Monitoring with the Skylar One Agent

Skylar One version 12.5.23

Table of Contents

Introduction to the Skylar One Agent **6**

What is the Skylar One Agent? 7

Agent Versions for Skylar One 7

How the Skylar One Agent Works in Different Environments 7

What System Vital Metrics Can the Skylar One Agent Collect? 10

Metric Descriptions 10

Supported Data Collection Methods for Monitoring Windows 12

Supported Data Collection Methods for Monitoring Linux 13

Extensible Collection 13

Agent PowerPacks 14

Agent-Compatible PowerPacks 16

Windows Devices 16

Java Management Extensions (JMX) Resources 16

Agent Architecture 16

Skylar One Distributed Architecture 16

Skylar One Extended Architecture 17

Installing the Skylar One Agent **18**

Caveats 19

Installing an Agent from the Agents Page 19

Installing a Linux Agent 21

Using Agents with SELinux 23

Installing an AIX Agent 23

Installing a Solaris Agent 25

Installing a Windows Agent 26

Configuring the Windows Agent Using the Command Line 28

Configuring the Windows Agent to Run as Local System 29

Configuring the Windows Agent to Run as a Dedicated User 29

Example Install Commands 30

Example Upgrade Commands 31

Troubleshooting the Windows Installation Agent 32

Additional Considerations 32

Authenticating the Agent 33
Configuring the Streamer 33
Installing an Agent with a Token 33
Upgrading an Existing Agent to an Agent that Supports Tokens 33
Installing or Upgrading without Tokens 34
Windows Installation Commands (Example) 34
Linux Installation Commands-Debian and Ubuntu (Example) 34
Linux Installation Commands-RHEL, CentOs, and AIX (Example) 35
Generating GUIDs and Tokens Using GQL 35
Configuring a Skylar One Agent to Use a Proxy Server Connection 36
Configuring a Linux Agent to Use a Proxy 37
Installing and Configuring with the Proxy Settings 40
Configuring a Windows Agent to Use a Proxy 40
Installing and Configuring an Agent with the Proxy Settings 43
Upgrading an Agent 43
Stopping an Agent 46
Uninstalling an Agent 47
Uninstalling a Linux Agent 47
Uninstalling an AIX Agent 47
Uninstalling a Solaris Agent 48
Uninstalling a Windows Agent 48
Installing an Agent from the Devices Page or the Device Manager Page 48
Installing an Agent on a Linux System 50
Upgrading an Agent on a Linux System 50
Installing an Agent on a Windows System 50
Configuring a Skylar One Agent to Work with PhoneHome Collectors 50
Remotely Installing a Skylar One Agent 52
Remotely Installing a Linux Agent 52
Remotely Installing a Windows Agent 56
Configuring the Skylar One Agent 62
System Settings that Affect Agents 63
Configuring Agent Monitoring Based on Skylar One Architecture 63

Configuring Agent Settings from the Device Investigator 65
Configuring an Agent to Export Data to Skylar AI 68
Enabling Skylar AI Export Using Deploy 68
Manually Enabling Skylar AI 69
Validating the Skylar AI Export 70
Configuring Extensible Collection 71
Configuring Run Book Automations for a Skylar One Agent 73
Action Type 73
Configuring a Run Book Automation for an Agent Device 74
Configuring a Skylar One Agent on the Device Manager Page 76
Adding the "SL Agent" Column to the Device Manager Page 76
Configuring Agent Settings on a Device 77
Monitoring Logs Using the Skylar One Agent 78
What is a Log File Monitoring Policy? 79
Viewing the List of Log File Monitoring Policies 79
Creating a Log File Monitoring Policy 80
Aligning a Log File Monitoring Policy to Devices 81
Unaligning Log File Monitoring Policies from Devices 83
Editing a Log File Monitoring Policy 83
Deleting Log File Monitoring Policies 83
Viewing the List of Log File Monitoring Policies and Aligned Devices 84
Creating an Event Policy for Agent Logs 85
Creating an Event Policy for Agent Logs in the Classic User Interface 87
Skylar One Agent Troubleshooting 90
Was the Agent Download Successful? 91
Is the Windows Installation or Upgrade Failing? 91
Is the Agent Process Running? 91
Is the Agent Configuration File Valid? 92
Has Skylar One Discovery Completed? 93
Is the Agent Able to Upload Data? 93
Check the Agent Upload Directory 93
Run the Agent in Debug Mode (Linux) 94

Is Skylar One Receiving Agent Data? 95
Is the Agent Not Reporting Vital Data and Metrics? 95
Can Skylar One Process Agent Data? 95
Is the Number of Processes Inconsistent with Other Applications? 96
Is the Agent Unable to Connect to the Streamer Endpoint? 96
Validating Agent TLS Connections to the Skylar One Streamer Service 98
Troubleshooting Examples 99
Example /var/log/streamer_prime/streamer_prime.log for successful discovery 99
Example /var/log/uwsgi/streamer.log for successful discovery in the Skylar One Distributed Architecture 101
Save incoming data for a specific device ID in the Skylar One Distributed Architecture 101
Save incoming data for a specific device ID in the Skylar One Extended Architecture 101
Additional Troubleshooting Situations and Best Practices 101
Agent Communication with Skylar One 102
Return Codes 102
Sub-codes 102

Chapter 1

Introduction to the Skylar One Agent

Overview

This chapter describes the Skylar One agent and provides instructions for viewing device and interface data collected by the agent.

> **NOTE:** If your current ScienceLogic Skylar One solution subscription does not include the Skylar One Agent, contact your ScienceLogic Customer Success Manager or Customer Support to learn more.

This chapter covers the following topics:

What is the Skylar One Agent? 7

Agent Versions for Skylar One 7

How the Skylar One Agent Works in Different Environments 7

What System Vital Metrics Can the Skylar One Agent Collect? 10

Agent PowerPacks 14

Agent Architecture 16

What is the Skylar One Agent?

The ***Skylar One agent*** is a program that you can install on a device monitored by Skylar One. There is a Windows agent, an AIX agent, a Solaris agent, and a Linux agent. The agent collects data from the device and pushes that data back to Skylar One.

Similar to a Data Collector or Message Collector, the agent collects data about infrastructure and applications.

You can configure an agent to communicate with either the Message Collector or the Compute Cluster.

Agent Versions for Skylar One

The following table shows the most recent Linux and Windows agent versions for each Skylar One release.

Skylar One (SL1) Version	Linux Agent Version	Windows Agent Version
12.5.21	Linux v198	Windows v160
12.5.20	Linux v198	Windows v155
12.5.1-12.5.8	Linux v196	Windows v154
12.3.15	Linux v198	Windows v160
12.3.14	Linux v198	Windows v155
12.3.7-12.3.13	Linux v196	Windows v154
12.3.1-12.3.6	Linux v193	Windows v150
12.3.0	Linux v191	Windows v148

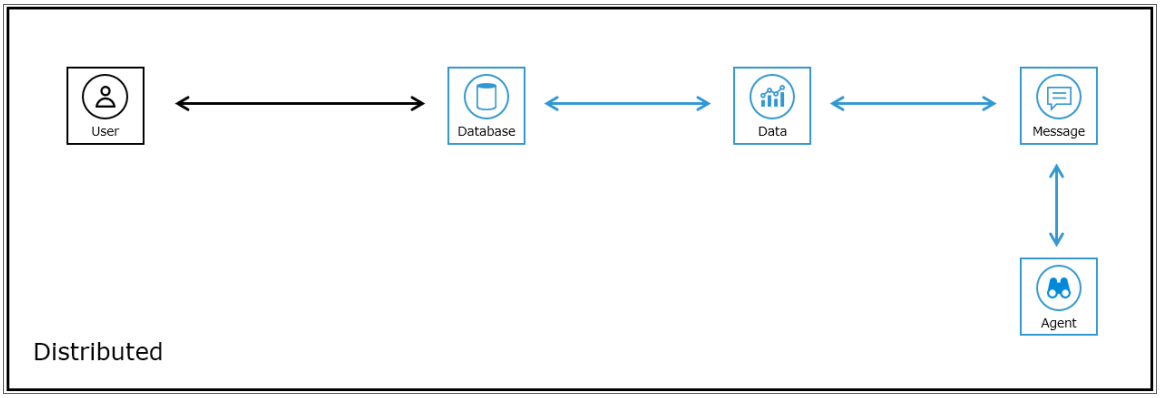
How the Skylar One Agent Works in Different Environments

There are two supported "generations" of the Skylar One agent that you can use to gather data: Gen 1 and Gen 3. The agent itself is the same from one generation to the next; the only difference between the generations is the *environment* where the agent is used.

In brief, a ***Gen 1 agent*** uses the Skylar One Distributed Environment to upload data directly to a Message Collector, while a ***Gen 3 agent*** uses the Skylar One Extended Architecture to upload data to the "Streamer" service running on the Skylar One Compute Node cluster.

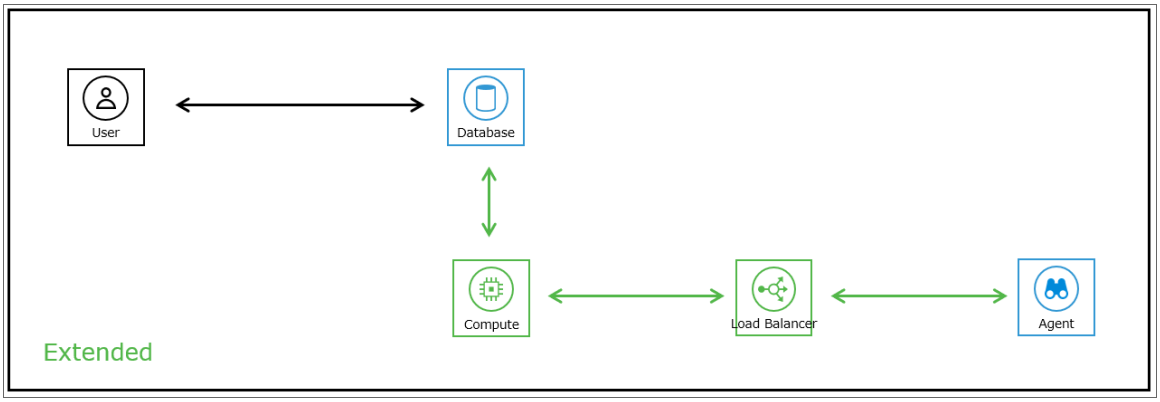
The following list provides more details about how Skylar One uses the different generations of the Skylar One agent:

- **Gen 1 agent**: Uses a Distributed Environment to upload data directly to a Message Collector. Next, Dynamic Applications running on a Data Collector query data from the Message Collector with an API. The Data Collector then sends the data to the Skylar One Database Server:



The Gen 1 agent provides limited infrastructure health reporting, including Log File Monitoring, Processes, and System Vitals like Availability, CPU Usage, Memory Usage, and Disk Usage. This agent is sometimes called the "P0" agent.

- **Gen 2 agent**: This version of the agent has been deprecated.
- **Gen 3 agent**: Introduced in Skylar One version 8.12.0, the Gen 3 agent uses the Skylar One Extended Architecture to upload data through a load balancer to the "Streamer" service running on the Skylar One Compute Node cluster. The Streamer service then forks and forwards data to other services, and eventually some services will store parsed data in the Skylar OneDatabase Server:



The Gen 3 agent provides full infrastructure health reporting, including system vitals (file system, network interface, and service data), log monitoring, and optionally allows you to align PowerShell Dynamic Applications to your device. Also, the Dynamic Applications in the *ScienceLogic: Agent* PowerPack are automatically aligned with the device with the Gen 3 agent installed on it.

> **NOTE:** The ***Collection Frequency*** column on the **Network Interfaces** page (Registry > Networks > Interfaces) is grayed out for Gen 3 agents because it does not apply to interfaces of agent-monitored devices. The agent interface collection data defaults to 5 minutes and is not configurable for Gen 3 agents.

If you are already running the Gen 1 agent in a distributed environment and you upgrade your Skylar One system to use the Skylar One Extended Architecture, any existing devices monitored by Skylar One agents will work the same as before (streaming data to a Message Collector). However, if you upgrade to the Skylar One Extended Architecture, ScienceLogic recommends that you start streaming your agents to the Compute Node cluster instead of to a Message Collector.

To identify the version of the agent installed on a device:

- **Gen 1**. On the **Devices > Device Investigator > [Settings]** tab for that device, the ***Collection Poller*** field displays the name of the collector group that includes the Message Collector used by the agent. On the **Device Manager** page (Devices > Classic Devices), the **Collection Group** column displays the name of the collector group that includes the Message Collector used by the agent.
- **Gen 3**. On the **Devices > Device Investigator > [Settings]** tab for that device, the ***Collection Poller*** field displays the label *Agents*. On the **Device Manager** page (Devices > Classic Devices), the **Collection Group** column displays the label *Agents*.

The following table provides an overview of the features available when using the Gen 1 or the Gen 3 agent:

Product Capability	Gen 1 Agent	Gen 3 Agent
System Vitals		
Availability and Uptime	Yes	Yes
CPU, Memory, File Systems, and Network Interface	CPU and Memory only	Yes
Processes and Services	Processes only	Yes
Installed Software	No	Yes
Log Monitoring		
Event Logs	Yes	Yes
Syslog	Yes	Yes
Text Logs	Yes	Yes
Extensible Collection		
PowerShell	No	Yes
JMX	No	Yes

For a detailed list of the system vital metrics you can monitor with the Skylar One agent, see *What System Vital Metrics Can the Skylar One Agent Collect?*

For a current list of supported operating systems and host system requirements for the Skylar One agent, see the System Requirements for the Agent page at the ScienceLogic Support Site.

NOTE: Users who are running version 102 or later of the "Microsoft: Windows Server" PowerPack can collect data via the Skylar One agent. For more information, see *Monitoring Windows Systems with PowerShell*.

What System Vital Metrics Can the Skylar One Agent Collect?

The following sections describe the system vitals that can be collected with Skylar One and with the Skylar One Agent, including definitions of each metric type and the collection methods that are and are not supported for each.

Metric Descriptions

The following table describes the system vital metrics that can be collected with Skylar One and the Skylar One Agent:

Metric	Type	Description
Availability	Performance	The ability to communicate with the managed entity or device.
File Systems	Configuration	The configuration of the file systems found within a managed entity that can include attributes like name, size, and type.
File Systems	Performance	Time series data associated with file system utilization that can include metrics like free space, size, and usage percentage.
Installed Software	Configuration	The software found on a managed entity that can include attributes like name, version, and installation date.
Network Interfaces	Configuration	The configuration of the network interface found within a managed entity that includes attributes like MAC address, IP address, position, and speed.
Network Interfaces	Performance	Time series data associated with physical memory utilization that includes metrics like inbound and outbound utilization, number of errors, and discard and usage percentage.
Physical Memory	Configuration	The configuration of the physical memory found within a managed entity that can include attributes like memory size.
Physical Memory	Performance	Time series data associated with physical memory utilization that can include metrics like memory used.
Ports	Configuration	The ports discovered on a managed entity.
Ports	Performance	Time series data associated with port availability.
Ports (Illicit)	Performance	An analysis of ports. When a port from the illicit port list is found on a managed system, the system will trigger an event indicating an illicit port has been found.
Processes	Configuration	The processes found on a managed entity that can include attributes like name, process ID (PID), and state.
Processes	Performance	Time series data associated with process performance that can include metrics like availability percentage.
Processor	Configuration	The configuration of the processor found within a managed entity that can include attributes like number of cores, processor model, processor speed, cache size, and CPU ID.
Processor	Performance	Time series data associated with processor utilization that can include metrics like CPU idle time, CPU wait time, and overall CPU time.
Restarts	Performance	An analysis of uptime. When uptime is less than 15 minutes, the system triggers an event indicating the system was restarted.
Services	Configuration	The services found on a managed entity that can include attributes like name and state.
Services	Performance	Time series data associated with service performance that can include metrics like availability percentage.
SSL Certificates	Configuration	The certificates found on a managed system.
SSL Certificates	Performance	An analysis of certificate expiration date. The system will trigger an event when certificates are nearing expiration.
Uptime	Performance	The timespan since the managed entity was last initialized.

