C:\Windows, or any other folder you want to monitor.

```
Option Explicit
On Error Resume Next
Dim lstArgs, path, fso, objDir, objFiles, objFiles2
Set lstArgs = WScript.Arguments
If lstArgs.Count = 1 Then
    path = Trim( lstArgs( 0 ))
Else
    WScript.Echo "Message: Usage: wscript.exe filelist.vbs [pathToFiles]"
    &"&vbCRLF &"[pathToFiles] Local or UNC Path"
    WScript.Echo "Statistic: 0"
    WScript.Echo "Message: Usage: wscript.exe filelist.vbs [pathToFiles]"
    &"&vbCRLF &"[pathToFiles] Local or UNC Path"
    WScript.Echo "Statistic: 0"
    WScript.Quit( 1 )
End If
Set fso = Wscript.CreateObject( "Scripting.FileSystemObject" )
If fso.FolderExists( path ) Then
    Set objDir = fso.GetFolder( path )
    If( IsEmpty( objDir ) = True ) Then
        WScript.Echo "Message: Object Not Initialized"
        WScript.Echo "Statistic: 0" WScript.Quit( 1 )
    End If
    Set objFiles = objDir.Files
    If( IsEmpty( objFiles ) = true) Then
        WScript.Echo "Message: Object Not Initialized"
        WScript.Echo "Statistic: 0"
        WScript.Quit( 1 )
    End If
    WScript.Echo "Message.Total: " &"& CInt( objFiles.Count ) &"" files in this folder.""
    WScript.Echo "Statistic.Total: " &"& CInt( objFiles.Count )
    WScript.Echo "Message.Twice: " &"& CInt( objFiles.Count*2 ) &"" = twice the number of files in this folder.""
    WScript.Echo "Statistic.Twice: " &"& CInt( objFiles.Count*2 )
```
There are several examples of Windows Script component monitors included in templates. These include: File Count, File Modified, LDAP Connection Monitor, Run 3rd Party Application, and Windows Event Log Count.

Sample scripts are installed on your Orion server, in the folder: C:\Program Files\SolarWinds\Orion\APM\SampleScriptMonitors\WindowsScripts

**Linux/Unix Script monitor**

This component monitor uses SSH to upload a script to a Linux/Unix server, runs the script on the server, and then processes the return value and text output. To create this monitor, see [Create a Linux or Unix script monitor](#).

To use scripts with Nagios formats, we recommend using the [Nagios Script monitor](#).

**Statistic**

The statistic for this component monitor is the Statistic value returned by the script.

A maximum of 10 output pairs can be returned. If you exceed the maximum allowed, remove the excess output pairs or they will simply be ignored.

**Prerequisites for Orion agent for Linux**

You need to include credentials that can run scripts on the monitored Linux-based computer. Agentless monitoring does not require these credentials.

You need to verify your installation of python and python-xml. Depending on your Linux distribution, you may have need to install these libraries. You can verify your Python installed libraries with the following command:

```python
python -c "import sys; from xml.dom.minidom import parseString; print sys.version"
```

See the [Python site](#) for documentation and install commands. The following command typically updates and installs the required Python libraries:

```
apt-get install python
```

Depending on your Linux distribution, you may need to verify install and configure specific applications and services to use the Linux/Unix Script monitor. For example, to use the Linux/Unix Script monitor with Apache services (as with the Apache template), you need to configure Apache access. For full configuration details for all services, see [Linux/Unix system configurations for component monitors](#).
Troubleshooting high CPU usage

This component monitor uses SSH for connecting to target machines. Due to supported key exchange algorithms, the JobWorker process used by SAM may encounter high CPU usage. If the issue is due to the algorithm, you can modify the order of supported algorithms. If you have concerns making these changes, contact Support.

1. Locate the file c:\Program Files (x86)\SolarWinds\Orion\APM\SolarWinds.APM.Probes.dll.config.

2. Edit the text file and locate `<LinuxScriptSettings PromptWait="2" ColumnCount="200" TemporaryScriptFileNamePrefix="APM_" />`.

3. Modify the order of algorithms, such as: `<LinuxScriptSettings PromptWait="2" ColumnCount="200" TemporaryScriptFileNamePrefix="APM_" KeyExchangeList="diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group1-shal,diffie-hellman-group14-shal" />`

4. Save changes.

Field Descriptions

Description

This field provides a default description of the monitor. You can override the default description by adding to or replacing the text, which is automatically saved. The variable to access this field is `${UserDescription}`.

Enable Component

Determines whether the component is enabled. Disabling the component leaves it in the application in a deactivated state not influencing either SolarWinds SAM application availability or status.

Authentication Type

Choose either User name and Password or User name and Private Key. The second option allows you to use certificates for authentication.

Credential for Monitoring

Select a credential from the list provided or use the `<Inherit credential from node>` option. If the credential you need is not in the credentials list, add it in the Credentials Library that you can access by clicking Settings > All Settings > SAM Settings in the Orion Web Console.

When providing credentials, make sure that the account:

- Can access the Linux or Unix server over SSH.
- Has sufficient rights to run scripts.
- Exists on the system receiving the executed script.

To learn about credentials used by Orion agents, see the Orion Platform Administrator Guide.

![Tips and Notes]

If you do not enter credentials or use the Inherit-from-node option, the monitor executes the script using the agent credentials (SWIAgent). These credentials may not
have the elevated permissions required for executing scripts.

**Port Number**

This field allows you to specify the port number used for the SSH connection. The default value for this field is 22.

**Script Working Directory**

This field allows you to specify the working directory of the script process.

**Count Statistic as Difference**

Changes the statistic to be the difference in query values between polling cycles.

**Command Line**

This field allows you to specify the shell command run after the SSH connection is established. The default command line value `perl ${SCRIPT} arg1 arg2` attempts to run in a Perl interpreter the script defined in the Script Body field using the parameters `arg1 arg2`.

The length of the Command Line field is limited to 266 characters for Solaris systems, minus the length of the `${SCRIPT}` variable after being resolved to a file name such as the following: APM_937467589.pl. Since the length of the file name will typically be around 16 characters, this means that the actual user Command Line input cannot be longer than 266 – 16, or about 250 characters (not including the length of the 9 characters for the `${SCRIPT}` variable itself).

If you need to pass a longer command line to the target node, you can create a shell script on the target node (for example `myscript.sh`) that contains the long command line, and place the call to this script in the Command Line field, for example: `/opt/sw/myscript.sh`.

**Script Body**

This field allows you to specify the script you want to run on the Linux or Unix server.

**Status Roll-Up**

This option allows you to choose how you would like the monitor to report based on the output provided by the script. The default selection is “Show worst status.”

**User Notes**

This field allows you to add notes for easy reference. You can access this field by using the variable, `${UserNotes}`.

Create a Linux or Unix script monitor

Linux and Unix Script component monitors allow you to execute a command line script that can return statistical data. When collecting information for this monitor, SAM runs the script with the credentials defined with the Credential Library.

A maximum of 10 output pairs can be monitored per script monitor. More than 10 defined pairs results in a failure of the monitor.
Adapt an existing Perl script to a Linux/Unix Script component monitor in a new template

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Create a new template, and then name the template.
2. Click Add Component Monitor, expand the Custom Component Monitors group, and then select Linux/Unix Script Monitor.
3. Click Submit then select credentials with SSH permissions in the Credential for Monitoring field.
4. Type a valid working directory in the Script Working Directory field.
5. Click Edit to open the script dialog and enter your script into the Script Body field.
6. Type the Linux command that runs the script in the Command Line field. It should be similar to the following: perl ${SCRIPT} arg1 arg2.

   {SCRIPT} is replaced by the actual file name of the script after it’s deployed to the target node. A temporary file is created in temp directory for the script.
7. Click Get Script Output. SAM then tests the script by executing it and parse its output, then return the values.
8. Click Save, then specify the critical and warning thresholds and click Submit.

