Microsoft: Windows Server PowerPack Release Notes

Version 119

Overview

Version 119 of the "Microsoft: Windows Server" PowerPack adds a number of new Dynamic Applications, updates multiple existing Dynamic Applications, and addresses multiple issues.

Minimum Required Skylar One (SL1) Version: 12.3.1

This document covers the following topics:

Before You Install or Upgrade

Ensure that you are running version 12.3.1 or later of Skylar One before installing the "Microsoft: Windows Server" PowerPack.

NOTE: For details on upgrading Skylar One, see the relevant *Skylar One Platform Release Notes*.

Installing or Upgrading to this Version

TIP: By default, installing a new version of a PowerPack will overwrite all content in that PowerPack that has already been installed on the target system. You can use the ***Enable Selective PowerPack Field Protection*** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent the new version of the PowerPack from overwriting local changes for some commonly customized fields.

NOTE: If you are currently using the Dynamic Applications in this PowerPack to monitor devices, collection errors might occur for one or two polling cycles during the installation of a new version. To prevent collection errors during an upgrade, you can optionally disable collection for monitored devices before performing the following steps and re-enable collection after the upgrade.

To install this PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site.
2. In Skylar One, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **Actions** menu and choose *Import PowerPack*. The **Import PowerPack** modal appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPack Manager** page.

WARNING: The internal collections Dynamic Applications (ICDAs) for Port, Process, and Service monitoring are disabled by default. If you are currently using the ICDAs to monitor devices, they will be disabled again after installing a new version of this PowerPack. ScienceLogic recommends that you use **Enable PowerPack Field Protection** or re-enable them after the installation is completed.

For more information about using the PowerPack, see the *Monitoring Windows Systems with PowerShell* manual.

Features

This release includes the following features:

- Dynamic Applications that collect configuration and performance data about Windows Servers
- Event Policies that are triggered when Windows Server devices meet certain status criteria
- Device Classes for each type of Windows Server

NOTE: The Device Classes include older device types that can be discovered but are no longer supported by ScienceLogic.

- Run Book Policies and Run Book Actions that align a more detailed device class with each discovered device
- A sample Credential for discovering Windows Server devices
- A Credential Test to ensure that your Windows credential works as expected

Enhancements and Issues Addressed

The following enhancements and addressed issues are included in this release:

- Added the new "Microsoft: Windows Server CPU Average Performance" Dynamic Application to provide an alternate method for calculating the average CPU performance between polling cycles instead of the "Microsoft: Windows Server CPU Performance" Dynamic Application.

> **NOTE:** The "Microsoft: Windows Server CPU Average Performance" Dynamic Application is disabled by default and is not currently included in templates or run book automation policy alignment. To reduce monitoring load, provide correct CPU vitals, and prevent duplicate alerts, only one of the two Dynamic Applications ("Microsoft: Windows Server CPU Average Performance" and "Microsoft: Windows Server CPU Performance") should be aligned to a device at one time.

- Added the new "Microsoft: Windows Server Thread Status Performance" Dynamic Application, which monitors the counts of threads in each possible status, along with a total process count. This Dynamic Application is enabled by default, but will only align automatically during a dynamic discovery. If you want to use it with existing monitored Windows devices, it must be manually aligned to those devices.
- Added the new "Microsoft: Windows Server Certificate Configuration" Dynamic Application to provide information and alerts about expiring certificates.
- Added the new "Microsoft: Print Server Configuration" Dynamic Application, which allows you to monitor printers, ports, and print jobs.
- Added the new "Microsoft: Windows Server Large Open Files" Dynamic Application which provides information on the top 20 open files by size per logical disk.

> **NOTE:** This Dynamic Application is disabled by default and requires additional setup to be monitored. For more information, see the "Configuring the Microsoft: Windows Server Large Open Files Dynamic Application" section in the Monitoring Windows Powershell manual.