Metric	Type	Description
Virtual Memory (Swap)	Configuration	The configuration of the virtual memory found within a managed entity.
Virtual Memory (Swap)	Performance	Time series data associated with virtual memory utilization.

Supported Data Collection Methods for Monitoring Windows

The following table describes which methods of data collection are supported when running Skylar One and the Skylar One Agent on monitored Windows systems:

Metric	Type	Agentless			Agent-Based	
		SNMP	WMI	PowerShell	Gen-01	Gen-03
Availability	Performance	Yes	Yes	Yes	Yes	Yes
File Systems	Configuration	Yes	Some	Yes	Some	Yes
File Systems	Performance	Yes	Some	Yes	Some	Yes
Installed Software	Configuration	Yes	No	Yes	No	Yes
Network Interfaces	Configuration	Yes	Some	Yes	Some	Yes
Network Interfaces	Performance	Yes	Some	Yes	Some	Yes
Physical Memory	Configuration	Yes	Yes	Yes	Yes	Yes
Physical Memory	Performance	Yes	Yes	Yes	Yes	Yes
Ports	Configuration	Yes	No	Yes	Yes	No
Ports	Performance	Yes	No	Yes	Yes	No
Ports (Illicit)	Performance	Yes	No	Yes	Yes	No
Processes	Configuration	Yes	Some	Yes	Yes	Yes
Processes	Performance	Yes	No	Yes	Yes	Yes
Processor	Configuration	Yes	Yes	Yes	Yes	Yes
Processor	Performance	Yes	Yes	Yes	Yes	Yes
Restarts	Performance	Yes	No	Yes	Yes	Yes
Services	Configuration	Yes	Some	Yes	No	Yes
Services	Performance	Yes	Some	Yes	No	Yes
SSL Certificates	Configuration	Yes	No	No	No	No
SSL Certificates	Performance	Yes	No	No	No	No
Uptime	Performance	Yes	No	Yes	Yes	Yes
Virtual Memory (Swap)	Configuration	Yes	Yes	Yes	Yes	Yes
Virtual Memory (Swap)	Performance	Yes	Yes	Yes	Yes	Yes

Supported Data Collection Methods for Monitoring Linux

The following table describes which methods of data collection are supported when running Skylar One and the Skylar One Agent on monitored Linux systems:

Metric	Type	Agentless		Agent-Based	
		SNMP	SSH	Gen 1	Gen 3
Availability	Performance	Yes	Yes	Yes	Yes
File Systems	Configuration	Yes	Yes	Some	Yes
File Systems	Performance	Yes	Yes	Some	Yes
Installed Software	Configuration	Yes	No	No	Yes
Network Interfaces	Configuration	Yes	Yes	Some	Yes
Network Interfaces	Performance	Yes	Yes	Some	Yes
Physical Memory	Configuration	Yes	Yes	Yes	Yes
Physical Memory	Performance	Yes	Yes	Yes	Yes
Ports	Configuration	Yes	Yes	Yes	No
Ports	Performance	Yes	Yes	Yes	No
Ports (Illicit)	Performance	Yes	Yes	Yes	No
Processes	Configuration	Yes	Yes	Yes	Yes
Processes	Performance	Yes	Yes	Yes	Yes
Processor	Configuration	Yes	Yes	Yes	Yes
Processor	Performance	Yes	Yes	Yes	Yes
Restarts	Performance	Yes	Yes	Yes	Yes
Services	Configuration	N/A	N/A	N/A	N/A
Services	Performance	N/A	N/A	N/A	N/A
SSL Certificates	Configuration	Yes	No	No	No
SSL Certificates	Performance	Yes	No	No	No
Uptime	Performance	Yes	Yes	Yes	Yes
Virtual Memory (Swap)	Configuration	Yes	Yes	Yes	Yes
Virtual Memory (Swap)	Performance	Yes	Yes	Yes	Yes

Extensible Collection

In addition to the capabilities listed above, you can use the Skylar One agent for "extensible collection", where you align the agent with Dynamic Applications to gather metrics and attributes from other infrastructures and applications.

The Skylar One Extended Architecture supports aligning PowerShell Dynamic Applications to devices monitored by the Skylar One Windows agent. The Skylar One Extended Architecture supports aligning JMX Dynamic Applications to devices monitored by the Skylar One Linux agent.

In addition, Dynamic Applications that leverage the Low Code No Code CLI/SSH framework can execute using the agent.

For more information, see *Configuring Extensible Collection*.

Agent PowerPacks

Skylar One includes two PowerPacks that can be used to collect agent-based system configuration and performance data: the "ScienceLogic: Agent" PowerPack and the "Host Agent" PowerPack.

Both PowerPacks are installed by default on your Skylar One system, and they include the following features:

- Dynamic Applications that collect configuration data and performance metrics from devices that are using agent-based collection
- Event Policies and alerts that are triggered when devices that are using agent-based collection meet certain status criteria

The "ScienceLogic: Agent" PowerPack collects agent-based data for devices on Skylar One systems running on the Skylar One Extended Architecture (Gen 3 agents). This PowerPack contains two Dynamic Applications:

- The "ScienceLogic Agent: System Configuration" Dynamic Application collects the following data:
    - CPU
    - CPU Information
    - CPUs
    - Hardware Totals
    - Memory
    - Speed (MHz)
    - Swap Capacity
- The "ScienceLogic Agent: System Performance" Dynamic Application collects the following data:
    - CPU Name
    - CPU Utilization
    - CPU Utilization Breakdown
    - Disk Average Queue Length
    - Disk IO Utilization
    - Disk Name
    - Memory Utilization
    - Network Read

    - Network Write
    - Sample Time
    - Swap Utilization

The "Host Agent" PowerPack, which collects agent-based data for devices on Skylar One systems running on a Distributed Architecture (Gen 1 agents). This PowerPack contains two Dynamic Applications:

- The "Host Agent: System Config" Dynamic Application collects the following data:
    - CPU
    - CPU Information
    - CPUs
    - Disk
    - Disk Information
    - Disk Space
    - Disks
    - Hardware Totals
    - Memory
    - Size
    - Speed (MHz)
- The "Host Agent: System Perf" Dynamic Application collects the following data:
    - CPU Name
    - CPU Utilization
    - CPU Utilization Breakdown
    - Disk Average Queue Length
    - Disk Name
    - Disk Utilization
    - Memory Utilization
    - Network Read
    - Network Write
    - Sample Time
    - Storage Available
    - Storage Name

- Storage Total
- Storage Utilization

> **NOTE:** Because the "ScienceLogic: Agent" PowerPack is required to collect data from devices that are using agent-based collection, Skylar One does not enable you to delete or modify this PowerPack.

Agent-Compatible PowerPacks

In addition to the "ScienceLogic: Agent" PowerPack and the "Host Agent" PowerPack, there are several other Agent-compatible PowerPacks that you can use to collect data from specific device types.

Windows Devices

> **NOTE:** For more information, see the section on *Executing the Skylar One Agent with Windows PowerShell*.

Java Management Extensions (JMX) Resources

You can also use the "JMX Base Pack" PowerPack to monitor JMX resources with the Skylar One agent.

For more information, see the section on *Executing the Skylar One Agent with JMX*.

Agent Architecture

The following sections describe how the Skylar One agent works in the Skylar One Distributed Architecture and in the Skylar One Extended Architecture.

Skylar One Distributed Architecture

In a Skylar One Distributed Architecture, the Skylar One Agent collects data from the device on which it is installed and transfers that data to a Message Collector in a Skylar One system using the HTTPS protocol. The Data Collector on which the Dynamic Applications and collection processes run then poll the Message Collector using the HTTPS protocol to transfer data to Skylar One.

TCP port 443 must be open between the Message Collector and the device on which an agent is installed.

In a Distributed Architecture, the Skylar One agent requires a standalone, dedicated Message Collector. The Message Collector does not need to be dedicated to agent usage, but the Message Collector cannot be a Data Collector that also performs message collection

NOTE: Message Collectors that process data from the agent have different system requirements than Message Collectors that do not process data from the agent. For more information about the system requirements when running agents in a Distributed Architecture, see the System Requirements page at the ScienceLogic Support Center.

Skylar One Extended Architecture

In the Skylar One Extended Architecture, a ***Skylar One agent*** collects data from the device on which it is installed and sends that data to a Load Balancer in front of a Compute Cluster. The Compute Cluster transforms the data and stores high-volume performance data in the Storage Cluster and other performance and configuration data in the Database Server.

If required, agents can use an HTTP proxy server as an intermediate step in sending data to Skylar One.

Using an agent in the Skylar One Extended Architecture provides more configuration and performance data than using an agent in a Distributed Architecture. This additional data includes system vitals, log data, and extensible collection.

NOTE: ScienceLogic highly recommends that your agents upload in 1-minute summarized uploads. You can verify your uploads from the **[Settings]** tab in the current Skylar One user interface.

NOTE: For more information about the system requirements when running agents in an Extended Architecture, see the System Requirements page at the ScienceLogic Support Center.

Chapter 2

Installing the Skylar One Agent

Overview

This chapter describes how to install, upgrade, and uninstall Skylar One agents for Windows, Linux, AIX, and Solaris operating systems. All agent installations, upgrades, and uninstallations require command-line interface access to the targeted device.

This chapter covers the following topics:

Caveats *19*

Installing an Agent from the Agents Page *19*

Authenticating the Agent *33*

Configuring a Skylar One Agent to Use a Proxy Server Connection *36*

Upgrading an Agent *43*

Stopping an Agent *46*

Uninstalling an Agent *47*

Installing an Agent from the Devices Page or the Device Manager Page *48*

Remotely Installing a Skylar One Agent *52*

Caveats

- If you use the ***Skylar One Extended Architecture*** (which includes Compute Nodes, Storage Nodes, and a Management node), install the agent from the **Agents** page (Devices > Agent). For details, see *Installing an Agent from the Agents Page*.

> **NOTE:** With on-premises Skylar One Extended Architecture systems, the TLS handshake can fail between the Windows Agent on a monitored device and the Skylar One streamer service. For details and the workaround, see the following Knowledge Base article: Extended Architecture.

- If you use the ***Skylar One Distributed Architecture*** (where the Skylar One Agent collects and transfers data to a Message Collector), you can install the agent from the **Devices** page or from the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface). For details, see *Installing an Agent from the Devices page or the Device Manager Page*.

> **NOTE:** If you are using PhoneHome Message Collectors in a Distributed Architecture, see *Configuring a Skylar One Agent to Work with PhoneHome Collectors* before you install the agent.

- RPM installer packages must be signed. Therefore, when installing an RPM package, you might receive a warning message if the RPM store does not contain ScienceLogic's public GPG key.

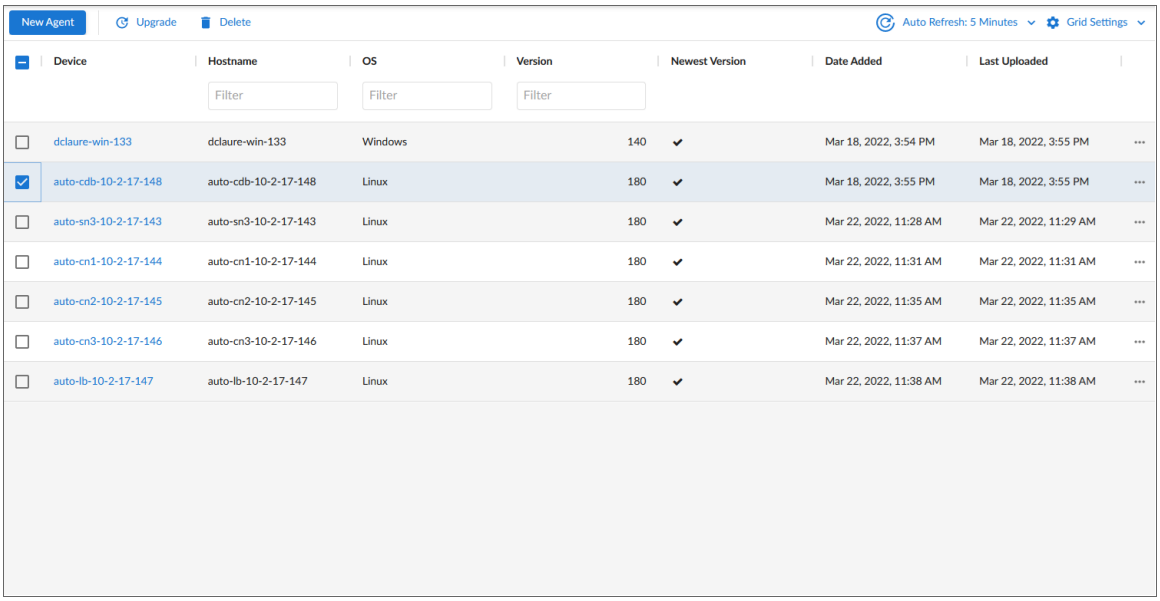
  To address or prevent this warning, you can obtain the ScienceLogic key and then add it to the RPM store:

  1. Go to http://keys.openpgp.org/search?q=devops%40sciencelogic.com.
  2. Download the key.
  3. Import the key into the RPM store using the following command:

```
rpm --import <file name>
```

Installing an Agent from the Agents Page

When running Skylar One on the Skylar One Extended Architecture, you can use the **Agents** page (Devices > Agents) to help you install the Skylar One agent on a Linux, Windows, AIX, or Solaris device.



You can also upgrade and delete agents from Skylar One on the **Agents** page.

TIP: You can sort the list by a column's values by clicking on its column heading. You can also filter the items on the page by typing or selecting filter options in one or more of the filters found above the columns. For more information, see the topic on *Filtering Inventory Pages*.

TIP: The **Agents** page appears as an option on the **Devices** menu *only* when you are using the Skylar One Extended Architecture. If your version of Skylar One does not have an **Agents** page, see *Installing an Agent from the Device Manager Page*.

NOTE: If you are using a Skylar One system with the Skylar One Extended Architecture, you ***must*** use the **Agents** page (Devices > Agents) to install agents, including AIX and Solaris agents. AIX and Solaris agents are not available for non-Skylar One Extended systems.

NOTE: If you want to remotely install one or more agents, see *Remotely Installing a Skylar One Agent*.

You do not install the agent from the **Agents** page itself. Instead, the **Agents** page enables you to gather the information or files you need to then install the agent on a particular device.

The instructions on the relevant tab of the **Agents** page also include command line parameters such as **URLFRONT**, which is used to define the Streamer service you are using as part of the Skylar One Extended Architecture.

If the agent installation is successful, one of the following automatically happens:

- If the primary IP address of the device is not currently monitored by Skylar One, then Skylar One creates a device record for the device and populates the device record with data provided by the agent. The device record is assigned a device class based on data reported by the agent.
- If the primary IP address of the device is currently monitored by Skylar One, the device record for the existing device is updated with data provided by the agent.

During initial discovery, the agent returns operating system type and version information to Skylar One. Based on this information, Skylar One assigns the corresponding device classes to a device monitored only by an agent. If a device is monitored by an agent and via SNMP, the device class assigned by SNMP discovery will take precedence.

Installing a Linux Agent

For a Linux system, the **Agents** page (Devices > Agents) provides version-specific commands that you will need to run on the Linux system you want to monitor. This page includes commands for the following operating systems:

- Ubuntu and Debian (64-bit)
- Red Hat and CentOS (64-bit)
- Red Hat and CentOS - OS Libs (64-bit)

NOTE: Linux agent version 196 is the minimum version required for Skylar One 12.5.1.

The Linux agent installation program checks for the relevant version of libcurl before installing the agent. If libcurl is not installed, the agent installation program installs the most recent version of libcurl.

Skylar One supports the use of proxy server connections when using the agent on Linux systems. For more information about installing and configuring a Linux agent with a proxy server, see *Configuring the Linux Agent to Use a Proxy*.