Here is a sample Perl script using the Linux/Unix component monitor returning multiple output pairs, in this case, two:

```perl
#!/usr/bin/perl
if (@ARGV[0] =~ /hel\b/) {
    print "MemoryUsage.pl SNMPver community hostname\n";
    print "SNMPver - version of SNMP protocol\n";
    print "community - community name of SNMP protocol\n";
    print "hostname - Target host\n";
    exit 1;
}
# Get hostname and trim newline
$localhost = `hostname`;
$localhost =~ s/\s*$//g;
$hostname = shift || $localhost; $community = shift || "public";
$version = shift || "v1";
$results = "";
$MIB_TotalMemory = "UCD-SNMP-MIB::memTotalReal.0";
#$MIB_TotalMemory = "UCD-SNMP-MIB::memTotalReal.0";
$outres = "snmpget -$version -c $community $hostname $MIB_TotalMemory |";
open(OUTMEM,$outres) || die "Unable read pipe\n";
```

The code in red shows where the output pairs are defined.
while ($line =~ /\bINTEGER\b/) {
    if ($line =~ /:\/\n/) {
        $indval = index($line,":");$indval = index($line,"="); $val = substr($line,$indval+1,length($line) - $indval); $val =~ s/[a-zA-Z]/\n/; print "Message.1: Available memory at host "hostname": $val in kB"; print "Statistic.1: $val \n"; print "Message.2: Again, the available memory at host "hostname": $val in kB"; print "Statistic.2: $val \n"; exit 0;
    }
} print "Statistic: 0\n"; exit 1;

Below is the output from this script:

```
Script Output: Message.1: Available memory at host "lab-fedora32": 1027012 in kB
Statistic.1: 1027012
Message.2: Again, the available memory at host "lab-fedora32": 1027012 in kB
Statistic.2: 1027012
```

Linux/Unix scripts

Linux/Unix scripts are uploaded by SSH and then run on the target node using the string from the Command Line field.

You can use the following variable in the command line field:

- `{SCRIPT}` - Replaced with the script body.

You can use the following variables in the script body field:

- `{IP}` - Replaced with the target node's IP address.
- `{USER}` - Replaced with the user name from the credential.
- `{PASSWORD}` - Replaced with the password from the credential.

Example scripts

There are several sample scripts installed with SolarWinds SAM you can use to create Linux/Unix script component monitors. These sample scripts are installed on your Orion server, in the following folder: C:\Program Files\SolarWinds\Orion\APM\SampleScriptMonitors\LinuxScripts

Scripts must report status through exit codes

Scripts must report their status by exiting with the appropriate exit code. The exit code is used to report the status of the monitor, which is seen by the user through the interface.
To correctly create this component monitor, you must first return an exit code which results in an Up (0), Warning (2), or Critical (3) status. When one of these exit codes is received the appropriate dynamic evidence table structure is created and all further exit codes are handled correctly. If the component only returns Down (1) or Unknown (4) on first use, the appropriate dynamic evidence table structure is not created appropriately.

<table>
<thead>
<tr>
<th>EXIT CODE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Up</td>
</tr>
<tr>
<td>1</td>
<td>Down</td>
</tr>
<tr>
<td>2</td>
<td>Warning</td>
</tr>
<tr>
<td>3</td>
<td>Critical</td>
</tr>
<tr>
<td>Any other value</td>
<td>Unknown, for example 4</td>
</tr>
</tbody>
</table>

For example, if you want to inform SolarWinds SAM that a script reports an Up status, exit the script using code similar to the following, where 0 reports Up: `Wscript.quit(0)`

Scripts with Text Output

Scripts report additional details by sending text to the script's standard output.

SAM supports multiple values returned by a script using the following format.

```
Statistic.Name1: x
Message.Name1: abc
```

```
Statistic.Name2: y
Message.Name2: abc
```

<table>
<thead>
<tr>
<th>DETAIL TYPE</th>
<th>REQUIRED</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>Yes</td>
<td>A numeric value used to determine how the monitor compares to its set thresholds. This must be an integer value. Negative numbers are supported.</td>
</tr>
</tbody>
</table>
|             |          | `Statistic.Name1: 123`
|             |          | `Statistic.Name2: 456`

<table>
<thead>
<tr>
<th>Message</th>
<th>No</th>
<th>An error or information message to be displayed in the monitor status details. Note: Multi-line messages are supported. To use this functionality, print each line using a separate command. For example:</th>
</tr>
</thead>
</table>
|             |          | `Message.Name1: abc`
|             |          | `Message.Name2: def`

There is a limit of ten Statistic and Message pairs for the script. These can be placed anywhere in the script output. The Statistic and Message names you give must contain valid letters and/or numbers.

Sample output:
Nagios scripting

If you are using this component monitor with Nagios and polled by Orion agent for Linux, you need to enter a script as follows. The argument would be python $(SCRIPT) with a body of:

```python
from random import randint
STAT_PAIRS_COUNT = 10

for stat_id in range(STAT_PAIRS_COUNT):
    print 'Statistic.random%d: %d
    print 'Message.random%d: %s' % (stat_id, randint(1,100), stat_id, 'random number: ' + str(randint(1,100)))
```

To use scripts with Nagios formats, we recommend using the Nagios Script monitor.

Nagios Script monitor

This component uses SSH to upload a Nagios script to a Linux/Unix server, runs the Nagios script on the server and then processes the script's exit code and text output. This monitor can return multiple values.

Statistic

The statistic for this component monitor is the value returned by the script. This component monitor can return multiple results and process Nagios resulting output. For more information, see http://nagios.sourceforge.net/docs/3_0/pluginapi.html.

A maximum of 10 output pairs can be returned. If you exceed the maximum allowed, remove the excess output pairs or they are ignored.

Prerequisites for Orion agent for Linux

Include credentials that can run scripts on the monitored Linux-based computer. Agentless monitoring does not require these credentials.

If you are using this component monitor with Nagios and polled by Orion agent for Linux, you need to enter a script as follows. The argument would be python $(SCRIPT) with a body of:

```python
from random import randint
STAT_PAIRS_COUNT = 10

for stat_id in range(STAT_PAIRS_COUNT):
```
Troubleshooting high CPU usage

This component monitor uses SSH to connect to target machines. Due to supported key exchange algorithms, the Job Worker process used by SAM may encounter high CPU usage. If the issue is due to the algorithm, you can modify the order of supported algorithms. If you have concerns making these changes, contact Support.

1. Locate the file c:\Program Files (x86)\SolarWinds\Orion\APM\SolarWinds.APM.Probes.dll.config
2. Edit the text file and locate <LinuxScriptSettings PromptWait="2" ColumnCount="200" TemporaryScriptFileNamePrefix="APM_" />
3. Modify the order of algorithms, such as: <LinuxScriptSettings PromptWait="2" ColumnCount="200" TemporaryScriptFileNamePrefix="APM_" KeyExchangeList="diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group1-shal,diffie-hellman-group14-shal" />
4. Save changes.

Field Descriptions

Description

This field provides a default description of the monitor. You can override the default description by adding to or replacing the text, which is automatically saved. The variable to access this field is ${UserDescription}.

Component Type

This describes the type of monitor you are using.

Enable Component

This option determines if the component is enabled. Disabling this component leaves it in the application as deactivated and does not influence application availability or status.

Authentication Type

Choose either User name and Password or User name and Private Key. The second option allows you to use certificates for authentication.

Credential for Monitoring

Select a Windows credential that is both a user who can log on to the Orion server, and has sufficient rights on the target node to do whatever the script needs to do. For example, if the script does something with WMI, the credentials also need WMI rights on the target node.

Click a credential in the list, or use the <Inherit credential from node> option. If the credential you need is not in the credentials list, add it in the Credentials Library that you can access by clicking Settings > All Settings > SAM Settings in the Orion Web Console.
Port Number

This field allows you to specify the port number used for the SSH connection. The default value is 22.

Script Working Directory

This field allows you to specify the working directory of the script process.

Check Type

Set the check type to Service or Host. To prevent false positives, Nagios allows you to define how many times a service or host should be (re)checked before alerting for an issue. Depending on the select, the monitor checks using the configured value on the server for max_check_attempts option in the host and service definitions.