- Renamed the "Microsoft: Print Server" Dynamic Application to "Microsoft: Print Server Performance".
- Run book action policy logs are now written to a dedicated file at **/var/log/em7/rba_windows_server.log** to simplify troubleshooting Windows Server automation actions. This change affects the following run book action policies:
    - Microsoft: Windows Server Check Concurrent (Collector)
    - Microsoft: Windows Server Device Class Alignment
    - Microsoft: Windows Server Dynamic Application Alignment
    - Microsoft: Windows Server Unselect Dynamic Discovery
- Updated the "Microsoft: Windows Server IC Cache Trigger" Dynamic Application to prevent it from saving data for all of its collection objects. This change was implemented due to storage issues caused by the Dynamic Application where it generated a large amount of unused data. (Case: 00389374)

- Updated the "Microsoft: Windows Server CPU Performance" and "Microsoft: Windows Server CPU Average Performance" PowerShell requests to use the "Processor Information" counter set instead of "Processor" on servers where "Processor Information" is available. If the new option is not available, the requests will fall back to use "Processor" instead.
- Updated the classification alert formula for the "Microsoft: Windows Server Device Discovery" Dynamic Application to address an issue that caused prolonged device classification times due to the use of the `Prior()` function in alert formulas.(Cases: 00442248, 00467630, 00525210)
- Added invalid data checks to the "Microsoft: Windows Server Filesystem Inventory" and "Microsoft: Windows Server Filesystem Performance" Dynamic Applications to address an issue that caused unhandled exception errors. (Case: 00450992)
- Added a discovery snippet to the "Microsoft: Windows Server Device Discovery" Dynamic Application that aligns it on the first pass. That alignment triggers the new "Microsoft: Windows Server Check Concurrent" and "Microsoft: Windows Server Dynamic Application Alignment" run book action policies, which will align the cache consumer Dynamic Applications using run book action policies. (Case: 00534741)
- The "Microsoft: Windows Server Device Discovery" Dynamic Application now triggers the "Microsoft Windows Server Device Found" event, which now triggers the "Microsoft: Windows Server Device Class Alignment" run book action policy. As a result, devices are classified as Windows servers much sooner after discovery.
- Updated the PowerShell request to improve performance of the "Microsoft: Windows Server Software Configuration" Dynamic Application, which reduces agent CPU usage during software inventory collection. (Case: 00626107)
- Updated PowerShell requests calling the "Get-Counter" cmdlet that previously required English language counters to support other languages.
- Removed the "Discovery Object" collection object from the "Microsoft: Windows Server IC Cache Trigger" Dynamic Application to prevent automatic alignment when Windows devices are discovered with Powershell disabled.
- Moved the "Microsoft: Windows Server Unselect Dynamic Discovery" run book action policy from the "Microsoft: Windows Server Device Class Alignment" run book automation policy to the new "Microsoft: Windows Server Dynamic Application Alignment" run book automation policy.
- Added the new "CPU Work Utilization" presentation object to the "Microsoft: Windows Server CPU Performance" Dynamic Application to measure the percentage of CPU time spent doing actual productive work, excluding system overhead.

- Replaced all "Get-WMIObject / GWMI" calls with "Get-CimInstance / GCIM" calls in the following Dynamic Applications:
    - Microsoft: Windows Server Configuration Cache
    - Microsoft: Windows Server Performance Cache
    - Microsoft: Windows Server IC Process Service Cache
    - Microsoft: Windows Server Memory Configuration
    - Microsoft: Windows Server Print Server
    - Microsoft: Windows Server BIOS Configuration
- Updated the "Microsoft: Windows Server OS Configuration" Dynamic Application to collect data about the key management service (KMS), including the activation type, KMS machine name, and port.
- Updated the alert formulas in the "Microsoft: Windows Server Certificate Configuration" Dynamic Application so that non-CA certificate alerts only trigger for certificates in the "My" store, which reduces the amount of alerts triggered for "Root", "AuthRoot", and "TrustedPublisher" store certificates.
- Updated the following Dynamic Applications to run their own PowerShell instead of consuming a cache:
    - Microsoft: Windows Server IPStats Performance
    - Microsoft: Windows Server TCPStats Performance
    - Microsoft: Windows Server UDPStats Performance
- Removed the following PowerShell requests from the "Microsoft: Windows Server Performance Cache" Dynamic Application:
    - Windows Server Interface IPv4 Statistics Performance
    - Windows Server Interface IPv6 Statistics Performance
- Improved the efficiency of the PowerShell requests of the following Dynamic Applications:
    - Microsoft: Windows Server BIOS Configuration
    - Microsoft: Windows Server Memory Configuration
- Improved the efficiency of the following PowerShell requests in the " Microsoft: Windows Server Performance Cache" Dynamic Application:
    - Windows Server Details - ICDA
    - Windows Server Disk Capacity Performance
- Improved the efficiency of the "Windows Server Server CPU Configuration" PowerShell request in the "Microsoft: Windows Server Configuration Cache" Dynamic Application.
- Improved the efficiency of the "Windows Server Port Details - ICDA" PowerShell request in the "Microsoft: Windows Server IC Process Service Cache" Dynamic Application.