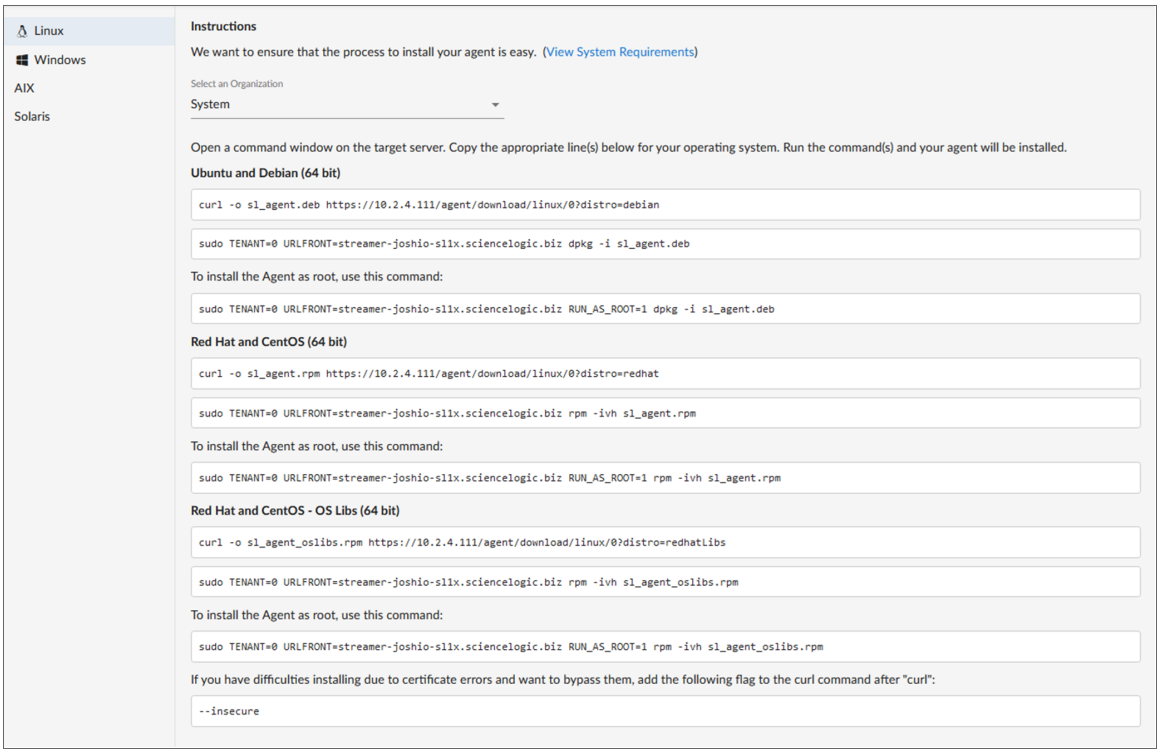
Beginning with Linux Agent version 184, by default the agent will install and run as a dedicated non-root user with the user name "scilog". To run the Linux agent as root, use the installation command which includes `RUN_AS_ROOT=1`. This configuration has the following limitations:

- Process data is only Non-Intercepted Process Data (NIPD). There are no intercepts or library data.
- Polled Data only works for the current user.
- Log files can only be monitored if they are readable by the `scilog user` or `scilog group`.
- You cannot upgrade the agent from the Skylar One user interface because the installation and uninstallation processes require root privileges.

If you are installing on SELinux, see *Using Agents with SELinux*.

To install a Linux agent:

1. On the **Agents** page (Devices > Agents), click **[New Agent]**. The **Agent Installation** page appears:



2. Click the **[Linux]** tab at top left.
3. From the ***Select an Organization*** drop-down list, select an organization for the device on which you are installing the agent. The **TENANT** value in the commands on this page is updated with the Skylar One Organization ID for the organization you selected.

> **TIP:** The **URLFRONT** value is the URL for the Streamer service you are using as part of the Skylar One Extended Architecture.

4. Based on the version of Linux running on the device you want to monitor, run the relevant commands from the **[Linux]** tab on the Linux device (the "target server"). After the commands complete, the Linux agent starts running in the background. The device on which you installed the agent appears on the **Agents** page (Devices > Agents), and the device is also added to the list of devices on the **Devices** page.

You can also install the Linux agent via the command line.

To install the Linux agent (Debian or Ubuntu) using the command line, execute the following command:

```
sudo TENANT=<organization id> URLFRONT=<streamer_URL> dpkg -i sl_agent.rpm
```

- `TENANT` and `URLFRONT` are required.
- The `RUN_AS` state can be set via command line with `RUN_AS_ROOT=<0|1>`.
- The default `RUN_AS` state is "scilog" when there is no command line `RUN_AS_ROOT=X` parameter given.

To install the Linux agent (RHEL or CentOS) using the command line, execute the following command:

```
sudo TENANT=<organization_id> URLFRONT=<streamer_URL> rpm -ivh sl_
agent.rpm
```

- `TENANT` and `URLFRONT` are required.
- The `RUN_AS` state can be set via command line with `RUN_AS_ROOT=<0|1>`.
- The default `RUN_AS` state is "scilog" when there is no command line `RUN_AS_ROOT=X` parameter given.

Using Agents with SELinux

The Linux agent supports running in SELinux for CentOS or Red Hat in both permissive and enforcing modes. The RPM installer for Linux will detect if SELinux is active (permissive/enforcing) upon installation. The agent automatically creates and installs an appropriate SELinux policy for use with the agent daemon.

With the Linux Agent, a shared library gets injected into other processes. For non-root processes, this shared library should not generate any errors or audit logs. However, for root processes, this shared library might generate some additional audit logs where some of the shared library functionality is restricted by SELinux. If this is the case, you can update the agent configuration to exclude pre-loading into these root processes.

Installing an AIX Agent

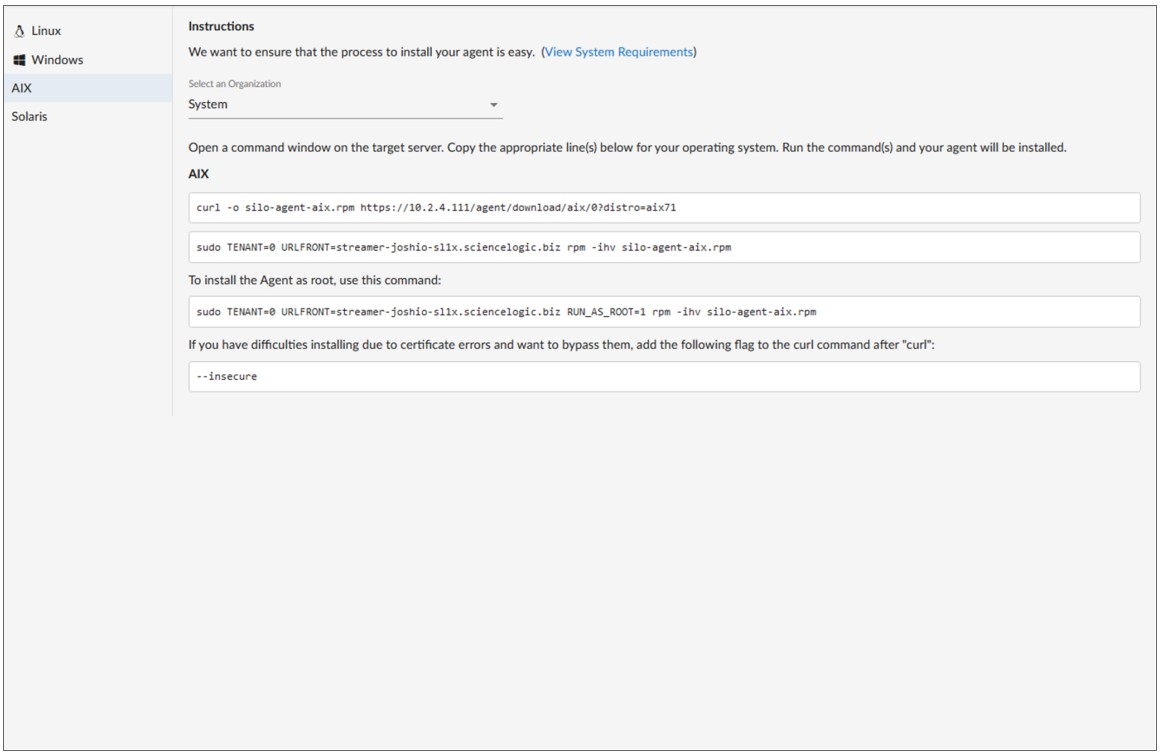
Beginning with AIX agent version 184, by default the agent will install and run as a dedicated non-root system user named "scilog". To run the AIX agent as root, use the installation command which includes `RUN_AS_ROOT=1`.

For an AIX system, the **Agents** page provides a set of commands that you will need to run on the AIX system you want to monitor.

> **NOTE:** AIX agent version 196 is the minimum version required for Skylar One 12.5.1.

To download and install an AIX agent:

1. On the **Agents** page (Devices > Agents), click **[New Agent]**. The **Agent Installation** page appears.
2. Click the **[AIX]** tab:



3. From the ***Select an Organization*** drop-down list, select an organization for the device on which you are installing the agent. The **Tenant** value in the commands on this page is updated with the Skylar One Organization ID for the organization you selected.

> **TIP:** The **URLFront** value is the URL for the Streamer service you are using as part of the Skylar One Extended Architecture.

4. Run the relevant commands from the **[AIX]** tab on the AIX device (the "target server"). After the commands complete, the AIX agent starts running in the background. The device on which you installed the agent appears on the **Agents** page (Devices > Agents), and the device is also added to the list of devices on the **Devices** page.

You can also install the AIX agent via the command line by executing the following command:

```
sudo TENANT=<organization_id> URLFRONT=<streamer_URL> rpm -ivh sl_agent_
aix.rpm
```

- `TENANT` and `URLFRONT` are required.
- The `RUN_AS` state can be set via command line with `RUN_AS_ROOT=<0|1>`.
- The default `RUN_AS` state is "scilog" when there is no command line `RUN_AS_ROOT=X` parameter given.

Installing a Solaris Agent

Beginning with Solaris agent version 184, by defaul the agent will install and run as a dedicated non-root system user named "scilog". To run the Solaris agent as root, add the file /etc/scilog/run_state.txt containing `RUN_AS_ROOT=1`.

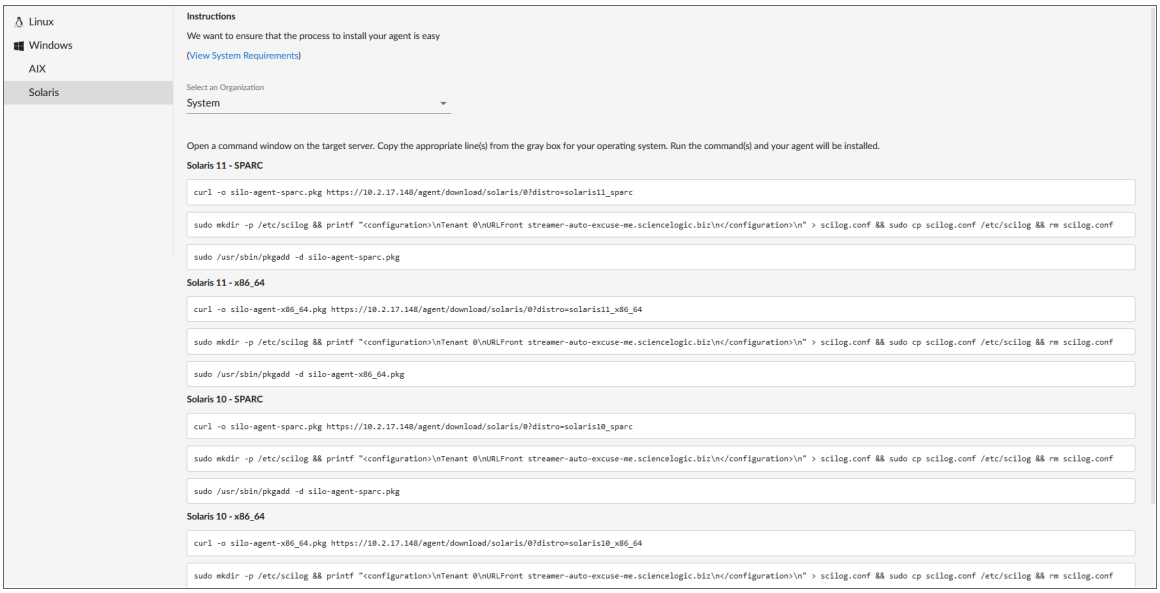
For a Solaris system, the **Agents** page provides version-specific commands that will need to run on the Solaris system you want to monitor. This page includes commands for the following versions:

- Solaris 11 - SPARC
- Solaris 11 - x86_64
- Solaris 10 - SPARC
- Solaris 10 - x86_64

NOTE: Solaris agent version 196 is the minimum version required for Skylar One 12.5.1.

To download and install a Solaris agent:

1. On the **Agents** page (Devices > Agents), click the **[New Agent]** button. The **Agent Installation** page appears.
2. Click the **[Solaris]** tab:



3. From the ***Select an Organization*** drop-down list, select an organization for the device on which you are installing the agent. The **Tenant** value in the commands on this page is updated with the Skylar One Organization ID for the organization you selected.

> **TIP:** The **URLFront** value is the URL for the Streamer service you are using as part of the Skylar One Extended Architecture.

4. Based on the version of Solaris running on the device you want to monitor, run the relevant commands from the **[Solaris]** tab on the Solaris device (the "target server"). After the commands complete, the Solaris agent starts running in the background. The device on which you installed the agent appears on the **Agents** page (Devices > Agents), and the device is also added to the list of devices on the **Devices** page.

You can also install the Solaris agent via the command line by executing the following commands:

```
mkdir -p /etc/scilog && printf "<configuration>\nTenant <organization_id>
\nURLFront <streamer_URL>\n<configuration>\n" > scilog.conf && sudo cp
scilog.conf /etc/scilog && rm scilog.conf
```

```
sudo pkgadd -d silo-agent.pkg
```

- The file **/etc/scilog/scilog.conf** with fields `TENANT` and `URLFRONT` are required. Note that "<configuration>" is a valid component of the file content and not meant to be substituted with any company or organizational value.
- The `RUN_AS` state can be set by adding file **/etc/scilog/run_state.txt** with `RUN_AS_ROOT=<0|1>`.
- The default `RUN_AS` state is "scilog" when a **/etc/scilog/run_state.txt** file doesn't exist.

Installing a Windows Agent

For a Windows system, the **Agents** page provides an executable file (.exe) that you will need to run on the Windows system you want to monitor.

ScienceLogic provides an executable (.exe) installer called SiloAgent-install.exe. You will need to install and run the SiloAgent-install.exe file on the Windows system you want to monitor.

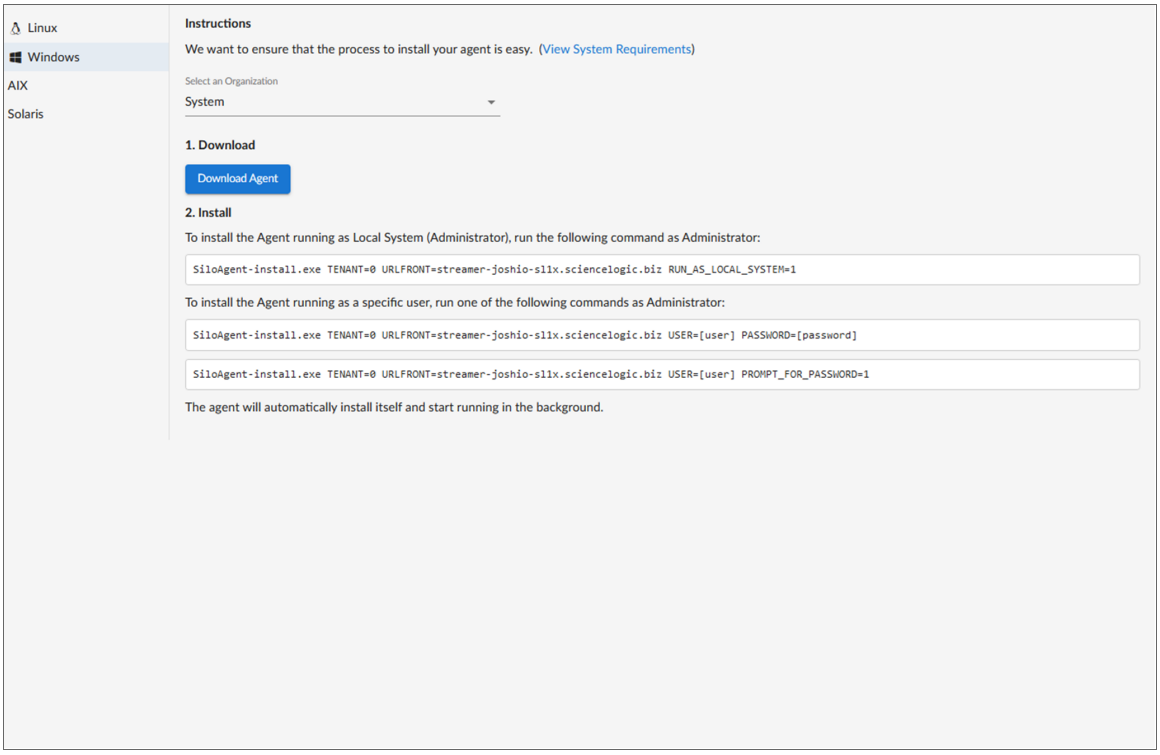
ScienceLogic also provides an MSI (.msi) installer upon request. The .exe installer is essentially a wrapper around the .msi file that allows for additional checks before installation, unwraps and runs the .msi file, prints information to the terminal, and provides information about failed installations. ScienceLogic recommends you use the .exe installer except for special cases. You can ask your ScienceLogic contact for an .msi file, if needed.