Count Statistic as Difference

Changes the statistic to be the difference in query values between polling cycles.

Command Line

This field is in the script editing window and allows you to specify the script you want to run on the target node followed by the arguments. To enter your script, click the Edit button that reveals the script editing window. For more information, see http://nagios.sourceforge.net/docs/3_0/pluginapi.html.

- The length of the Command Line field is limited to 266 characters for Solaris systems, minus the length of the ${SCRIPT} variable after being resolved to a file name such as the following: APM_937467589.pl. Since the length of the file name will typically be around 16 characters, this means that the actual user Command Line input cannot be longer than 266 - 16, or 250 characters (not including the length of the 9 characters for the "${SCRIPT}" variable itself). If you need to pass a longer command line to the target node, you can create a shell script on the target node (for example: myscript.sh) that contains the long command line and place the call to this script in the Command Line field, for example: /opt/sw/myscript.sh

Script Body

This field is in the script editing window and allows you to enter your script via typing or pasting. You can test the script to receive output definitions. You can save these definitions to the component monitor for further configurations. Every saved definition is listed as Script Output with an assigned number and name. See Test script output for details.

Status Roll-Up

This option allows you to choose how you would like the monitor to report based on the output provided by the script. The default selection is “Show worst status.”

User Notes

This field allows you to add notes for easy reference. You can access this field by using the variable, ${UserNotes}.
Scripts must report status through return codes

Nagios determines the status of a host or service by evaluating the return code. The following table shows a list of valid return codes, along with their corresponding service or host states.

To correctly create this component monitor, you must first return an exit code which results in an Up (0), Warning (2), or Critical (3) status. When one of these exit codes is received the appropriate dynamic evidence table structure is created and all further exit codes are handled correctly. If the component only returns Down (1) or Unknown (4) on first use, the appropriate dynamic evidence table structure is not created appropriately.

<table>
<thead>
<tr>
<th>RETURN CODE</th>
<th>SERVICE STATE</th>
<th>HOST STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
<td>Up</td>
</tr>
<tr>
<td>1</td>
<td>Warning</td>
<td>Up or Down/Unreachable†</td>
</tr>
<tr>
<td>2</td>
<td>Critical</td>
<td>Down/Unreachable</td>
</tr>
<tr>
<td>3</td>
<td>Unknown</td>
<td>Down/Unreachable</td>
</tr>
</tbody>
</table>

†If the Use Aggressive Host Checking option is enabled, return codes of 1 will result in a host state of Down, otherwise return codes of 1 will result in a host state of Up.

Nagios scripts must exit with a valid return code and a line of text output. The exit code determines the status of the component. If the exit code is 0 (OK), the component status may be further modified by thresholds from the optional statistics. To return up to ten optional statistics, separate the statistics from the status message with the pipe (|) symbol using the following syntax:

```
statusMessage | ['statisticName']=value
```

Below is an example of valid output with a status message and two statistics:

```
The script ran. | 'CPU%'=75.2 'MemoryRemainingInKB'=600784
```

Test script output

You should test the script output while editing the script prior to testing the script in the template or application component pages. If the output formatting or values are not correctly defined or missing, you may receive an error: "Script output values are not defined or improperly defined." This error displays if the named fields could not be located in the script output.

To test the script and save output definitions:

1. Open the template or application monitor using the Nagios Script monitor. To open, click Settings > All Settings > SAM Settings > Manage Application Monitors. Locate and edit the application monitor or template with the Nagios Script monitor.
2. Locate and expand the component using the Nagios Script monitor type in the template.
3. Locate the Script Body field and click Edit Script.
4. On the Edit Script dialog, click Get Script Output. You may be prompted to specify a test node and credentials.

5. Wait for the Output Result. The results should populate with values returned by the script. Review the results to ensure all formatting is correct and fields properly load.

6. You can store the output definitions returned by the script test as Script Output in the component monitor. Click Save to add the output definitions. The component monitor will display the definitions with a unique ID, display name, and additional configuration options.

7. To save changes to the template or application monitor, click Submit.

Create a Windows PowerShell monitor

Before creating a PowerShell monitor, review Using PowerShell in SAM to learn about PowerShell requirements, security considerations, installation, and enabling remote access with the Windows Remote Management (WinRM) service.

This component monitor runs a Windows PowerShell script on the Orion server or a remote target node to collect metrics as follows:

1. The monitor checks if the execution mode is Local Host or Remote Host.
   If Local Host, the script executes using the script arguments on the Orion server.
   If Remote Host, the script connects via SSH connection to run the script on the target server.
2. The script executes and collects metrics from the target server using entered credentials.
3. SAM parses the text output, saves data, and reports values using output formats from the component monitor.

This monitor can return up to ten pairs — 10 statistic values and 10 optional messages. If you exceed the maximum allowed, remove the excess output pairs or they will simply be ignored. You may need an administrator account to perform this action.

This section includes field descriptions and a Windows PowerShell monitor example. Before creating a PowerShell monitor, review Using PowerShell in SAM to learn about PowerShell requirements, security considerations, installation, and enabling remote access with the Windows Remote Management (WinRM) service.

Your organization should internally review and assess to what extent PowerShell scripts will be incorporated into your environment. See PowerShell security considerations for details.

Windows PowerShell monitor field descriptions

Description

Add or replace text to override the default description of the monitor. The variable to access this field is \${UserDescription}.
Enable Component

Determines if the component is enabled. Disabling the component leaves it in a deactivated state that does not influence either SolarWinds SAM application availability or status.

Credential for Monitoring

Select a Windows credential with rights to log into the Orion server plus sufficient rights on the target node to do whatever the script needs to do. For example, if a script does something with WMI, the credentials also need WMI rights on the target node. Some PowerShell commands require the use of the ${CREDENTIAL} variable; see Script Body for details.

To increase security, SolarWinds recommends using a dedicated Windows account with minimal privileges for PowerShell monitors, especially for scripts executed on the main polling engine (that is, the Orion server). For details, see How SAM handles credentials based on Execution Mode.

Execution Mode

Leave this value set to the default value, Local Host, to run scripts locally from the Orion server. Make sure that WinRM is properly configured on the Orion server so scripts can run on remote target servers.

If you select Local Host but do not enable the "Run the script under specified account" option, the script has the same unlimited access privileges as other Orion services, which presents high risk from a security perspective. To learn more, see How SAM handles credentials based on Execution Mode.

Select Remote Host to execute scripts on the selected target node. The following options are available for Remote Host mode:

- **Use HTTPS Protocol**: The default value is HTTP. Select HTTPS if you want the monitor to send and receive encrypted Web Services (WS)-Management protocol requests and responses for increased security.
- **URL Prefix**: Specify the URL prefix on which to accept HTTP or HTTPS requests. The default is wsman.
- **Port Number**: Specify the TCP port used to listen for traffic. For WinRM 1.1 and earlier, the default port is 80. For WinRM 2.0, the default port is 5985.

Count Statistic as Difference

Enable this option to include the difference between two polling intervals in script output.

Run the script under specified account

Select this option to enable impersonation with the component's credentials. (This works only if Local Host is selected as the Execution Mode.) See also How SAM handles credentials based on Execution Mode.
Script Body

Specify the PowerShell script you want to run.

SolarWinds recommends that you always review the Script Body to check for malicious code, Custom scripts you create or download from THWACK are not part of the SolarWinds software purchased from SolarWinds. Your organization should internally review and assess to what extent PowerShell scripts will be incorporated into your environment. You elect to utilize custom scripts at your own risk, and you will be solely responsible for the incorporation of the same, if any.

If a script includes PowerShell commands that require valid credentials for the Orion server and target servers (such as Get-WmiObject), use the ${CREDS} variable, as shown in this example:

```powershell
$avg = Get-WmiObject win32_process -ComputerName '${IP}' -Credential '${CREDS}' | Where-Object {$_.Name -eq "lsass.exe"} | Measure-Object -property ReadOperationCount -Average;
```

The ${CREDS} variable provides the user name and password specified in the Credential for Monitoring field when prompted by the script so there is no need to include that variable in the Script Argument — credentials are provided automatically.