- Added the new "Current Load" presentation object to the "Microsoft: Windows Server CPU Performance" Dynamic Application, which shows the server load based on processor queue length and the number of CPU cores.

Known Issues and Workarounds

The following known issues affect version 119 of the "Microsoft: Windows Server" PowerPack:

- Due to Skylar One platform behavior, ScienceLogic recommends you manually update the threshold values for the ***SL1 Agent for Microsoft: Windows Server Template*** device template after installing the PowerPack.
- Windows Server Process and Service monitoring will not work in environments in which Constrained Language mode is enabled for PowerShell.
- If the list of IP addresses assigned to an interface is longer than 235 characters, the "Microsoft: Windows Server Interface Configuration" Dynamic Application will strip the list of IP addresses after 235 characters and the following will appear in the logs:

```
90.PoolWorker-2.Extended_Internal_Collection: Skipping IPv6 Address
due to powershell collector characters limit of 235 chars. did:
<did> app_id: <app_id> Interface: <if>90.PoolWorker-2.Extended_
Internal_Collection: Skipping IPv4 Address due to powershell
collector characters limit of 235 chars. did: <did> app_id: <app_
id> Interface: <if>If an IPv4 address is invalid, the "Microsoft:
Windows Server Interface Configuration" Dynamic Application will
remove the address and the following notice will appear in the
logs:90.PoolWorker-2.Extended_Internal_Collection: Skipping IPv4
Mask due to powershell collector characters limit of 235 chars.
did: <did> app_id: <app_id> Interface: <if>
```

- When updating the PowerPack, in the "Microsoft: Windows Server IC Interface Inventory" Dynamic Application, corrupted IPs with empty spaces, curly brackets ({}), or ellipses (...) will need to be deleted. Review the rows that will be deleted with the following query in the SL1 database:

```
SELECT id, did, ip, netmask FROM master_dev.device_ip_addr WHERE ip
REGEXP '(^([{])|([}]$)|([.]{3}$)|([...}]$)|(\\s))';
```

  This will display all rows that will be deleted in the next step.

  Use the following query in the SL1 database to delete the rows:

```
DELETE FROM master_dev.device_ip_addr WHERE ip REGEXP '(^([{])|
([}]$)|([.]{3}$)|([...}]$)|(\\s))';
```

  The deletion cannot be reversed after running this query.

- For mount point paths, all instances of "\" have been changed to "/" in the "Windows: Server IC Filesystem Inventory" and "Windows: Server IC Filesystem Performance" Dynamic Applications. Drives that are hidden will not be loaded, but will be visible in the "Microsoft: Windows Server Disk Configuration" Dynamic Application.
- The ***Collector Affinity*** setting for Windows Server internal collections Dynamic Applications (ICDAs) changes to *Default* if there are any changes made under the **[Properties]** tab. To keep the setting as *Assigned Collector*, run the following query:

```
UPDATE master.dynamic_app SET cu_affinity=2 WHERE ppguid IN ('<PP-GUID>');
```

- The "Microsoft: Windows Server Software Configuration" Dynamic Application cannot properly parse installation dates that are not in yyyy-mm-dd hh:mm:ss format, such as "Wed Jul 05 12:41:46 EDT 2017".

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010