> **NOTE:** Skylar One supports the use of proxy server connections when using the Skylar One agent on Windows systems. For more information about installing and configuring a Windows agent with a proxy server, see *Configuring the Windows Agent to Use a Proxy*.

NOTE: Windows agent version 154 is the minimum required version for Skylar One 12.5.1

To download and install a Windows agent in Skylar One:

1. On the **Agents** page (Devices > Agents), click **[New Agent]**. The **Agent Installation** page appears.
2. Click the **Windows** tab at top left:



3. From the ***Select an Organization*** drop-down list, select an organization for the new agent. The **Tenant** value in the command on this page is updated with the Skylar One Organization ID for the organization you selected.
4. Click **[Download Agent]**. An **Opening SiloAgent-install.exe** dialog appears.
5. Save the **SiloAgent-install** .exe file for installing the agent.
6. Copy the **SiloAgent-install** file you downloaded to the Windows system on which you want to run the agent. To do so, you can either go to the console of the Windows system or use a utility like WinSCP.

7. To install the agent, use Windows PowerShell as an Administrator to run the command on the Windows system (the command is listed in step 2 on the **Windows** tab on the **Agents** page):

```
.\SiloAgent-install.exe tenant=<organization_id> urlfront=<streamer_URL> RUN_AS_LOCAL_SYSTEM=1
```

or

```
.\SiloAgent-install.exe tenant=<organization_id> urlfront=<streamer_URL> USER=<user> PASSWORD=<password>
```

where `<organization_id>` is Organization ID for the organization you selected in step 3, and `<streamer_URL>` is the URL for the Streamer service you are using as part of the Skylar One Extended Architecture.

For example:

```
.\SiloAgent-install.exe tenant=1 urlfront=streamer2020.int.sciencelogic.com RUN_AS_LOCAL_SYSTEM=1
```

> **NOTE:** RUN_AS_LOCAL_SYSTEM=1 indicates the agent service will run with administrator privileges. USER and PASSWORD designate a dedicated user to run the service. See *Configuring the Windows Agent* for more information.

8. When the agent installation completes, the PowerShell window displays "info: install complete", and the Windows agent starts running in the background. The device on which you installed the agent appears on the **Agents** page (Devices > Agents), and the device is also added to the list of devices on the **Devices** page.
9. To verify that the installation was successful, open the Windows Task Manager or enter the TASKLIST command to view running processes. The SiloAgent process will be running on the Windows machine.

> **NOTE:** When you upgrade to a newer version of the Windows agent, the installer upgrades the agent without generating a new CollectorID. The previous version's configuration file is passed to the upgraded version of the agent. The upgraded Windows agent continues to upload data to the same device record in Skylar One.

Configuring the Windows Agent Using the Command Line

The Windows agent can be configured with two command-line modes, `RUN_AS_LOCAL_SYSTEM` and `RUN_AS_USER`. You should choose which command line configuration your device needs. Prior to the Windows agent version 145, the agent was configured with user permissions equal to an administrator user when the agent ran under the Local System user account. For version 145 and later, you can specify any user account as the Windows agent runner. For security purposes, when a non-administrator user account is designated, the agent will have the same permissions and rights as that user account.

Configuring the Windows Agent to Run as Local System

Configuring the Windows agent service to run as a Local System with administrator privileges requires that you include `RUN_AS_LOCAL_SYSTEM=1` in your installation command. This configuration is simple, allows the agent to collect many types of data from your device, and guarantees that Dynamic Applications(e.g. PowerShell collections through the agent) execute without issue.

Configuring the Windows Agent to Run as a Dedicated User

Configuring Windows agent service to run as a dedicated user account requires that you specify `RUN_AS_USER=1` on the install or upgrade command to indicate that the agent should run as a specific user account. Using this mode allows you to have precise control over the agent permissions and an improved ability to audit agent actions.

Before installing the Windows agent:

1. Create a Windows user account (local user on the device or domain user) that will be used to run the SiloAgent Service.
2. Grant the user the exact permissions required to run the agent. Some permissions may be granted as part of the installation command but you may also configure the use before installation.

Using `RUN_AS_USER=1` requires additional command-line options that must be configured properly for the agent to collect data and execute actions.

1. Include `USER=<user>` in your installation command (required):
   - `USER`. Defines the user that will run the agent service when using `RUN_AS_USER=1`. `USER` may be a username, which indicates a local user account, or it may be in the doman\username form. `USER` is required for the agent to successfully install with `RUN_AS_USER=1`. It is not necessary to specify `USER` during an upgrade except to modify the property.
2. Include `PASSWORD=<password>` or `PROMPT_FOR_PASSWORD=1`. One or the other is required:
   - `PASSWORD`. Defines the password associated with `USER`. The password is required to set the `USER` as the agent's service runner and to add domain accounts to groups.`PASSWORD`, or `PROMPT_FOR_PASSWORD` is required if `RUN_AS_USER=1` is designated. It is not necessary to specify `PASSWORD` during an upgrade except to modify the property.
   - `PROMPT_FOR_PASSWORD`. This property is only available if using the .exe installer. Use `PROMPT_FOR_PASSWORD=1` on the command line instead of `PASSWORD` so that the .exe installer will prompt for a password before starting the installation. The value entered takes the place of `PASSWORD`. This option can be used to avoid entering plaintext passwords into the command line.

There are additional command line options in your installation command that grant permissions to the account specified by `USER`.You may want to grant permissions to the `USER` before the installation for the most control. There are a few permissions that are essential to the agent's central functions and you have the option to grant these during installation.

- LOG_ON_AS_A_SERVICE. User accounts require the "Log on as a service" user right to run services. `LOG_ON_AS_A_SERVICE=1` grants this permission to the user. The agent installation may succeed, but the agent service will fail to start if the `USER`does not have this right.

IMPORTANT: The "Log on as a service" user right is **required** to run the the SiloAgent service as a dedicated user. You must either use the LOG_ON_AS_A_SERVICE=1 option, or manually grant the user this right.

- DEBUG_PROGRAMS. To view the process details of processes owned by other users, a user account must have the required "Debug programs" user right. `DEBUG_PROGRAMS=1` grants this right to `USER`. Without this permission, only processes owned by `USER` can be monitored.
- PERFORMANCE_MONITOR_USERS. `PERFORMANCE_MONITOR_USERS=1` adds the `USER` to the Windows group "Performance Monitor Users" which enables the agent to read Windows Performance Counters such as CPU utilization, memory utilization, and disk utilization. This permission is required for the Microsoft Windows PowerShell dynamic applications to work correctly and allows the Windows Management Instrumentation (WMI) queries to execute properly.
- PERFORMANCE_LOG_USERS. `PERFORMANCE_LOG_USERS=1` adds the `USER` to the Windows group "Performance Log Users" which enables the agent to monitor application events through log monitoring and allows local WMI queries to execute properly.
- EVENT_LOG_READERS. `EVENT_LOG_READERS=1` adds the `USER` to the Windows group "Event Log Readers" and enables the agent to monitor application events through Log Monitoring.

NOTE: If neither mode is specified in a new install, the agent will default to `RUN_AS_USER=1`. If neither mode is specified during an upgrade, the agent will default to the mode already designated. Therefore, any agent running on version 144 and earlier will upgrade to `RUN_AS_LOCAL_SYSTEM`if not specified in the command line.

Example Install Commands

When installing the agent for the first time on a device running as Local System:

```
SiloAgent-install.exe TENANT=0 URLFRONT=10.1.1.2 RUN_AS_LOCAL_SYSTEM=1
```

1. **Tenant** and **URLFront** values must be specified.
   - `TENANT`. Corresponds to Organization ID in Skylar One. `TENANT` is required for agent installation. The Windows agent will continue to use the same **Tenant** value during an upgrade.
   - `URLFRONT`. The URL or IP for the streamer service. `URLFRONT` is required for Agent Installation. The Windows agent will continue to use the same **URLFront** value during an upgrade.
2. `RUN_AS_LOCAL_SYSTEM=1` must be specified.

When installing the agent for the first time on a device running as a dedicated user named "scilogrunner" (local account on the device) with a password "qwerty".

```
SiloAgent-install.exe TENANT=0 URLFRONT=10.1.1.2 RUN_AS_USER=1
USER=scilogrunner PASSWORD=qwerty
```

When installing the agent for the first time on a device running as a dedicated user named "MYDOMAIN\scilogrunner" (domain account) on "MYDOMAIN" with password "qwerty".

```
SiloAgent-install.exe TENANT=0 URLFRONT=10.1.1.2 RUN_AS_USER=1
USER=MYDOMAIN\scilogrunner PASSWORD=qwerty
```

When installing the agent for the first time on a device running as a dedicated local user named "scilogrunner" with the password "qwerty", and granting the suer the "Log on as a service" user right, the "Debug programs" user right and adding the user to the "Performance Monitor Users" group.

```
SiloAgent-install.exe TENANT=0 URLFRONT=10.1.1.2 RUN_AS_USER=1
USER=scilogrunner PASSWORD=qwerty LOG_ON_AS_A_SERVICE=1 DEBUG_PROGRAMS=1
PERFORMANCE_MONITOR_USERS=1
```

1. Grant the user the "Log on as a service" and "Debug programs" user rights.
2. Add to the Performance Monitor Users group.

Example Upgrade Commands

When you upgrade the agent installation:

To upgrade an existing agent installation currently running as Local System, you are not required to specify any properties. The properties configured in the previous version are stored and the new version of the agent will continue to run as Local System, using the same **Tenant** and **URLFront**.

```
SiloAgent-install.exe
```

To upgrade an existing agent installation currently running as a dedicated user, you are not required to specify any properties. The upgraded agent stores the preferences from the previous version. The new version of the agent will continue to run as `USER` with the same **Tenant** and **URLFront**.

```
SiloAgent-install.exe
```

To change from a current agent installation running as Local System to a dedicated user during an upgrade, you are not required to provide **Tenant** and **URLFront**, but the `RUN_AS_USER=1` and `USER` and `PASSWORD` must be specified. Once specified, the new version of agent will run as `USER`.

```
SiloAgent-install.exe RUN_AS_USER=1 USER=scilogrunner PASSWORD=qwerty
```

To change from a current agent installation running dedicated user to Local System during an upgrade, you are not required to provide **Tenant** and **URLFront**, but `RUN_AS_LOCAL_SYSTEM=1` must be specified.

```
SiloAgent-install.exe RUN_AS_LOCAL_SYSTEM=1
```

To change the `USER` and `PASSWORD` of an existing agent installation, use the same installer and specify the new `USER` and `PASSWORD.`

```
SiloAgent-install.exe USER=scilogrunner_new PASSWORD=different_password
```

Troubleshooting the Windows Installation Agent

If you experience issues during the installation process, review the example scenarios below.

> **TIP:** When an installation fails, a SiloAgent_install_log.txt file is generated in the current directory. If you need to contact ScienceLogic Support, please have that file when you contact them.

The agent installation fails when attempting to install with `RUN_AS_USER=1`

- Specify a user account that already exists on the computer or domain.
- Ensure that either `PASSWORD=[password]` or `PROMPT_FOR_PASSWORD=1` is specified.
- Ensure that the `USER` and `PASSWORD` provided are valid.
- Ensure that all property names are spelled correctly.

The agent installation succeeds, but the SiloAgent service does not start. If `USER` and `PASSWORD` are valid, but `USER` does not have the"Log on as a service" user right, the installation will succeed but will not start. In this situation, you have two choices:

- manually grant `USER` the "Log on as a service" user right and then start the service
- re-run the installation or upgrade command with `LOG_ON_AS_A_SERVICE=1`

> **CAUTION:** Please ensure that your `USER` account or containing groups do not have the "Deny log on as a service" user right. This right supersedes the "Log on as a service" right and will prevent the agent service from starting.

Additional Considerations

Beyond installation errors, there may be other service-related concerns you could experience.

- The **[Upgrade]** button present on the Skylar One agent page sends a simple upgrade command to the Windows agent. The upgraded agent will have the same run-as mode as the previous version. There is currently no way to change `USER` which runs the agent service from the current Skylar One user interface. The upgrade must be done from the command line.
- For upgrades from the Skylar One user interface to be successful, the `USER` must be in the administrator group because administrator access is required for the agent to upgrade itself (i.e. initiating installation of new software). Otherwise, the upgrade must be conducted by an administrator via the command line.
- Other `USER` permissions should be limited (e.g. "Deny log on locally" or "Deny Access to this computer from the network").
- Exercise caution when adding the `USER` to the administrator group as you are passing administrator credentials during the installation.

- The USER must have read access to any directories where log file monitoring will occur or the log file monitoring will fail.
- The USER must have any permissions required to execute PowerShell commands or commands passed by Dynamic Applications or those commands will fail.
- Additional permissions or configuration may be required for other Dynamic Applications which execute scripts through the agent, such as PowerShell Performance and PowerShell Configuration Dynamic Applications. For example, the "Microsoft: SQL Server" PowerPack applications will not operate correctly unless the `USER` is granted permissions to the SQL server instance and databases.

Authenticating the Agent

Beginning with Unix Agent version 189 and Windows Agent Version 148, you can configure the Agent to make authenticated requests and uploads to the Skylar One. Skylar One Extended version 12.2.0 can generate tokens for agents, as well as receive authenticated requests from agents with tokens.

Configuring the Streamer

The streamer helm charts provide the tuneable helm variable `acceptStaticApiKey`. If `acceptStaticApiKey` is false, the streamer requires the agent to present a valid token and will reject any connection attempts by an agent without a token. If `acceptStaticApiKey` is not false, the streamer will accept connections from the agents presenting a valid token as well as agent connections where no token has been configured on that device.

Installing an Agent with a Token

To install a new agent that makes authenticated requests, that agent must acquire a valid **guid** and **token**. There are two methods to acquire guids and tokens. In both methods, the Skylar One user credential used must have permission to configure agents.

3. Provide you Skylar One credentials as part of the installation command. The installer will make a request to the Skylar One and acquire a guid and token for the new agent.
4. While you are logged in to Skylar One, use a GQL mutation to generate a guid and token. Provide the guid and token in the agent installation command.

Upgrading an Existing Agent to an Agent that Supports Tokens

You must upgrade the existing Agent to a version that supports tokens so the agent is able to make authenticated requests. In order to do that, the agent must acquire a token. There are two methods to upgrade an agent and also acquire a token at the same time. In both methods, the Skylar One credential used must have permission to configure agents. Both methods will preserve the existing device record in Skylar One.

1. Provide your Skylar One credentials as part of the upgrade command. The installer will make a request to the Skylar One and acquire a token for the existing Agent.

2. While you are logged in to Skylar One, use a GQL mutation to generate a token for the existing Agent. Provide the current guid and the new token in the Agent installation command.