To pass a custom property to a script, using ${Node.Custom.XXX}, where xxx is the name of the custom property.

Script Arguments

Specify arguments to pass to the script. You may include the variable ${IP}, which is replaced by the IP address of the target node. Do not include variables that are stored automatically, such as the ${CREDS} variable.

User Notes

Add notes for reference, accessible by using the variable, ${UserNotes}.

How SAM handles credentials based on Execution Mode

The following table describes how SAM handles credentials for a PowerShell script based on:

- The selected Execution Mode: Local Host or Remote Host
- If the "Run the script under specified account" option is enabled.
<table>
<thead>
<tr>
<th>MODE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Host Disabled</td>
<td>The Credential for Monitoring is not used. Instead, the PowerShell script uses the same Local Admin account as the Orion Platform to run the script via an Orion Job Engine V2 service.</td>
</tr>
<tr>
<td></td>
<td>This method grants the script the same unlimited access privileges as other Orion services, which presents high risk from a security perspective.</td>
</tr>
<tr>
<td>SolarWinds recommends using a dedicated Windows account with minimal privileges for PowerShell monitors, especially for scripts executed on the Orion server that performs as the main polling engine.</td>
<td></td>
</tr>
<tr>
<td>Local Host Enabled</td>
<td>SAM uses the Credential for Monitoring to run PowerShell scripts on the Orion server.</td>
</tr>
<tr>
<td>Remote Host N/A</td>
<td>The script connects via SSH to run the script on the target server. SAM uses the Credential for Monitoring to run the PowerShell script on the target server. Make sure that:</td>
</tr>
<tr>
<td></td>
<td>• WinRM is enabled and properly configured on the Orion server.</td>
</tr>
<tr>
<td></td>
<td>• The credential has rights to log into the Orion server.</td>
</tr>
<tr>
<td></td>
<td>• The credential has sufficient rights on the target server to perform tasks in the script.</td>
</tr>
</tbody>
</table>

**Windows PowerShell monitor example**

You can create a monitor that runs a Windows PowerShell script to monitor specific performance information for troubleshooting a Windows process that may be having issues.

For this example, the process being monitored is lsass.exe, which enforces security on the system for users who are logging on and changing passwords. In particular, you want to monitor the average number of read operations performed to check for spikes.

To use the Windows PowerShell monitor to run a PowerShell script with a Get-WmiObject call to measure the average ReadOperationCount for the lsass.exe process and monitor its value:
1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Create a New Template.

2. Name the template, for example, Lsass.exe PowerShell Monitor.

3. Click Add Component Monitor, expand the Custom Component Monitors group, select Windows PowerShell Monitor, and click Add.

4. Select the Credential for Monitoring with appropriate permissions to run the script on the Orion server, and that also has appropriate permissions to do whatever else the script requires (in this case, to get the average number of read operations performed on the target node).

5. Select the Execution Mode to use:
   - Local Host can run scripts only locally, that is, on the Orion server.
   - Remote Host can execute scripts remotely (on the remote target node to which the Windows PowerShell monitor is assigned) using the WinRM service, which must be properly configured on the main Orion server and target servers to run PowerShell commands remotely. See Enable remote access for PowerShell with WinRM.

6. Copy the following PowerShell script, which uses the `Get-WmiObject` call to measure the average `ReadOperationCount` for the `lsass.exe` process, into the Script Body field:

   ```powershell
   $avg = Get-WmiObject win32_process -ComputerName '${IP}' -Credential '${CREDENTIAL}' | Where-Object {$_ .Name -eq "lsass.exe"} | Measure-Object -property ReadOperationCount -Average; Write-Host 'Statistic: ' $avg.Averageexit(0)
   ```

The PowerShell code does the following:

a. Reads the average `ReadOperationCount` information for the process `lsass.exe` from the computer whose IP address is specified by the variable `${IP}` using the credential specified by the variable `${CREDENTIAL}`.

   - The user name from the Credential for Monitoring that is specified is stored automatically in the `${CREDENTIAL}` variable by the monitor. Do not add the `${CREDENTIAL}` variable in the Script Arguments field. When the script is run by PowerShell, it prompts for a password. The monitor automatically provides the password from the Credential for Monitoring.

b. Writes the statistic information gathered by the script.

c. Exits the script.

Scripts must report their status by exiting with the appropriate exit code. The exit code is used to report the status of the monitor, which is seen by the user through the interface. See Report status through exit codes.

The script does not perform error checking.
7. Enter the following Script Arguments:
   Use the token ${IP} and the IP address will be filled in with the IP address of the target node. You can then access the value in the script body using the variable ${IP}.
   For example, if you type ${IP} for Script Arguments the PowerShell script will be able to access the IP address for the target node using the variable ${IP} in the script body.

8. Select Run the script under specified account to enable impersonation with the component's credentials. This works only in local script execution mode.

9. Select Count Statistic as Difference to change the statistic to be the difference in query values between polling cycles.

10. Change the Statistic Warning Threshold to, greater than 800.

11. Change the Statistic Critical Threshold to, greater than 1000.

12. Click Set test node. Browse the tree view, select the desired target node for the PowerShell script, and then click Select.

13. Click Test, and then click Submit.

14. Click All in the Select tag to filter by list, and then locate the Lsass.exe PowerShell Monitor.

15. Select Lsass.exe PowerShell Monitor and then click Assign to Node.

16. Expand the tree view and select the target node, and then click Next.

17. Select Inherit credentials from template, and then click Test to confirm the credentials and component monitor against the test node.

18. Click Assign Application Monitors and then click Done.

Example tasks for application monitors

These are scenarios for monitoring with SAM.

- Create your first application and alert
- Scan for applications to monitor
- Monitor a specific URL
- Monitor VMware performance counters using an application monitor
- Monitor and restart stopped Windows services
- Monitor large directories using the Windows Script Monitor
- Monitor IIS application pools
- Use the JMX component monitor wizard
- Configure the SOAP monitor

Create your first application and alert

After being applied to a node, a template becomes an application, which is comprised of component monitors, also known as performance counters. Applications created from templates are used to report metrics based on your needs. This example provides steps for creating a template for monitoring WMI.
You will create a template with added monitors, configure the monitors, then apply the template to nodes for monitoring. We also provide information for creating an alert for the WMI application to send email notices based on monitored thresholds.

You may need to log in with an administrator account to perform this action.

Create a WMI monitor

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Create a New Template.
2. Provide a name for the template and click Add Component Monitor.
3. Select WMI Monitor and click Add.
4. Select credentials or Inherit Credentials from Node.
5. In the Query field, enter a WQL query to return a statistic to be reported.
6. Enter thresholds or select Use thresholds calculated from baseline data.
7. Click Submit.

Apply the WMI monitor template

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Locate your template and click Assign to Node.
3. Select the Windows node from the left pane, click the green arrow to move it to the right pane, and click Next.
4. Choose the credentials, click Assign Application Monitor, then click Done.

Alert on the WMI application via email

1. In the Orion Web Console, click Alerts & Activity > Alerts > Manage Alerts.
2. Click Add New Alert and name the alert.
3. On the Properties tab, enter required information, and click Next.
4. For the Trigger Conditions, complete the section as shown:

   Important: Add the second trigger condition by clicking [+]. The first trigger condition tests for Status (Down), the second for the specific application instance (indicated by 1 Object).

5. For Trigger Actions, click Add Action, select Send an Email/Page, then click Configure Action.
6. Enter the required information, click Next and complete the wizard as instructed, then click Submit.
7. Review and edit the Reset Actions and the Summary, then click Submit.
8. The Manage Alerts page indicates the alert was created successfully.

Scan for applications to monitor

You can scan for applications to monitor as added nodes on the Application Discovery page. SAM scans your network for applications. Select applications from the displayed list to add as monitored nodes.