Installing or Upgrading without Tokens

All current installation and upgrade commands continue to work as before. If an agent is not provided with a token, it will continue to attempt requests and uploads as before. Streamers with the tuneable helm variable `acceptStaticApiKey` not set to false will be able to receive these uploads as before.

Windows Installation Commands (Example)

To authenticate the Windows agent, follow the example on GQL mutations and how to acquire `AGENT_GUID` and `AGENT_TOKEN` with GQL. For more information about installation, see *Installing a Windows Agent*.

New agent installation with **`AP2_USER`** and **`AP2_PASSWORD`**

```
SiloAgent-install.exe TENANT=0 URLFRONT=url RUN_AS_LOCAL_SYSTEM=1 AP2_USER=em7admin AP2_PASSWORD=em7admin
```

New agent installation with **`AGENT_GUID`** and **`AGENT_TOKEN`**

```
SiloAgent-install.exe TENANT=0 URLFRONT=url RUN_AS_LOCAL_SYSTEM=1 AGENT_GUID=xxx AGENT_TOKEN=yyy
```

Upgrading an agent without a token to Windows version 148 or later and acquiring a tokeN with **`AP2_USER`** and **`AP2_PASSWORD`**

```
SiloAgent-install.exe AP2_USER=em7admin AP2_PASSWORD=em7admin
```

Upgrade an agent without a token to Windows version 148 or later and acquiring a token with **`AGENT_GUID`** and **`AGENT_TOKEN`**

```
SiloAgent-install.exe AGENT_GUID=<existing CollectorID> AGENT_TOKEN=<acquired token>
```

Linux Installation Commands-Debian and Ubuntu (Example)

. To authenticate the Linux agent for Debian and Ubuntu, follow the example on GQL mutations and how to acquire `AGENT_GUID` and `AGENT_TOKEN` with GQL. For more information about installation, see *Installing a Linux Agent*.

New agent installation with **`AP2_USER`** and **`AP2_PASSWORD`**

```
sudo TENANT=0 URLFRONT=url RUN_AS_ROOT=1 AP2_USER=em7admin AP2_PASSWORD=em7admin dpkg -i sl_agent.rpm
```

New agent installation with **`AGENT_GUID`** and **`AGENT_TOKEN`**

```
sudo TENANT=0 URLFRONT=url RUN_AS_ROOT=1 AGENT_GUID=xxx AGENT_TOKEN=yyy
dpkg -i sl_agent.rpm
```

Upgrading an agent without a token to Linux version 189 or later and acquiring a tokeN with **AP2_USER** and **AP2_PASSWORD**

```
sudo AP2_USER=em7admin AP2_PASSWORD=em7admin dpkg -i sl_agent.rpm
```

Upgrade an agent without a token to Linux version 189 or later and acquiring a token with **AGENT_GUID** and **AGENT_TOKEN**

```
sudo AGENT_GUID=<existing CollectorID> AGENT_TOKEN=<acquired token> dpkg -
i sl_agent.rpm
```

Linux Installation Commands-RHEL, CentOs, and AIX (Example)

To authenticate the Linux agent for RHEL and CentOs, follow the example on GQL mutations and how to acquire `AGENT_GUID` and `AGENT_TOKEN` with GQL. For more information about Linux installation, see *Installing a Linux Agent*. For more information about AIX installation, see *Installing an AIX Agent*.

New agent installation with **AP2_USER** and **AP2_PASSWORD**

```
sudo TENANT=0 URLFRONT=url RUN_AS_ROOT=1 AP2_USER=em7admin AP2_
PASSWORD=em7admin rpm -ivh sl_agent.rpm
```

New agent installation with **AGENT_GUID** and **AGENT_TOKEN**

```
sudo TENANT=0 URLFRONT=url RUN_AS_ROOT=1 AGENT_GUID=xxx AGENT_TOKEN=yyy
rpm -ivh sl_agent.rpm
```

Upgrading an agent without a token to Linux version 189 or later and acquiring a tokeN with **AP2_USER** and **AP2_PASSWORD**

```
sudo AP2_USER=em7admin AP2_PASSWORD=em7admin rpm -U sl_agent.rpm
```

Upgrade an agent without a token to Linux version 189 or later and acquiring a token with **AGENT_GUID** and **AGENT_TOKEN**

```
sudo AGENT_GUID=<existing CollectorID> AGENT_TOKEN=<acquired token> rpm -U
sl_agent.rpm
```

Generating GUIDs and Tokens Using GQL

It is possible to generate Agent guids and tokens using GQL, rather than using your Skylar One credentials in the installation or upgrade command. GQL mutations produce guid and token values which can be used as the `AGENT_GUID` and `AGENT_TOKEN` properties.

To execute a GQL mutation, log in to Skylar One, go to the **GQL Browser** page (System > Tools > GQL Browser), and complete the following steps:

1. Generate a guid and token for a new Agent device. Substitute the appropriate organization ID `(TENANT)`.

```
mutation {
  createAgentTokenSecret(organization: 0) { #USE DESIRED TENANT HERE
    id
    agent
    jwt
  }
}
```

The mutation returns an `agent` field, which is used as `AGENT_GUID`, and a `jwt` field, which is used as `AGENT_TOKEN`

2. Generate a token for an existing agent device. An existing agent already as a GUID, also known as the `CollectorID`. The existing agent's CollectorID can be found in `/etc/scilog/scilog.conf` (Unix) or `C:\Program Files\ScienceLogic\SiloAgent\conf\scilog.conf` (Windows).

   In the **Query Variables** panel, enter:

```
{
  "guid": "<Current CollectorID>"
}
```

   For the command, enter:

```
mutation ($guid: ID!) { updateAgentTokenSecret(agent: $guid) { id agent jwt } }
```

The mutation returns a `jwt` field, which is used as `AGENT_TOKEN`. Note that the agent field is the provided `CollectorID`.

Configuring a Skylar One Agent to Use a Proxy Server Connection

The following section covers how you can configure a proxy server connection when using an agent on Linux or Windows systems.

Configuring a Linux Agent to Use a Proxy

In the **scilog_proxy.conf file**, you can specify up to four Linux proxy server configurations when installing the agent. If the connection through the primary proxy fails or the proxy is unavailable, the agent will attempt to make the connection through the other specified proxies.

> **NOTE:** In version 12.3.6 and earlier, you can configure only one proxy server when installing the agent.

To configure the Linux agent to use a proxy server connection:

1. Create a configuration file named **scilog_proxy.conf** and add the following proxy information to the new file:

```
Proxy-Server
```

```
Proxy-Port
```

```
Proxy-Username
```

```
Proxy-Password
```

```
Proxy-Server2
```

```
Proxy-Port2
```

```
Proxy-Username2
```

```
Proxy-Password2
```

```
Proxy-Server3
```

```
Proxy-Port3
```

```
Proxy-Username3
```

```
Proxy-Password3
```

```
Proxy-Server4
```

```
Proxy-Port4
```

```
Proxy-Username4
```

```
Proxy-Password4
```

```
Primary-Proxy-Check
```

> **NOTE:** The `Primary-Proxy-Check` value is in minutes. Any value less than "15" will run at 15 minutes, while any value over "15" will run at the interval you specify. If you do not set a value for `Primary-Proxy-Check`, then the agent will continue to use the primary proxy server until a connection fails; at that point, it will try the next proxy in the list to see if it is successful.

For example:

```
Proxy-Server 10.1.1.1
```

```
Proxy-Port 8080

Proxy-Username admin1

Proxy-Password passw0rd

Proxy-Server2 10.1.1.1

Proxy-Port2 8901

Proxy-Username2 admin1

Proxy-Password2 passw0rd

Proxy-Server3 10.1.1.1

Proxy-Port3 8902

Proxy-Username3 admin1

Proxy-Password3 passw0rd

Proxy-Server4 10.1.1.1

Proxy-Port4 8903

Proxy-Username4 admin1

Proxy-Password4 passw0rd

Primary-Proxy-Check 60
```

NOTE: The agent requires at least the proxy server and proxy port information. If no authentication is required, leave the username and password commented out with a "#". If values are present for the username and password, the agent will try to use them. The agent currently supports basic authentication.

2. Add the new file to the **/etc/scilog** directory on the Skylar One server. This directory is also where the **scilog.conf** file resides.

NOTE: If the **scilog_proxy.conf** file exists, the agent will try to connect to a proxy. If you do not need a proxy, remove or rename this file to prevent the agent from accessing it. Alternatively, you can leave the parameters blank in the **scilog_proxy.conf** file.

Installing and Configuring with the Proxy Settings

You can set the proxy parameters at install time by including them as parameters for the install command, just as you do with the **Tenant** and **URLFront** values. Use the package name for the package being installed.

For example:

deb

```
sudo TENANT=0 URLFRONT=10.1.1.2 proxyserver=10.2.1.1 proxyport=3128
proxyusername=1 proxypassword=1 dpkg -i sl_agent_v173.15.deb
```

rpm

```
sudo TENANT=0 URLFRONT=10.1.1.2 proxyserver=10.2.1.1 proxyport=3128
proxyusername=1 proxypassword=1 rpm -ivh sl_agent_v173.16.rpm
```

aix

```
sudo TENANT=0 URLFRONT=10.1.1.2 proxyserver=10.2.1.1 proxyport=3128
proxyusername=1 proxypassword=1 rpm -Uvh scilogd-0.173.17-
0.aix7.2.noarch.rpm
```

NOTE: If no proxy is used, you can remove the proxy parameters in the install command. Similarly, if a proxy is used without credentials, remove the `proxyusername` and `proxypassword` parameters in the install command, and only include the `proxyserver` and `proxyport` parameters .

NOTE: Also note that the **Tenant** and **URLFront** values can now be passed on the install command for AIX.

Configuring a Windows Agent to Use a Proxy

In the **scilog_proxy.conf** file, you can specify up to four Windows proxy server configurations when installing the agent. If the connection through the primary proxy fails or the proxy is unavailable, the agent will attempt to make the connection through the other specified proxies.

NOTE: In version 12.3.6 and earlier, you can configure only one proxy server when installing the agent.

To configure the Windows agent to use a proxy server connection:

1. Create a configuration file named **scilog_proxy.conf** and add the following proxy information to the new file:

```
Proxy-Server

Proxy-Port

Proxy-Username

Proxy-Password

Proxy-Server2

Proxy-Port2

Proxy-Username2

Proxy-Password2

Proxy-Server3

Proxy-Port3

Proxy-Username3

Proxy-Password3

Proxy-Server4

Proxy-Port4

Proxy-Username4

Proxy-Password4

Primary-Proxy-Check
```

> **NOTE:** The `Primary-Proxy-Check` value is in minutes. Any value less than "15" will run at 15 minutes, while any value over "15" will run at the interval you specify. If you do not set a value for `Primary-Proxy-Check`, then the agent will continue to use the primary proxy server until a connection fails; at that point, it will try the next proxy in the list to see if it is successful.

For example:

```
Proxy-Server 10.1.1.1
```

```
Proxy-Port 8900

Proxy-Username admin1

Proxy-Password passw0rd

Proxy-Server2 10.1.1.1

Proxy-Port2 8901

Proxy-Username2 admin1

Proxy-Password2 passw0rd

Proxy-Server3 10.1.1.1

Proxy-Port3 8902

Proxy-Username3 admin1

Proxy-Password3 passw0rd

Proxy-Server4 10.1.1.1

Proxy-Port4 8903

Proxy-Username4 admin1

Proxy-Password4 passw0rd

Primary-Proxy-Check 60
```

NOTE: The agent requires at least the proxy server and proxy port information. If no authentication is required, leave the username and password values blank. If values are present for the username and password, the agent will try to use them. The agent currently supports basic authentication. The agent will discover the type of supported authentication, so there is no need to specify it in the configuration file.

2. Add the new file to the **C:\Program Files\ScienceLogic\SiloAgent\conf** directory on the Skylar One server. This directory is also where the **scilog.conf** file resides.

> **NOTE:** If the **scilog_proxy.conf** file exists, the agent will try to connect to a proxy. If you do not need a proxy, remove or rename this file to prevent the agent from accessing it. Alternatively, you can leave the parameters blank in the **scilog_proxy.conf** file. If you use the .msi installer, you can input the parameters in the dialog box for the installer.

When configuring a proxy password with the above procedure, Skylar One encodes the proxy password in the **scilog_proxy.conf** file. The password displays as a seemingly random string of characters.

> **NOTE:** You can manually edit the **scilog_proxy.conf** file, you can manually enter the proxy password. The agent will still work with a proxy password in plain text. The agent will detect if the setup script encoded the password; otherwise, it assumes the password is in plain text.

Installing and Configuring an Agent with the Proxy Settings

You can set the proxy parameters at install time by including them as parameters for the install command, just as you do with the **Tenant** and **URLFront** values.

For example:

```
 .\SiloAgent-install.exe Tenant=0 URLFront=10.1.1.2 proxyserver=10.2.1.1
proxyport=3128 proxyusername=1 proxypassword=1
```

> **NOTE:** If no proxy is used, you can remove the proxy parameters in the install command. Similarly, if a proxy is used without credentials, remove the `proxyusername` and `proxypassword` parameters in the install command, and only include the `proxyserver` and `proxyport` parameters.

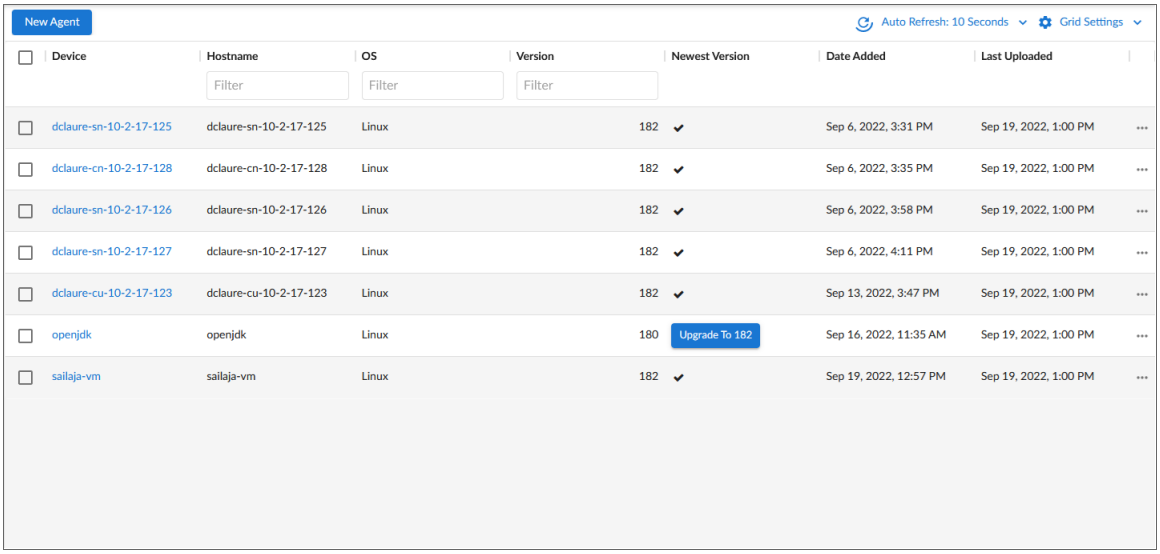
> **NOTE:** If no parameters are passed to **SiloAgent-install.exe**, including **Tenant** and **URLFront**, a pop-up dialog will appear where you can enter these values. If you do not require proxy credentials, you can leave those values blank.

Upgrading an Agent

When you have the latest version of the Skylar One agent, a check mark icon (✔) appears in the **Newest Version** column for that agent. If you do no have the latest version of the agent, an **[Upgrade]** button appears in the **Newest Version** column on the **Agents** page.

To upgrade to the latest version of an agent:

1. On the **Agents** page (Devices > Agents), locate the agent you want to upgrade:



2. Click the **[Upgrade]** button for that agent. The agent starts the upgrade process, which might take up to a minute to complete.
3. Click the **[Refresh]** button in your browser to ensure that a check mark icon (✔) appears in the **Newest Version** column for that agent.

> **NOTE:** It is not possible to configure the service `USER` when upgrading through the Skylar One user interface as the agent will continue using the service that `USER` already uses.

It is also possible to upgrade the agent from device command line.

To upgrade from the Windows command line:

1. Open the command line as an administrator and run the new .exe installer. It is not necessary to specify **Tenant** or **URLFront** values during an upgrade as those parameters carry over.
2. Change or configure any properties described in *Installing a Windows Agent* directly in the upgrade command line.

To upgrade with Linux :

Debian or Ubuntu

```
sudo dpkg -i sl_agent.deb
```

- The `RUN_AS` state can be changed via the command line with `RUN_AS_ROOT=1` or by updating the file /etc/scilog/run_state.txt with the same information.
  - The command line option will always take precedence.
  - The default `RUN_AS` state is "root" when no command line `RUN_AS_ROOT=X` parameter is given and there is no /etc/scilog/run_state.txt file.

RHEL or CentOS

```
sudo rpm -U sl_agent.rpm
```

- The `RUN_AS` state can be changed via the command line with `RUN_AS_ROOT=1` or by updating the file /etc/scilog/run_state.txt with the same information.
  - The command line option will always take precedence.
  - The default `RUN_AS` state is "root" when no command line `RUN_AS_ROOT=X` parameter is given and there is no /etc/scilog/run_state.txt file.

To upgrade with AIX:

```
sudo rpm -U sl_agent_aix.rpm
```

- The `RUN_AS` state can be changed via the command line with `RUN_AS_ROOT=1` or by updating the file /etc/scilog/run_state.txt with the same information.
  - The command line option will always take precedence.
  - The default `RUN_AS` state is "root" when no command line `RUN_AS_ROOT=X` parameter is given and there is no /etc/scilog/run_state.txt file.

To upgrade with Solaris:

```
sudo pkgadd -a /usr/share/overwrite.conf -d sl_agent.pkg scilogd
```

- The `RUN_AS` state can be set by adding the file /etc/scilog/run_state.txt with `RUN_AS_ROOT=<0|1>`.
- The default `RUN_AS` state is "root" when no /etc/scilog/run_state.txt file exists.

NOTE: Upgrades of the same version can be used to change the `RUN_AS` state.

Stopping an Agent

You can use the "delete" option on the **Agents** page to stop an agent from running. When you delete an agent, Skylar One stops the agent from running on the device, deletes the data gathered by that agent, and removes that agent from the **Agents** page.

> **NOTE:** Using the "delete" option for an agent ***does not*** actually remove the agent from the device. As a best practice, use the delete process to delete the data gathered by the agent (the uninstallation process does not delete this data), and then uninstall that agent, if needed. For uninstallation details, see *Uninstalling an Agent*. Also, ScienceLogic recommends that you delete the *device* that had the agent installed on it, or at least remove the agent-related Dynamic Applications, to prevent Skylar One from attempting to use the data the agent gathered from that device.

To delete an agent and its data:

1. On the **Agents** page (Devices > Agents), locate the agent you want to delete.
2. Click the **[Actions]** button () for that agent and select *Delete*. Skylar One stops the agent from collecting data.

> **NOTE:** You can also delete an agent from Skylar One using the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface). Click the checkbox for the device with the agent and select *Delete Selected Devices* from the ***Select Action*** menu at the bottom of the page.

You can also stop an agent through the command line.

To stop an agent on a Linux system (Debian, Ubuntu, RHEL, or CentOS), execute the following command:

```
sudo service scilogd stop
```

To stop an agent on an AIX system, execute the following command:

```
sudo /etc/rd.d/init.d/scilogd stop
```

To stop an agent on a Solaris system, execute the following command:

```
sudo svcadm disable scilogd
```

Uninstalling an Agent

When you uninstall an agent, you remove that agent completely from Skylar One, but you do not lose the data collected by that agent.

IMPORTANT: When uninstalling the agent, be sure that the `/var/log/jmx-collector` and the `/temp/scilog` directories are removed along with the `/var/log/scilogd.log` file.

Uninstalling a Linux Agent

To uninstall an agent on a Linux system:

1. Log in to the Linux system via the console or SSH as a user that has sudo administrator permissions.
2. For Red Hat, CentOS, and Oracle, execute the following command:

   ```
   sudo rpm -e scilogd
   ```

3. For Debian and Ubuntu, execute the following command:

   ```
   sudo dpkg -r scilogd
   ```

4. Optionally, you can remove the agent configuration directory from the Linux system. The configuration directory can be found at:

   ```
   /etc/scilog (rm -rf /etc/scilog)
   ```

Uninstalling an AIX Agent

To uninstall an agent on an AIX system:

1. Log in to the AIX system via the console or SSH as a user that has sudo administrator permissions.
2. To determine which package version is currently running, execute the following command:

   ```
   sudo rpm -ga|grep -i scilogd
   ```

   To determine which package version is running from a previous command, execute the following command:

   ```
   sudo rpm -e
   ```

3. Optionally, you can remove the agent configuration directory from the Linux system. The configuration directory can be found at:

   ```
   /etc/scilog (rm -rf /etc/scilog)
   ```

Uninstalling a Solaris Agent

To uninstall an agent on a Solaris system:

1. Log in to the Solaris system via the console or SSH as a user that has sudo administrator permissions.
2. To uninstall the agent, execute the following command:

   ```
   sudo pkgrm scilogd
   ```

3. Optionally, you can remove the agent configuration directory from the Linux system. The configuration directory can be found at:

   ```
   /etc/scilog (rm -rf /etc/scilog)
   ```

Uninstalling a Windows Agent

To uninstall an agent on a Windows system:

1. On the Windows system, open the **Control Panel**.
2. Go to the **Programs and Features** page (Control Panel > Programs > Uninstall a program).
3. Select the SiloAgent program from the list, and then click **[Uninstall]**.

Installing an Agent from the Devices Page or the Device Manager Page

If you are using a distributed Skylar One system *without* the Skylar One Extended Architecture (which includes Compute Nodes, Storage Nodes, and a Management Node), you ***must*** install the agent from the **Devices** page or from the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface).

TCP port 443 must be open between the Message Collector and the device on which an agent is installed.

> **NOTE:** If you are using a Skylar One system with the Skylar One Extended Architecture, you ***must*** use the **Agents** page (Devices > Agents) to install agents, including AIX and Solaris agents. AIX and Solaris agents are not available for non-Skylar One Extended systems.

> **NOTE:** If you are *not* using the Skylar One Extended Architecture, the **Agents** page (Devices > Agents) will not be available in the Skylar One user interface.

When you install an agent from the **Devices** page or from the **Device Manager** page, you will receive different content based on which type of agent you are installing:

- For a Linux system, the page displays commands that must be executed on that system.
- For a Windows system, the page provides an executable file to run on the Windows system.

To gather the necessary commands and executable files to install an agent on a device:

1. Do one of the following:
   - Go to **Devices** page, then click the down arrow next to the **[Add Devices]** button and select *Download/Install Agent*. The **Download/Install Agent** modal appears.
   - Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface), then click **[Actions]** and select *Download/Install Agent*. The **Agent Installation** modal appears.
2. On the modal, complete the following fields:
   - ***Select an OS***. Select the operating system running on the device on which you want to install the agent.

     > **NOTE:** If you require a FIPS-compliant version of the Skylar One agent, select *RedHat/CentOS 64-bit (OS Libs)*.

   - ***Select an Organization***. Select an organization. The list of organizations is dependent on your user account. If the agent discovers a new device, that device will be associated with the organization you select here.

     > **NOTE:** If you are installing an agent on a device that has already been discovered, you must select the organization that is already aligned with the existing device.

   - ***Select a Message Collector***. Select the Message Collector to which the agent will send its collected data.
3. If you selected a Linux operating system in the ***Select an OS*** field, the modal displays a list of commands to execute on the Linux system. Copy the commands for use during the *installation on the Linux device*.
4. If you selected a Windows operating system in the ***Select an OS*** field, the modal displays a ***Download Windows Agent*** link. Click the link and save the executable file for use during the *installation on the Windows device*.

> **TIP:** If you are installing an agent on multiple devices that run the same operating system, are part of the same organization, and connect to the same Message Collector, you can re-use the same commands or executable file on each of those devices.

Installing an Agent on a Linux System

To install an agent on a Linux system:

Copy and execute the appropriate commands from the Agent Installation page to download and install the agent. For more information, see *Installing a Linux agent* for details about install commands.

> **NOTE:** Skylar One supports the use of proxy server connections when using the Skylar One agent on Linux systems. To do so, open a command window on the target server and first configure curl to use a proxy server using the CURLOPT_PROXY option, and then to use a username and password combination for that proxy server using the CURLOPT_PROXYUSERPWD option.

Upgrading an Agent on a Linux System

To upgrade the agent on a Linux system:

When using the classic Skylar One user interface, agents must be upgraded from the device command line. For more information, see *Upgrading an Agent*.

Installing an Agent on a Windows System

To install an agent on a Windows system:

1. Copy the SiloAgent-install.exe file you downloaded from the **Agent Installation** page to the Windows system.
2. Execute the appropriate command copied from the **Agent Installation** page to install the agent. For more details about install commands see *Installing a Windows Agent*.

> **NOTE:** Skylar One supports the use of proxy server connections when using the Skylar One agent on Windows systems. In Windows 10, go to Settings > Network & Internet > Proxy to set up a proxy; in older versions of Windows, you can do this by clicking on the Control Panel and going to Internet Options > Connections > LAN Settings. If a proxy server requires authentication, you will need to provide a username and password before traffic is allowed through the proxy server and to the Internet.

Configuring a Skylar One Agent to Work with PhoneHome Collectors

If you are using PhoneHome Message Collectors in a Distributed Architecture, you will need to configure the settings for the Message Collector to prevent Skylar One from incorrectly configuring the Skylar One agent to use the loopback IP address when it should be using the local IP address for the Message Collector.

If you do not perform the configuration steps below, when you try to install the agent from a Message Collector when it is configured as a PhoneHome Collector, the URLs for both the download and the internal configuration of the agent reference the loopback address instead of the accessible public/private IP address of the Message Collector.

> **NOTE:** For more information about PhoneHome Collectors, see *Configuring Skylar One for PhoneHome Communication*.

To configure the Skylar One agent to work with a PhoneHome Collector:

1. Go to the **Appliance Manager** page (System > Settings > Appliances).
2. In the Message Collector row, click the Agent Endpoint Configuration icon ( ) for the agent. The **Agent Endpoint Configuration** modal appears.
3. Make sure that the configuration information under the ***Appliance Details*** field is accurate.
4. Complete the following fields:
   - ***Download Link Visibility***. Select whether you want to display the download link in the **Agent Installation** modal page.
     - If you select *Enabled*, the link to the Message Collector appears on the **Agent Installation** modal, which is available on the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface) when you click the **[Actions]** button and select *Download/Install Agent*.
     - If you select *Disabled*, the link to the Message Collector does not appear in the **Agent Installation** modal.
   - ***Endpoint Name for Download Link***. This field contains the name of the Message Collector. If the name is not populated by default, you can enter the name.
   - ***Hostname/IP for Agent Download***. Make sure that this field contains the IP address for the Message Collector.
   - ***Hostname/IP for Data Collection***. Enter the IP address for the Message Collector.
   - ***Default CUG Alignment***. Select the collector group aligned with the agent.
   - ***Upload Interval***. Specify how often the agent should upload data to the Message Collector. Your options include:
     - *20 seconds*. Upload a data snapshot every 20 seconds.
     - *1 minute*. Upload a data summary every 60 seconds.
5. Click ***Save***. The agent is now configured to work with a Message Collector when it is configured as a PhoneHome Collector.

Remotely Installing a Skylar One Agent

This section covers how to configure remote installations of the Linux agent and the Windows agent.

> **NOTE:** The following procedures are examples of how you could set up these installations, using Ansible as the tool for the installation. You might use Ansible in a different way, or you might use another tool altogether. These examples represent just one way to configure the remote installation of agents.

Remotely Installing a Linux Agent

This topic provides an example of how you can use Ansible to remotely install the Skylar One Linux agent. Please refer to the Ansible product documentation as needed.

To remotely install the Linux agent using Ansible:

1. Set up your Ansible control node, using the documentation at Installing Ansible.
2. Download the Skylar One Linux agent installers from the **Agents** page (Devices > Agents > New Agent button) in Skylar One. For example:
   - **silo-agent-x86_64.rpm**
   - **silo-agent-x86_64.deb**
3. Create the **hosts.yml** file, using the following code as an example:

```
servers:
  hosts:
    centos_host:
      ansible_host: 172.16.54.195
      ansible_user: root
    debian_host:
      ansible_host: 10.2.16.125
      ansible_user: root
```

4. Create the **playbook.yml** file, using the following code as an example:

```
---
- hosts: servers
  vars:
    tenant: 0
    url_front: 10.2.16.49
    package_rpm: silo-agent-x86_64.rpm
    package_deb: silo-agent-x86_64.deb
  tasks:

  - name: check if agent exists
    stat:
      path: /usr/bin/scilogd
    register: agent_binary

  - name: copy rpm installer
    copy:
      src: "{{ package_rpm }}"
      dest: ~/{{ package_rpm }}
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'RedHat')

  - name: copy deb installer
    copy:
      src: "{{ package_deb }}"
      dest: ~/{{ package_deb }}
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'Debian')

  - name: install rpm agent
    shell: TENANT={{ tenant }} URLFRONT={{ url_front }} rpm -ihv ~/{{
package_rpm }}
    args:
      creates: /usr/bin/scilogd
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'RedHat')

  - name: install deb agent
    shell: TENANT={{ tenant }} URLFRONT={{ url_front }} dpkg -i ~/{{
package_deb }}
```

```
    args:
      creates: /usr/bin/scilogd
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'Debian')

  - name: remove rpm installer
    file:
      path: ~/{{ package_rpm }}
      state: absent
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'RedHat')

  - name: remove deb installer
    file:
      path: ~/{{ package_deb }}
      state: absent
    when: (agent_binary.stat.exists == False and ansible_facts['os_
family'] == 'Debian')
```

5. Run the playbook by executing the following command:

```
ansible-playbook playbook.yml -i hosts.yml --ask-pass
```

> **NOTE:** This example uses the `--ask-pass` parameter so that you are prompted for a single password to use with all remote computers.

The following code is an example of the output:

```
$ ansible-playbook playbook.yml -i hosts.yml --ask-pass
SSH password:

PLAY [servers]
*********************************************************************
***********************

TASK [Gathering Facts]
*********************************************************************
***************

ok: [centos_host]
ok: [debian_host]

TASK [check if agent exists]
*********************************************************************
*********
ok: [centos_host]
ok: [debian_host]

TASK [copy rpm installer]
*********************************************************************
************
skipping: [debian_host]
changed: [centos_host]

TASK [copy deb installer]
*********************************************************************
************
skipping: [centos_host]
changed: [debian_host]
```

```
TASK [install rpm agent]
*********************************************************************
*************
skipping: [debian_host]
changed: [centos_host]

TASK [install deb agent]
*********************************************************************
*************
skipping: [centos_host]
changed: [debian_host]

TASK [remove rpm installer]
*********************************************************************
**********
skipping: [debian_host]
changed: [centos_host]

TASK [remove deb installer]
*********************************************************************
**********
skipping: [centos_host]
changed: [debian_host]

PLAY RECAP
*********************************************************************
***************************
centos_host                : ok=5    changed=3    unreachable=0
failed=0    skipped=3    rescued=0    ignored=0
debian_host                : ok=5    changed=3    unreachable=0
failed=0    skipped=3    rescued=0    ignored=0
```

Remotely Installing a Windows Agent

This topic provides examples of how you can use the PowerShell console or Ansible to remotely install the Skylar One Windows agent. Please refer to the PowerShell or Ansible product documentation as needed.

To remotely install the Windows agent using InstallSoftwareRemotely.ps1:

1. Download the Windows Agent installer from the **Agents** page (Devices > Agents > New Agent button) in Skylar One and add the installer to your domain-controller n its own directory. For example, **C:\files\SiloAgent-install.exe**.