1. Click Settings > All Settings > SAM Settings > Scan Nodes for Applications.
2. Select nodes by clicking [+] to expand the node groups, then select the nodes you want to scan, and then click Next.
3. Select applications to find, and then click Next.
4. Enter the credentials for the servers you are scanning, and then click Start Scan.
5. Click View SAM Summary Page, then click View results after the SAM scan is complete.

To learn more, see Scan nodes for applications to apply templates automatically in SAM.

Monitor a specific URL

You can add a specific URL as a monitored node. All usage and performance displays as a node with details pages of collected data.

1. Click Settings > All Settings > Add Node.
2. Enter the Hostname or IP Address you want to monitor (for example, www.google.com).
3. Select External Node: No Status, and then click Next.
4. From the Show only: drop-down, select Web Pages.
5. Check Web Link and select Inherit credentials from Template.

![Add Node](image)

6. Click Test, and then click Next.

7. On the Change Properties page, make any changes, and then click OK, Add Node.

8. Navigate to the Node Details page to review the results. To do this, click My Dashboards > Home, and then drill down to your specific node in the All Nodes widget.

Monitor VMware performance counters using an application monitor

Learn more about monitoring VMware performance counters, used by application monitor templates and component monitors.

ℹ️ You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Component Monitor Wizard.
2. Select a VMware performance counter, then click Next.
3. Enter the IP address of the VMware node you want to browse, or click Browse, select the node from the list, and then click Select.
4. Select or enter the appropriate credentials, then click Next.
5. Select whether you want to monitor A single system or Multiple systems.
6. Choosing a single system provides specific counters that apply to the target system only, choosing multiple systems provides more generic counters that can be applied to multiple systems.

Choosing Multiple systems provides only aggregate performance counters (without instances). Therefore, the Application Monitor created based on them can be considered as generic and assigned to different ESX/vCenter target hosts. In this case, monitors in the application created contain the special `${VMWARE_ENTITY_NAME}` variable in the Entity Name field of the monitor. When the monitor runs, this variable is resolved to the first available entity on a target host with the desired Entity Type (for example, the first Host System).

7. Select the desired VMware Entity to monitor, then select the desired Performance Object (group) to monitor.

8. Select the counters to monitor for the selected Performance Object, then click Next.

9. On the Edit Properties page, modify settings as desired and click Next.

Add components to the application monitor

1. Click [+] next to Component list to expand the list of counters or counters to be added.
2. Select New Application Monitor. Then enter a name for the new Application Monitor, then click Next, then OK, Create.
3. Click View SAM Summary Page. Your new VMware monitor appears in the tree view for the All Applications widget. The polling results for the new monitor are updated after a few minutes.

- Cluster Compute Resource (only available if the VMware vCenter Performance Counter Monitor type is selected) – Data object that aggregates the computation resources of its associated Host System objects into one single computation resource for use by virtual machines. The cluster services such as HA (High Availability), DRS (Distributed Resource Scheduling), and EVC (Enhanced vMotion Compatibility), enhance the usefulness of this single computation resource. This Entity Type is specific to vCenter systems.
- Host System – Managed object type that provides access to a virtualization host platform.
- Resource Pool – Represents a set of physical resources which may be a single host, a subset of a host's resources, or resources spanning multiple hosts. You can subdivide Resource pools by creating child resource pools. to run, a virtual machine must be associated as a child of a resource pool. In a parent/child hierarchy of resource pools and virtual machines, the root resource pool is the single resource pool that has no parent pool.
- Virtual Machine – Managed object type for manipulating virtual machines, including templates that can be repeatedly deployed as new virtual machines. This object type provides methods for configuring and controlling a virtual machine.

Monitor and restart stopped Windows services

Learn more about monitoring and restarting stopped Windows services.

You may need to log in with an administrator account to perform these actions.
Create the monitor

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Component Monitor Wizard.
2. Select Windows Service Monitor, and then click Next.
3. Enter the IP address of the Windows node you want to browse, or click Browse, select the node from the list, then click Select.
4. Enter or select the appropriate credentials, then click Next.
5. Check the services to monitor, adjust thresholds as needed, then click Next.
6. Ensure that New Application Monitor Template Name is selected and then enter a name for your new template, then click Next.
7. Select the desired nodes to monitor and then click Next.
8. Click OK, Create, to create the assigned Application Monitor.

Create an alert that restarts any stopped Windows services and sends an email

1. In the Orion Web Console, click Alerts & Activity > Alerts > Manage Alerts.
2. Click Add New Alert > Configure Alerts.
3. Scroll to the bottom of the list in the Manage Alerts dialog, and check the box for Restart a service.
4. To send an e-mail notification for the service restart:
   a. Highlight Restart a Service and click Edit.
   b. Select the Trigger Actions tab and then click Add New Action.
   c. Select Send an E-Mail / Page and click OK.
   d. Complete the e-mail info and click OK.
   e. Click OK.
5. Click Done.

To learn more about alerts and alert actions, see the Orion Platform Administrator Guide.

Monitor large directories using the Windows Script Monitor

- You may need to log in with an administrator account to perform this action.
- Before coding and testing your script, review Windows script information and components and best practices for SAM.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > SAM Settings > Create a New Template.
2. Name the template, click Add Component Monitor, select Windows Script Monitor, and click Add.
3. Provide a brief Description.
4. Select credentials with appropriate permissions to run the script on the target server.
5. In the Script Arguments field, type the Universal Naming Convention (UNC) path name for the directory to monitor.

Use the variable `${IP}` for the IP address of the target node to which the monitor is assigned when the Windows Script monitor runs.

6. Copy the following Visual Basic script, which retrieves the directory size, into the Script Body field:

```vbnet
Dim folderPath
folderPath = WScript.Arguments(0)
Set fs=WScript.CreateObject ("Scripting.FileSystemObject")
Set folder= fs.GetFolder(folderPath)
WScript.Echo "Message: Folder " &folderPath & " is " & folder.Size & " bytes large"
WScript.Echo "Statistic: " & folder.Size
```

7. Specify the critical and warning thresholds for the desired directory size, and then click Submit.

8. Create an assigned Application Monitor by assigning the Large Directory Monitor template to the node.

   a. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manually Assign Application Monitors.
   
   b. Select All in the Show only list, click Large Directory Monitor, and then click Next.
   
   c. Click Manually Assign Application Monitors and Select All in the Show only list.
   
   d. Click Large Directory Monitor and then click Next.
   
   e. Locate and select the desired node, and then click Next.
   
   f. Enter itadmin in the Credential Name field and then enter your credentials.
   
   g. Click Test, click Assign Application Monitors, and then click Done.

When monitoring occurs, the Visual Basic code in the template will:

1. Read the first argument passed to the UNC path name for the directory to monitor and store it in `FolderPath`.

2. Create the `Scripting.FileSystemObject` and store it in `fs`.

3. Retrieve the folder name from the saved command line argument and store it in `FolderPath`.

4. Display the folder name and folder size as output.

5. Display the folder size, measured in bytes.

   The script does not perform error checking.

Monitor IIS application pools

SolarWinds SAM can monitor five instances (for example, w3wp.exe), differentiated by application pools separately if you specify the application pool names in the component monitors.
1. In the Orion Web Console, click Settings > All Settings.

2. Add the web server to the Orion database.
   a. Click Add Node then enter the hostname or IP address of the Intranet web server.
   b. Check the ICMP (Ping only) check box and then click Next.
   c. From the Add Application Monitors page, click Next.
   d. From the Change Properties page, Click OK, Add Node.

3. Find the \wp3w.exe process on the web server.
   a. In the Orion Web Console, click Settings > All Settings > SAM Settings > Component Monitor Wizard.
   b. Select Process Monitor - WMI as the monitor type, and then click Next.
   c. Click Browse, select the web server, and then click Select.
   d. Enter WebServerAdmin in the Credential Name field.
   e. Enter your credentials in the fields provided, then click Next.
   f. Click the last page button to view the last page, then check the check box next to \w3wp.exe, and then click Next.
   g. Change Monitor Name to Webpool1, then enter webpool1 in the Command Line Filter field.
   h. Change the CPU Warning Threshold to greater than 40, then Change the CPU Critical Threshold to greater than 50.
   i. Click Add Another Component. After creating the monitors for all five \w3wp.exe instances, click Next.
   j. Select New Application Monitor and enter Web Server Application Pools as the Application Monitor Name, and then click Next.
   k. Select the web server node, and then click Next, then click OK, Create.