2. Download the third-party PowerShell script to your domain-controller.
3. Create a comma-separated value (CSV) file with the names of the computers where you want to install the agent. For example: **C:\computers.csv**, such as:

```
rstlsw12t16ls01
```

```
rstlsw12t16ex01
```

```
computer3
```

```
computer99
```

4. Run the following command from a PowerShell console:

```
.\InstallSoftwareRemotely.ps1 -CSV C:\computers.csv -AppPath
'C:\foo\SiloAgent-install.exe' -AppArgs "Tenant=<organization_id>
urlFront=<streamer_URL> RUN_AS_LOCAL_SYSTEM=1"
```

where:

- `-CSV` uses the full path to the **.csv** file you created in step 3, above.
- `-AppPath` uses the full path to the installer in step 1, above. Make sure the file is in its own directory.
- RUN_AS_LOCAL_SYSTEM=1 is an example configuration. See *Configuring the Windows Agent* for more information.
- You can optionally use the `-Credential` option to be prompted for a different username and password to access all remote machines. Otherwise the script will use your current login account to access all remote machines.

The following code is an example of the output:

```
PS C:\> .\InstallSoftwareRemotely.ps1 -CSV C:\computers.csv -AppPath 'C:\-
foo\SiloAgent-install.exe'
-AppArgs "Tenant=0 URLFront=streamer-shared-e2e-test-
ing.int.sciencelogic.com"
Start remote installation on 2020-56-20 09:56:17
No credential specified. Using logon account
-----------------------------------------------------------------
Attempt 1 of 5
-----------------------------------------------------------------
COMPUTER RSTLSW12T16LS01 (1 of 2)
Coping C:\foo to \\rstlsw12t16ls01\C$\temp
Installing SiloAgent-install.exe on RSTLSW12T16LS01
Executing C:\temp\foo\SiloAgent-install.exe Tenant=0 URLFront=streamer-
shared-e2e-testing.int.sciencelogic.com
Deleting C:\temp\foo
```

```
SiloAgent-install.exe installed successfully.
COMPUTER RSTLSW12T16EX01 (2 of 2)
Coping C:\foo to \\rstlsw12t16ex01\C$\temp
Installing SiloAgent-install.exe on RSTLSW12T16EX01
Executing C:\temp\foo\SiloAgent-install.exe Tenant=0 URLFront=streamer-
shared-e2e-testing.int.sciencelogic.com
Deleting C:\temp\foo
SiloAgent-install.exe installed successfully.
-----------------------------------------------------------------
100% Success installing SiloAgent-install.exe on 2 of 2 computers:
rstlsw12t16ls01 rstlsw12t16ex01
```

To remotely install the Windows agent using Ansible:

1. Set up your Ansible control node. Refer to Ansible documentation for installing on other operating systems: Installing Ansible with pip.
2. Update your Ansible control node to manage Windows servers by running the following command:

   ```
   sudo pip3 install pywinrm
   ```

3. Ensure that remote machines are set up with PowerShell 3 or later:

   ```
   https://github.com/jborean93/ansible-
   windows/blob/master/scripts/Upgrade-PowerShell.ps1
   ```

4. Ensure that remote machines are setup for Windows Remote Management:

   ```
   https://github.com/ansible/ansible/blob/devel/examples/scripts/Config
   ureRemotingForAnsible.ps1
   ```

5. Download the Skylar One Windows agent installer, **SiloAgent-install.exe**, from the **Agents** page (Devices > Agents > New Agent button) in Skylar One.

6. Create the **hosts.yml** file, using the following code as an example:

```
group1:
  hosts:
    win2008r2:
      ansible_host: 10.2.16.41
      ansible_user: Administrator
    win2012:
      ansible_host: 10.2.16.42
      ansible_user: Administrator
    win2012r2:
      ansible_host: 10.2.16.40
      ansible_user: Administrator
group2:
  hosts:
    dc01:
      ansible_host: 10.2.16.60
      ansible_user: administrator@T16TESTDOMAIN
    ls01:
      ansible_host: 10.2.16.62
      ansible_user: administrator@T16TESTDOMAIN
    es01:
      ansible_user: administrator@T16TESTDOMAIN
      ansible_host: 10.2.16.64
```

6. Create the **my_playbook.yml** file, using the following code as an example:

```
---
- hosts: group1
  gather_facts: no
  tasks:
  - name: Ping Windows Hosts
    win_ping:
  - name: Copy agent installer
    win_copy:
      src: ./SiloAgent-install.exe
      dest: C:\Temp\SiloAgent-install.exe
  - name: Exec Installer
    win_command: C:\Temp\SiloAgent-install.exe Tenant=yyy
URLFront=zzz
  - name: Remove agent installer
    win_file:
      path: C:\Temp\SiloAgent-install.exe
      state: absent
```

7. Run the playbook by executing the following command:

```
ansible-playbook playbook.yml -i hosts.yml --ask-pass
```

> **NOTE:** This example uses the `--ask-pass` parameter so that you are prompted for a single password to use with all remote computers. Alternately, you could include the password in **hosts.yml** or **./group_vars/all.yml**.

The following code is an example of the output:

```
[em7admin@jkay88 demo_ansible]$ ansible-playbook my_playbook.yml -i host-
s.yml --ask-pass
SSH password:

PLAY [group1]
*********************************************************************_
******************************

TASK [Ping Windows Hosts]
*********************************************************************_
********************
ok: [win2012]
```

```
ok: [win2008r2]

TASK [Copy agent installer]
*********************************************************************_
******************
changed: [win2012]
changed: [win2012r2]
changed: [win2008r2]

TASK [Exec Installer]
*********************************************************************_
************************
changed: [win2012]
changed: [win2012r2]
changed: [win2008r2]

PLAY RECAP
*********************************************************************_
************************************
win2008r2                  : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
win2012                    : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
win2012r2                  : ok=3    changed=2    unreachable=0
failed=0    skipped=0    rescued=0    ignored=0
```

Chapter

3

Configuring the Skylar One Agent

Overview

This chapter covers how to configure the agent based on the Skylar One architecture you are using. This chapter also describes additional agent configurations, such as enabling extensible collection with a Skylar One agent.

This chapter covers the following topics:

System Settings that Affect Agents *63*

Configuring Agent Monitoring Based on Skylar One Architecture *63*

Configuring Agent Settings from the Device Investigator *65*

Configuring an Agent to Export Data to Skylar AI *68*

Configuring Extensible Collection *71*

Configuring Run Book Automations for a Skylar One Agent *73*

Configuring a Skylar One Agent on the Device Manager Page *76*

System Settings that Affect Agents

There are several system-level settings in Skylar One that define default behavior for all agents. To define these settings:

1. Go to the **Behavior Settings** page (System > Settings > Behavior).
2. On the **Behavior Settings** page, edit the values in one or more of the following fields:
   - ***Require TLS Validation for Gen 1 Agent***. When this checkbox is selected, Gen 1 agents that operate outside of the Extended Architecture will require TLS validation to upload data.

     > **NOTE:** To enable this validation, all Data Collectors and Message Collectors that will ingest agent data must have a valid and signed TLS certificate.

   - ***Gen3 Agent Unavailable Event Suppression Threshold***. Limits the number of per-agent "unavailable" events during large outages. When the number of unavailable Gen 3 agents rapidly exceeds this threshold, the system generates a single, aggregated "Extended Outage" event instead of one event per agent, and then clears the event once availability recovers below the threshold. This reduces event noise and improves triage during widespread agent outages.
   - ***Responder API Base URL***. This field lets you update the Responder API Base URL, which is required if you want to align PowerShell Dynamic Applications to the agent. In Skylar One version 11.2.0 and later, Skylar One completes this field for you.
3. Click **[Save]**.

Configuring Agent Monitoring Based on Skylar One Architecture

Based on your Skylar One architecture, you will have access to different workflows for configuring the monitoring settings on a Skylar One agent:

- If the Skylar One agent is used in a ***Skylar One Extended Architecture*** (which includes Compute Nodes, Storage Nodes, and a Management node), use the following configuration workflows:
  - To configure basic settings related to how the agent collects data, see *Configuring Agent Settings from the Device Investigator Page*.
  - To configure log file monitoring and Windows event log monitoring with the agent, see *Monitoring Device Logs Using an Agent*.
  - To configure and view data collected by the agent, see the relevant tabs on the **Device Investigator** page, including the following tabs:

        - Settings
        - Collections
        - Interfaces
        - Logs
        - Processes
        - Services

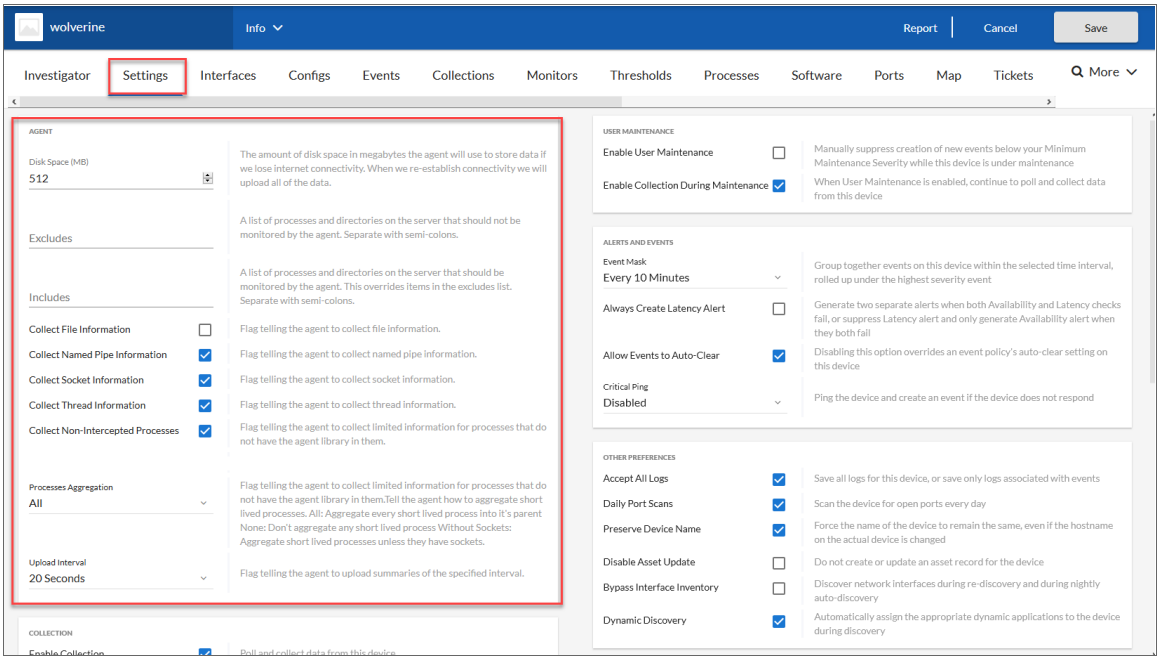
      For more information about these tabs, see *Using the Device Investigator*.

- If the Skylar One agent is used in a ***Skylar One Distributed Architecture*** (where the Skylar One Agent collects and transfers data to a Message Collector), use the following configuration workflows:
    - To configure basic settings related to how the agent collects data, see *Configuring a Skylar One Agent on the Device Manager Page*.
    - To configure log file monitoring and Windows event log monitoring with the agent, see *Monitoring Device Logs Using an Agent*.
    - To configure and view data collected by the agent, see the relevant tabs on the **Device Investigator** page, including the following tabs:
        - Settings
        - Collections
        - Interfaces
        - Logs
        - Processes
        - Services

      For more information about these tabs, see *Using the Device Investigator*.
    - To configure port monitoring with the agent, see the section on *Monitoring Ports*.

Configuring Agent Settings from the Device Investigator

If a device does *not* have an agent running on it, the **[Settings]** tab of the **Device Investigator** page for the device will display the **Collection** section at top left. However, if the device has an agent running on it, the **Agents** section appears above the **Collection** section:



In the **Agents** section, you can configure the disk space, excludes, includes, and other settings related to how you want the agent to collect data.

> **NOTE:** The **Agents** section appears on the **[Settings]** tab *only* when you are using the Skylar One Extended Architecture. If your version of Skylar One does *not* have an **Agents** section, see *Configuring a Skylar One Agent on the Device Manager Page*.

To configure agent settings:

1. From the **Devices** page, select the device with the agent that you want to configure. The **Device Investigator** page appears.

> TIP: To quickly find all devices on the **Devices** page that have agents installed on them, type "agent" in the ***Search*** field and select the "ANY: agent" search.

2. Click the **[Settings]** tab and click **[Edit.]**
3. In the **Agent** section, complete the following fields, as needed:

- ***Disk Space***. Specify the amount of disk space in MB that the agent can use to store data. If the agent loses connectivity to Skylar One, this disk space is used to store collected data until the connection to Skylar One is restored. When connectivity is re-established, the agent uploads all of its stored data.
- ***Excludes***. Type a list of processes and directories, separated by semicolons, that you do not want the agent to monitor.
- ***Includes***. Type a list of processes and directories, separated by semicolons, that you want the agent to monitor. This field ensures that specific processes are monitored.

> **NOTE:** If a process or directory is included in both the ***Excludes*** field and the ***Includes*** field, the item in the ***Includes*** field overrides the item in the ***Excludes*** field.

- ***Collect File Information***. Select this option if you want the agent to report the names of files accessed by each monitored process.
- ***Collect Named Pipe Information***. Select this option if you want the agent to collect named pipe information.
- ***Collect Socket Information***. Select this option if you want the agent to collect socket information.
- ***Collect Thread Information***. Select this option if you want the agent to collect thread information.
- ***Collect Non-Intercepted Processes***. Select this option if you want the agent to collect limited information for processes that do not contain the agent library.
- ***Processes Aggregation***. Specify how you want the agent to collect limited information for processes that do not have the agent library in them, and how to aggregate short-lived processes. Your options include the following:
    - *All*: Aggregate every short-lived process into its parent.
    - *None*: Do not aggregate any short-lived process.
    - *Without Sockets*: Aggregate short-lived processes unless those processes have sockets.
- ***Upload Interval***. Specify how often the agent should upload data. Your options include the following:

- *20 Seconds*. Upload a data snapshot every 20 seconds.
- *60 Seconds*. Upload a data summary every 60 seconds. This is the default setting. This option uses an improved data format that requires fewer Skylar One resources. The Skylar One agent continues to internally collect and poll data every 20 seconds, but the agent summarizes and uploads that data every 60 seconds. There is no data loss even though the data is uploaded less frequently.

> **NOTE:** If you specify 60 seconds for the upload interval, the summary upload now includes "watched" or "monitored" files, just like the snapshot upload does.

- ***300 seconds***. Upload a data summary every 300 seconds (five minutes).

4. Click **[Save]** to save your configuration settings for the agent.

> **TIP:** You can view and configure additional agent settings from the **Device Investigator** page for the device on which the agent is installed, including the following tabs: **[Interfaces]**, **[Processes]**, **[Services]**, and **[Collections]**. For more information about these tabs, see *Using the Device Investigator*.

Configuring an Agent to Export Data to Skylar AI

You can send agent data to Skylar AI so it can analyze the data for anomaly detection, predictive alerting, and new data visualizations.

For Gen 1 agents, which use the Skylar One Distributed Environment, you only need to configure the in the "Getting Started with Skylar Analytics" topic in the ***Skylar Analytics*** manual.

For Gen 3 agents, which use the Skylar One Extended Environment, you will need to complete the steps in the "Getting Started with Skylar Analytics" topic, and then complete the following additional procedures.

Enabling Skylar AI Export Using Deploy

After you complete the steps in the "Getting Started with Skylar Analytics" topic:

1. Retrieve the following values from the **/etc/sl-otelcol/sl-otelcol.conf** and **/etc/sl-otelcol/sl-otelcol-system-id.conf** files:
   - `OTELCOL_SYSTEM_ID`
   - `OTELCOL_SKYLAR_ENDPOINT`
   - `OTELCOL_SKYLAR_KEY`
   - `OTELCOL_SKYLAR_KEY_HEADER`
2. On the Management Node, use the prior values from the Database Server and add them to **sl1x.inv.yml**:

- `enable_skylar_integration: true`
- `skylar_otel_endpoint: "<OTELCOL_SKYLAR_ENDPOINT>"`
- `skylar_otel_key:"<OTELCOL_SKYLAR_KEY>"`
- `skylar_otel_key_header: "<OTELCOL_SKYLAR_KEY_HEADER>"`
- `skylar_system_id: "<OTELCOL_SYSTEM_ID>"`

3. Run the following command:

```
run docker-compose -f sl1x-deploy/docker-compose.external.yml run --
rm deploy app
```

As long as deploy is set to run for Skylar One 12.3.0 and later, the deploy will pull down the necessary services and configure them to enable the export of data to Skylar AI.