Use the JMX component monitor wizard

Creating a standard template for this monitor is not practical because of the amount of variables in any one specific environment. However, adding this monitor to your environment is simplified by using a wizard driven interface, as described here. See also JMX monitor.

Only values that return numerical data can be monitored. String data is not supported at this time. Non-numerical data is shown without a check box.

Add a JMX component monitor

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Component Monitor Wizard.
2. Select JMX Monitor from the drop down list and then click Next.
3. For Server IP Address, click Browse, then select the node you want to monitor.
4. Add the Port number, Protocol type, URL path, and Credentials for the remaining fields.

5. Click Next. The following MBean selection screen appears:

![MBean selection screen](image)

6. Expand the folders by clicking the arrows (or [+] to expand the tree view folder structure. From here you can drill down to select the attributes you want by checking them.

7. Click Next.

Configure the SOAP monitor

The are two ways to configure the SOAP monitor:

- By loading a WSDL file.
- By entering XML code manually.

Load WSDL Files

The SOAP monitor within SAM currently supports the WSDL schema, which must be exposed on a URL. After the WSDL file is successfully loaded, the file is parsed automatically, the fields populate (as shown next), and you can specify values for the available arguments. There are two types of arguments: simple and complex.

A simple argument is one where you define the value directly. A complex argument (for example, one that includes structures, classes, and lists) must be user-defined in the XML format. SolarWinds recommends that only experienced users compose complex arguments.

Advanced Settings are read-only and display additional information. Because the SOAP envelope is based on the WSDL schema, the Advanced Settings values should not be changed.

The SOAP XML field contains the SOAP envelope which is generated by the WSDL and dynamically changes as you change the SOAP settings. Use this field to confirm what will be sent to the web service.
If changes are made to any base SOAP settings, the content of the SOAP XML is re-generated and your changes become lost.

Manually Enter XML

The following is an example of a Complex argument and would be placed in the Parameters field. SAM automatically takes the code and place it where it belongs in the envelope:

```xml
<tempPhoneNumber xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns2:Array" ns2:arrayType="ns1:TemporaryPhoneNumber[3]">
  <item xsi:type="ns1:TemporaryPhoneNumber">
    <startDate xsi:type="xsd:int">37060</startDate>
    <endDate xsi:type="xsd:int">37064</endDate>
    <phoneNumber xsi:type="xsd:string">+1-212-5551234</phoneNumber>
  </item>
</tempPhoneNumber>
```
Understand application monitor thresholds

Thresholds are markers that indicate certain levels have been reached. Many component monitors used in SAM application monitors have thresholds that can be adjusted and set to act as trigger points. For example, if monitoring CPU usage, a Critical threshold set at 90% would be typical. You can use this threshold as a trigger for SAM to send you an email alert when a Critical status threshold is breached.

In general, you need to monitor applications for several weeks to collect enough data to use as a baseline. After you establish a baseline, you can make an educated guess about how to set Warning and Critical thresholds for component monitors.

If thresholds are set too low, you'll receive frequent alerts. If set too high, problems can occur without your knowledge.
Baseline data, as well as Warning and Critical thresholds for application monitors, can also be gathered and calculated automatically, as described in Adjust threshold settings and apply baseline data. The option to enter thresholds manually remains available.

By default, I/O thresholds of Windows Service Monitors are not set.

For more information, see:

- Thresholds in the Orion Platform (Orion Platform Administrator Guide)
  - Set general thresholds
  - Customize thresholds for single objects
  - Baselines and baseline calculations
  - Define baselines for nodes
  - General threshold types
- Inheriting thresholds in SAM
- Multi-Value scripts and thresholds
- Apply baseline thresholds at the template level
- Apply baseline thresholds at the application level
- Use Min/Max Average Statistic charts to set thresholds based on 95th percentile data (Success Center)
- View latest baseline details
- Baseline thresholds (THWACK)

If an application is slow, you can use performance thresholds and baselines in SAM to identify the current problem and isolate it to an infrastructure, application, or database issue.

Did you know you can use SolarWinds Database Performance Analyzer (DPA) to identify specific queries that contribute to an issue? With DPA, you can use histogram analysis to find the offending query and dissect it with response-time analysis.

Inheriting thresholds in SAM

Thresholds can be adjusted at the template level or on the individual component monitor level of an application.
Threshold Adjustments at the Template Level

If adjusting thresholds on the template level, either manually entered or calculated using baseline data, any changes made trickle down to the component monitor level for applications based on this template. This means the component monitors within the application inherit the changes made in the template.

Threshold Adjustments at the Component Monitor Level

If you change thresholds at the component monitor level, only thresholds of the individual component monitor are affected. Thresholds on the parent template, or other applications based on that parent template, are not be affected.

Multi-Value scripts and thresholds

Each component monitor includes a number of thresholds for pulling and saving data. To use these values in multi-value scripts, you need to use the following threshold macros:

- ${Threshold.Warning.DisplayName} - Macro for using the display name for a warning threshold
- ${Threshold.Critical.DisplayName} - Macro for using the display name for a critical threshold
The following screenshot displays the use of the macros in a script.

![Screenshot of script output](image)

The following screenshots display the multiple display names used in a script for the thresholds values for warning and critical.

![Screenshot of script output](image)

Apply baseline thresholds at the template level

Applying and editing thresholds at the template level affect all applications that are based on that template. Thresholds can be used based on calculated baseline data or you can create your own thresholds based on your needs.

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click Settings > All Settings > SAM Settings > Manage Templates.
2. Select a template and click Edit from the toolbar.
3. Select a component monitor, click [+ ] to expand the monitor details.

4. Select Use thresholds calculated from baseline data:

Once this box is checked, the Warning and Critical fields automatically populate with the macro, \( \text{${USE_BASELINE}$} \). This macro can be used when configuring alerts.

5. Select the options for sustained thresholds, and then click Submit.

You can edit multiple component monitors and their thresholds if the monitors are the same type and thresholds are available. Multi Edit only become available when the selected component monitors are the same type, as shown. After you click Multi Edit, select Statistic Threshold > Use thresholds calculated from baseline data. Edit thresholds, click Save > Submit.
By default, I/O thresholds of Windows Service Monitors are not set.

Apply baseline thresholds at the application level

To customize your monitoring, you can apply baseline thresholds at the application level. The Orion Web Console has a set of global thresholds that apply to all monitored nodes. Using the following instructions, you can modify those thresholds per specific node.

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click My Dashboards > Applications > SAM Summary.
2. From the All Applications widget, expand the tree and then click an application.
3. From the Application Details widget, click Edit Application Monitor.
4. If selecting only one Component Monitor, click [+ ] to expand the monitor details.
   a. Click Override Template.
      The current values for the thresholds appear in the Warning and Critical fields.
   b. Click Use Latest Baseline Thresholds, as shown:

      ![Response Time Threshold](image)

      c. When applied, the values change and a blue icon appears indicating that baseline thresholds are being used.

      ![Statistic Threshold](image)

5. If selecting more than one component monitor, use the check boxes to select the monitors you want to edit, and then click Multi-Edit.
a. Click Multi-Edit.

Multi-Edit only becomes available when the selected component monitors are of the same type.

b. Check the Statistic Threshold check box on the pop-up window, then check the Use thresholds calculated from baseline data.

Once the second box is checked, the Warning and Critical fields automatically populate with the macro, \${USE_BASELINE}.

6. Click Save > Submit.

Adjust threshold settings and apply baseline data

In general, baseline data is calculated on demand; however, seven days of data is the recommended minimum amount of data needed for baseline calculations to be considered accurate. Baseline data for macros, such as \${USE_BASELINE}, are automatically calculated during nightly database maintenance.

After thresholds are calculated and applied to component monitors, the thresholds remain static until manually re-applied. This is not a moving baseline that is calculated nightly based on the last seven days of data. A moving baseline would mask data spikes and other anomalies that need to be highlighted.

Thresholds set manually to meet the needs of your environment may yield more desirable results.

How SAM calculates the baseline

SolarWinds SAM uses the following macros to calculate the baseline:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
</table>

page 285
Currently used baseline in SAM, used in threshold fields of SAM component monitors. Recommended baseline values are calculated using the following formulas:

**Warning:** ${\text{MEAN} + 2 \times \text{STDDEV}}$ (or ${\text{MEAN} - 2 \times \text{STDDEV}}$)

**Critical:** ${\text{MEAN} + 3 \times \text{STDDEV}}$ (or ${\text{MEAN} - 3 \times \text{STDDEV}}$)

Baseline thresholds are not suitable for all metrics. If calculated values do not meet expectations, consider setting the thresholds manually.

${\text{USE_BASELINE}}$ does not support math functions. To adjust the baseline calculation, replace ${\text{USE_BASELINE}}$ with the formulas above and change the calculation as necessary.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
</table>
| ${\text{USE_BASELINE}}$ | Currently used baseline in SAM, used in threshold fields of SAM component monitors. Recommended baseline values are calculated using the following formulas: **Warning:** ${\text{MEAN} + 2 \times \text{STDDEV}}$ (or ${\text{MEAN} - 2 \times \text{STDDEV}}$) **Critical:** ${\text{MEAN} + 3 \times \text{STDDEV}}$ (or ${\text{MEAN} - 3 \times \text{STDDEV}}$)
| ${\text{MEAN}}$ | Current Mean or Average. You can use this macro with math functions in threshold fields. |
| ${\text{STDDEV}}$ | Standard Deviation. You can use this macro with math functions in threshold fields. |

Change the amount of data used in baseline calculations

You may need to log in as an administrator to perform this action.

1. Click **Settings > All Settings > SAM Settings > Data & Database Settings**.

   ![Data & Database Settings](image)

   - **Baseline Data Collection Duration**: Specify how many days of data to include in the baseline.

2. Enter a number of days, and then click **Submit**.

   The value for the Baseline Data Collection Duration field cannot exceed the value defined for the Detail Statistics Retention field, as displayed at the top of the Data & Database Settings section.

Apply baseline thresholds at the template level

You may need to log in as an administrator to perform this action.

1. In the Orion Web Console, click **Settings > All Settings > SAM Settings > Manage Templates**.
2. Check a template and click **Edit** from the toolbar.
3. If selecting only one component monitor, click [+] to expand monitor details.
   a. Click Use Latest Baseline Thresholds to display the \( \text{USE\_BASELINE} \) macro in the Warning and Critical fields.
   b. Select options for sustained thresholds as needed.

![Thresholds Expansion]

4. If selecting more than one component monitor, select monitors by checking the boxes next to their names.
   a. Click Multi-Edit.

   > Multi-Edit is only available if selected component monitors are the same type.

   b. Review Statistic Threshold data that on the pop-up window, then check the Use thresholds calculated from baseline data.

   > The Use thresholds calculated from baseline data check box is not available until Statistic Threshold is checked. After Use thresholds calculated from baseline data is checked, the Warning and Critical fields automatically populate with the macro, \( \text{USE\_BASELINE} \).

5. Click Save, then click Submit.

Apply baseline thresholds at the application level

> You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click My Dashboards > Applications > Summary.
2. From the All Applications widget, expand the tree and then click an application.
3. From the Application Details widget, click Edit Application Monitor.
4. If selecting only one Component Monitor, click [+] to expand the monitor details.
   a. Click Override Template.
      
      "The current values for the thresholds appear in the Warning and Critical fields."

   b. Click Use Latest Baseline Thresholds, as shown:
      
      ![Image of Response Time Threshold]

   c. When applied, values change and a blue icon indicates that baseline thresholds are used.

5. If selecting more than one component monitor, use the check boxes to select the monitors you want to edit.
   a. Click Multi-Edit.
      
      "Multi-Edit is only available if selected component monitors are the same type."

   b. Check the Statistic Threshold check box on the pop-up window, then check the Use thresholds calculated from baseline data.
      
      "Once the second box is checked, the Warning and Critical fields automatically populate with the macro, \( \text{$\{USE\_BASELINE\}$}."

6. Click Save, then click Submit.
Use latest baseline details

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click My Dashboards > Applications > Summary.
2. From the All Applications widget, expand the tree and then click an application.
3. From the Application Details widget, click Edit Application Monitor.
4. Find a component monitor in the list and click [+] to expand the monitor details.
5. Click Override Template, then click Latest Baseline Details.

<table>
<thead>
<tr>
<th>Warning:</th>
<th>greater than</th>
<th>Critical:</th>
</tr>
</thead>
<tbody>
<tr>
<td>for a single poll</td>
<td></td>
<td>greater than</td>
</tr>
<tr>
<td>for a single poll</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Edit capacity planning thresholds

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click a node to navigate to the Node Details page for that node.
2. In the Management widget, click Edit Node.
3. Click Manage Orion General Thresholds to display Capacity Planning options appear under each object:

> Avg CPU Load

<table>
<thead>
<tr>
<th>Critical Level</th>
<th>90</th>
<th>1% to 100%</th>
<th>Nodes with CPU Load above this level will</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning Level</td>
<td>80</td>
<td>1% to 100%</td>
<td>Nodes with CPU Load above this level will</td>
</tr>
</tbody>
</table>

| Capacity Planning | Calculate exhaustion using average daily values | Calculate exhaustion using peak daily values |

4. Make your selections, then click Submit on both the Orion General Thresholds screen and the Edit Properties screen.

View latest baseline details

Details about how baseline data, as well as Warning and Critical thresholds are calculated, can be found on the Latest Baseline Details page. This page details the data collection and calculation process using several graphs and tables.

View the Latest Baseline Details Page

You may need to log in with an administrator account to perform this action.

1. In the Orion Web Console, click My Dashboards > Applications > SAM Summary.
2. In the All Applications widget, expand the tree and then click an application.
3. From the Application Details widget, click Edit Application Monitor.
4. Find a component monitor in the list and click [+] to expand the monitor details.
5. Click Override Template click Latest Baseline Details.

Use PowerShell in SAM

Some SAM templates include component monitors that use PowerShell scripts to gather information. You can also create custom scripts for SAM templates that support PowerShell scripts. SAM supports PowerShell 2.0—5.x.

Created by Microsoft, PowerShell is a task automation and configuration management framework that consists of a command-line shell and associated scripting language, built on the .NET Framework. PowerShell scripts are used in several out-of-the-box SAM templates, including Log Parser and Office 365 templates.

The ability to deploy PowerShell scripts to remote machines within SAM is a powerful advantage for system administrators. With an interactive prompt and scripting environment, PowerShell provides access to the file system on remote computers, along with data stores such as the registry.

Your organization should internally review and assess to what extent PowerShell is incorporated into your environment. This is especially important when importing scripts from third parties, including content posted by other customers in the SolarWinds online IT community, THWACK. To learn more, see PowerShell security considerations.

PowerShell includes built-in commands with a consistent interface (as shown here), plus an Integrated Scripting Environment (ISE) host application.

![PowerShell](https://example.com/powershell.png)

PowerShell does not process text; it processes objects based on the .NET Framework.

PowerShell also includes default cmdlets — simple commands you can use to manipulate objects — and you can write your own cmdlets. Cmdlets have a unique format — a verb and noun separated by a dash (-), such as Get-Help. You can use them separately or combine them in scripts that perform complex tasks.

SolarWinds provides customer support for PowerShell scripts and functionality built into SAM, but not for scripting languages or custom scripts. For scripting support from the SolarWinds online IT community, visit THWACK.
Some ways you can use PowerShell scripts with SAM include:

- Automatically deploy a Windows agent to established instances on Amazon Web Services.
- Monitor Office 365 services. See Using Microsoft Office 365 templates in the Success Center.
- Create a Windows PowerShell component monitor.

Review these topics to learn more:

- PowerShell security considerations
- PowerShell requirements
- Enable remote access for PowerShell with WinRM
- Working with Exchange Management Tools
- Learn more about PowerShell

PowerShell security considerations

Depending on how you configure PowerShell in your environment, it can make your system vulnerable to unauthorized access. For example, running monitors as Local Host on the Orion server with Admin privileges gives scripts unlimited power. Your organization should internally review and assess to what extent PowerShell scripts will be incorporated into your environment.

PowerShell can be used with the Orion Platform in many ways. For example, you can use it with the Orion Software Development Kit (SDK) to perform IP Address Manager (IPAM) operations, such as creating subnets. When used in SAM component monitors and application templates, PowerShell provides the ability to:

- Access file systems on computers.
- Access data stores, including the system registry.
- Deploy scripts to run on multiple remote machines.

While PowerShell enhances SAM functionality, it's important to consider the security risks inherent in using PowerShell scripts. SolarWinds recommends the use of a dedicated Windows account with low-level privileges for PowerShell monitors, especially for scripts executed on a polling engine that uses the same Windows account as the Orion server.

You can also avoid security risks, such as malicious OS command injections, by using PowerShell's built-in security, as described in Microsoft PowerShell documentation (© 2018 Microsoft Corp., link available at https://docs.microsoft.com/en-us/powershell/, obtained on October 30, 2018).

PowerShell requirements

Following are PowerShell requirements for a typical SAM environment:

- **PowerShell version**: Version 1.0 or later is required for local execution. For remote execution of scripts, PowerShell 2.0 or later is required on the main polling engine (that is, the Orion server), Additional Polling Engines (APEs), and target servers. PowerShell is included in most installations of
Microsoft Windows Server or you can install it, if necessary. See Microsoft PowerShell Documentation.

Some scripts require PowerShell 2.0 to execute certain actions. Later versions of PowerShell shipped with Windows Server include a backwards compatibility mode you can enable to run 2.0 with a later version.

Another consideration is whether to use 32-bit (x86) or 64-bit (x64) PowerShell. For best results, match the PowerShell version to the OS and application configurations on a server. For example, on a 64-bit main polling engine that polls a 64-bit server, install PowerShell 64 bit (x64).

- **Accounts and permissions:** Local Admin rights are required to run scripts on the Orion server. To execute scripts on target servers, select a Windows credential with rights to log into the Orion server plus sufficient rights on the target node to execute tasks in the script. For example, if a script does something with WMI, the credentials also need WMI rights on the target node.

  SolarWinds recommends using a dedicated Windows account with minimal privileges for PowerShell monitors, especially for scripts executed on the main polling engine.

Without the correct permissions for a target server, scripts return an Unknown status.

- **Microsoft .NET Framework:** Many PowerShell scripts require .NET 3.5.x but the latest Orion Platform products include later versions during installation. For example, SAM 6.7 includes .NET 4.6.2. If necessary, use Server Manager's Add Roles and Features wizard to add .NET Framework 3.5.x.

- **Remote access:** To use Remote Host as the Execution Mode for a PowerShell script, enable the Windows Remote Management (WinRM) service on the Orion server so it can access remote target servers, as described next.

### Enable remote access for PowerShell with WinRM

If you select Remote Host as the Execution Mode for a SAM component monitor, the WinRM service must be enabled and properly configured on the main Orion server. WinRM cannot be enabled on target servers remotely, but you can configure the Orion server to grant permission for PowerShell to access the target servers.

There are several automated ways to enable remote access for PowerShell on servers:

- Use the free [Remote Execution Enabler for PowerShell](https://support.solarwinds.com/ Knowledgebase/Article/View/36253/1/Remote-Execution-Enabler-for-PowerShell) tool. See this [THWACK post](https://thwack.solarwinds.com) for instructions.
- Add the server as a new node. In the Orion Web Console, click Settings > All Settings > Add Node.
- Configure the target server as an [AppInsight for IIS node](https://support.solarwinds.com/Knowledgebase/Article/View/36240/1/AppInsight-for-IIS) via the Node Details page.

You can also enable remote access for servers in the PowerShell console, as described next.

1. On the Orion server and each remote server you want to run PowerShell on:
   a. Change the startup type for the WinRM service to Automatic.
   b. Start the WinRM service.
   c. Run the `get-service winrm` PowerShell command to verify WinRM is running.

3. In the PowerShell console, run:
   ```powershell
   Set-PSSessionConfiguration Microsoft.PowerShell -ShowSecurityDescriptorUI -force
   ```

4. Enable the "Full Control for everyone" option.

5. Verify the group to which the polling user belongs can access PowerShell.

6. Repeat these steps for all remote target servers.

When finished, each remoting server should include the following elements:

- An SSL certificate
- A WinRM Listener
- A firewall exception to allow outside requests to reach the WinRM service;
- The WinRM service to receive requests from other computers.

**Working with Exchange Management Tools**

Many SAM PowerShell components rely on Exchange Management tools.

To monitor a specific version of Microsoft Exchange:

- Use the corresponding version of Exchange Management Tools.
- Install Exchange Management Tools on the Orion server and any additional polling engines that use PowerShell scripts. Any snap-ins that PowerShell may require must also be installed in the same location.

💡 To ease PowerShell plug-in management in an environment with multiple polling engines, consider assigning all nodes with PowerShell templates to a single polling engine.

**Learn more about PowerShell**


**Troubleshoot PowerShell issues**

This section provides various troubleshooting tips for issues you may encounter when using SAM with PowerShell. You can find additional information in the following resources:

- [Use PowerShell in SAM](https://community.solarwinds.com/t5/Thwack-Territory/Use-PowerShell-in-SAM/td-p/939556)
- The [SolarWinds Success Center](https://community.solarwinds.com/t5/Thwack-Territory/SolarWinds-Success-Center/td-p/939558)
"PowerShell 2.0 was not detected on the Exchange server" message

The Orion server uses Windows PowerShell 2.0 or later to run PowerShell commands on remote machines. If the Orion server cannot execute PowerShell commands on a Microsoft Exchange server, you may receive error messages indicating that a request could not be processed, the content type is absent/invalid, or remote configuration failed.

In SAM, AppInsight for Exchange uses PowerShell 3.0 or later, depending on the SAM version. PowerShell 3.0 may not be able to run PowerShell cmdlets for Exchange due to a broken registry key, depending on the operating system and Microsoft Exchange versions. This is a known Microsoft issue; see Installing the Windows PowerShell 2.0 Engine.

Microsoft Server installations usually include PowerShell version 3.0 and later with backward compatibility support. However, some scripts and programs may be incompatible with a particular PowerShell version and return error messages mentioned above in the Overview.

To learn more, see PowerShell 2.0 was not detected on the Exchange server in the Success Center.

Troubleshoot the Windows PowerShell Monitor

If you encounter connection- or authentication-related error messages when using the Windows PowerShell Monitor, verify that you are entering the user name and password of an administrator-level account. If you think you have entered an incorrect credential, correct the mistake and then retest.

Make sure that the RPC server is available and that the Windows PowerShell execution policy is not set to Restricted (the default setting). You can check the execution policy by entering the following command at the PowerShell command prompt: Get-ExecutionPolicy

If you encounter issues using the Windows PowerShell Monitor, visit the SolarWinds Success Center and see the following article for tips: https://support.solarwinds.com/success_center/Server_%26_Application_Monitor_(SAM)/Problems_using_the_Orion_APM_Windows_PowerShell_Monitor

If PowerShell was installed on Windows Server 2012 with Exchange 2013 and subsequently uninstalled, a Microsoft error removes the required registry key for remote PowerShell to work properly. Security patches or updates may also remove registry keys. If you encounter this issue, see steps for recreating the required registry key in Set PowerShell permissions for Exchange.