Manually Enabling Skylar AI

This procedure enables the export of data to the Skylar AI platform from Skylar One Extended by installing or upgrading helm charts from the command line.

On the Database Server, complete the following steps if you have not done them already:

1. Generate the Skylar AI config files:

```
sudo sl-otelcol-mgmt.py --skylar-api-key <API KEY --skylar-endpoint
<SKYLAR ENDPOINT> --skylar-all --simple-logging skylar
```

2. Ensure that the Skylar AI service connection is enabled on the Database Server from the Skylar One user interface by going to the **Service Connections** page (Manage > Service Connections) and clicking **[Add Service Connection]**.

3. Get the following values from **/etc/sl-otelcol/sl-otelcol.conf** file:

   - `OTELCOL_SYSTEM_ID`
   - `OTELCOL_SKYLAR_ENDPOINT`
   - `OTELCOL_SKYLAR_KEY`

4. On the Extended Management Node, use the values from **sl-otelcol.conf** when installing the helm charts that enable Skylar AI:

```
helm repo update sl1

helm upgrade --version 1.0.42 agent-vitals-service sl1/agent-vitals-
service
--set env.SKYLAR_EXPORT_ENABLED=true
helm upgrade --version 1.3.8 da-postprocessing-service sl1/da-
postprocessing-service
--set skylarExportEnabled=True -f output-files/da-postprocessing-
service-values.yml
helm upgrade --version 2.1.26 avail-store sl1/avail-store
--set skylarExportEnabled=true
-f output-files/avail-store-values.yml
helm upgrade --version 2.1.32 interface-store sl1/interface-store
--set skylarExportEnabled=true -f output-files/interface-store-
values.yml

# Might need to run with --install if not in place already.
helm upgrade --version 0.0.2 apl-optel-publisher sl1/apl-optel-
publisher
--set optelSettings.skylar.enabled=true
--set optelSettings.skylar.endpoint=<OTELCOL_SKYLAR_ENDPOINT>
--set optelSettings.skylar.key=<OTELCOL_SKYLAR_KEY>
"--set optelSettings.skylar.systemId=<OTELCOL_SYSTEM_ID>
```

NOTE: Line breaks were added to the lines of code, above, to allow the code sample to display properly.

After the helm charts are installed, data will begin exporting to the Skylar AI platform.

Validating the Skylar AI Export

Check the logs for any of the services:

- agent-vitals-service
- da-postprocessing-service
- avail-store
- interface-store

You should see logging similar to the following:

```
interface-store-8548989dcc-hjkzz 2024-09-17
15:30:00,861::INFO::1::interface_store.handlers.223:::Sending skylar
message for did: 5, sink: skylar.data
```

- apl-optel-publisher

```
apl-optel-publisher-67b7c6558b-shb2z 2024-09-17
15:31:21,643::INFO::1::apl_optel_publisher.handlers.109::: ** Published 1
payloads apl-optel-publisher-67b7c6558b-shb2z {'__OTLP_Resource__':
{'attributes': {'sl.service.name': 'Extended', 'sl.datatype.name':
'dynamic_app'}, 'schema_url': ''}}
```

If the helm charts were installed prior to 12.3.0, you might need to manually create the Kafka topic:

```
JMX_PORT=5557 /opt/bitnami/kafka/bin/kafka-topics.sh --bootstrap-server
kafka-service-headless:9092 --create --topic skylar.data --partitions 5 --
replication-factor 3
```

Configuring Extensible Collection

You can use the Skylar One agent for "extensible collection", where you align a Skylar One agent with Dynamic Applications to gather metrics and attributes from other infrastructures and applications.

Dynamic Applications are customizable policies, created for a specific vendor and a specific type of device or system, that tell Skylar One:

- What data to collect from devices
- How to present the data that has been collected
- When to generate alerts and events based on the data that has been collected

Skylar One includes Dynamic Applications for the most common hardware and software. You can customize these default Dynamic Applications to best work in your environment. You can also create custom Dynamic Applications.

You can align the agent with PowerShell Dynamic Applications (for Windows agents) and JMX Dynamic Applications (for Linux agents).

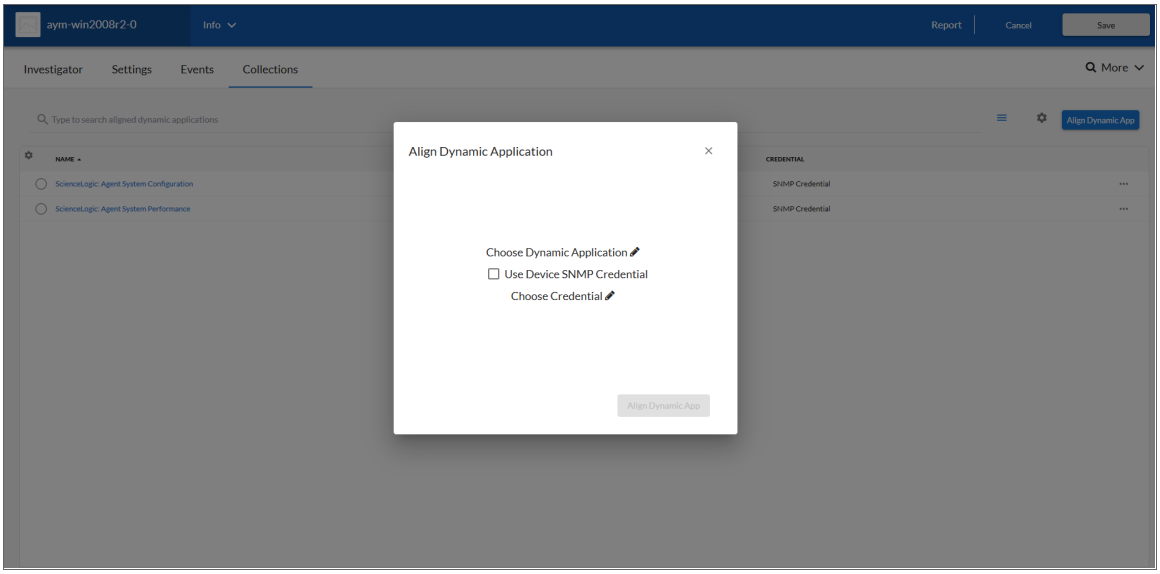
In addition, Dynamic Applications that leverage the Low Code No Code CLI/SSH framework can execute using the agent.

To enable extensible collection:

1. On the **Devices** page, search for the device with the agent installed on it.

> **TIP:** To quickly find all devices that have agents installed on them, type "agent" in the ***Search*** field on the **Devices** page, and select the "ANY: agent" search.

2. Select the device. The **Device Investigator** page appears.
3. Click the **[Collections]** tab, click **[Edit]**, and click **[Align Dynamic App]**. The **Align Dynamic Application** window appears:

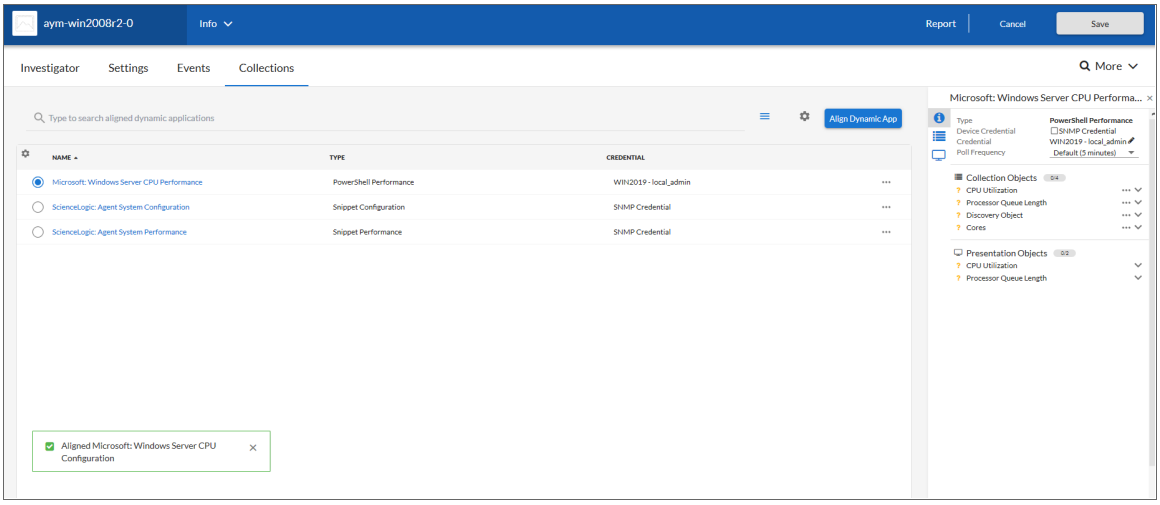


4. Click **Choose Dynamic Application** and search for the Dynamic Application you want to align.
5. Select the Dynamic Application and click **[Select]**. The **Align Dynamic Application** window appears again.
6. Click **Choose Credential** to search for a credential for the agent. The **Choose Credential** page appears.

> **NOTE:** The ***Account Type*** for the credential must be *Local* for the agent. Do not use the SNMP credential.

7. Select the credential and click **[Select]**. The **Align Dynamic Application** window appears again.

8. Click **[Align Dynamic App]**. The **[Collections]** tab appears, and a pop-up message displays when the Dynamic Application is aligned.



9. Click **[Save]**.
10. To view the metrics from that newly aligned Dynamic Application, click the **[Investigator]** tab.
11. Click the **[EDIT]** button in the right-hand pane of the **[Investigator]** tab.
12. In the **Add a metric** field, select a metric from the new Dynamic Application to view its data in the right-hand pane. The widget is added to the tab.

> **NOTE:** The metrics from the new Dynamic Application might take a minute or two to display in the list of metrics. Click **[Refresh]** in your browser if needed.

Configuring Run Book Automations for a Skylar One Agent

You can enable a Skylar One agent to execute run book automation by customizing the automation and action policies in the following PowerPacks:

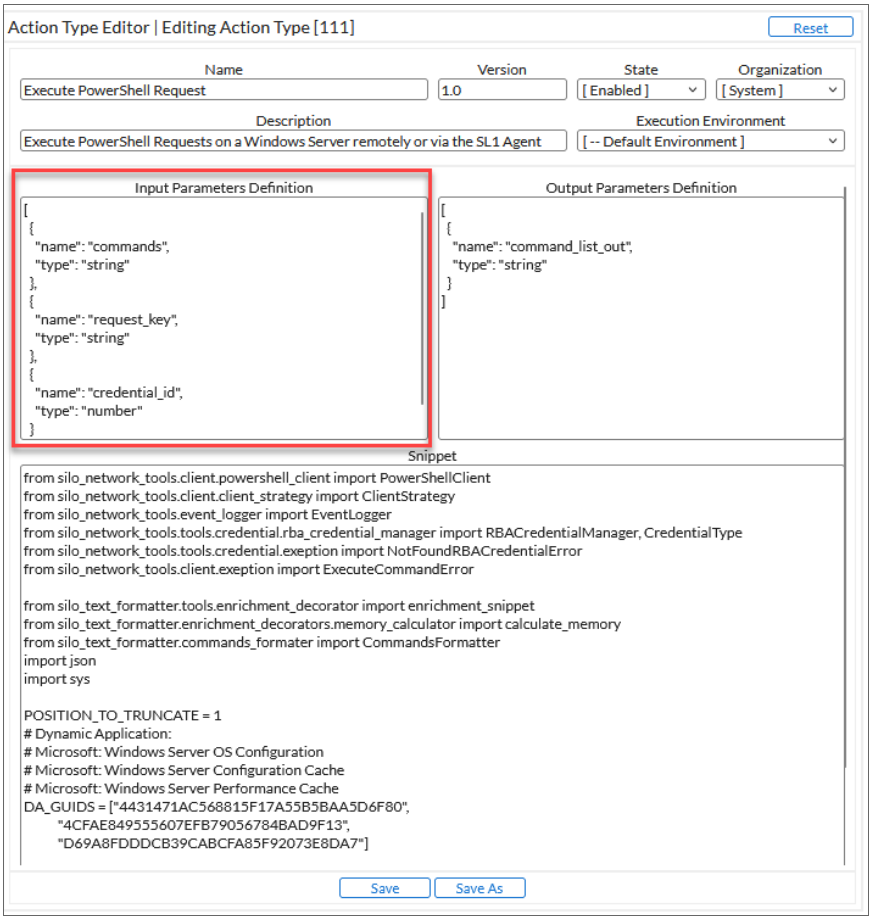
- "Linux SSH Automations" PowerPack version 103 or later
- "Windows PowerShell Automations" PowerPack version 103 or later

Action Type

For your agent-based run book action, you will need to use one of the following action types:

- "Execute PowerShell Request " from the "Windows PowerShell Automations PowerPack" version 103 or later
- "Execute Shell Commands" from the "Linux SSH Automations" PowerPack version 103 or later

These action types (Registry > Run Book > Action Types) contain the code in the ***Input Parameters Defintion*** pane that lets you need to run the actions on the agent:



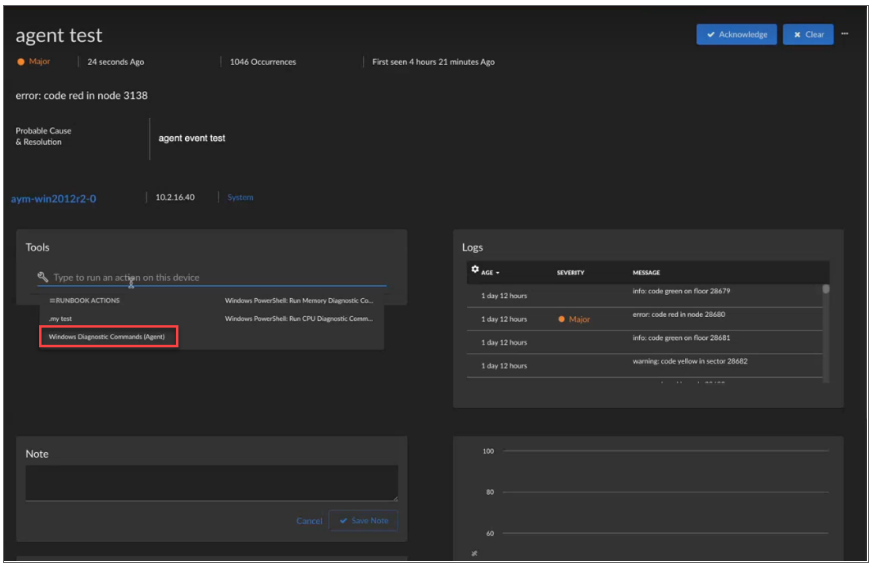
Configuring a Run Book Automation for an Agent Device

You configure a run book automation with a Skylar One agent in the same way you configure an automation on a non-agent device. For more information about creating run book action and automation policies, see the ***Run Book Automation*** manual.

The following steps provide an example of configuring a User-initiated Automation with a Skylar One agent:

1. Make sure you have the latest version of the "Linux SSH Automations" or "Windows PowerShell Automations" PowerPacks installed on your Skylar One system.
2. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
3. Click the **[Create]** button to create a new run book automation policy or click the edit icon (🔧) to edit an existing PowerShell or Linux SSH automation policy.
4. In the **Aligned Devices** field, make sure that the Skylar One agent device is selected.

5. In the ***Available Events*** pane, select the Skylar One event that will trigger this automation and click the right-arrow button.
6. In the **Available Actions** pane, select an action policy by highlighting it and clicking the right arrow-button.
7. Complete the other fields as needed, and then click **[Save]**.
8. Go to the **Action Policy Manager** page (Registry > Run Book > Actions).
9. Click the **[Create]** button to create a new run book action or click the edit icon ( ) to edit an existing PowerShell or Linux SSH action.
10. For the ***Action Type*** field, select *Execute PowerShell Request* or *Execute Shell Commands*.
11. Update the commands in the ***Input Parameters*** section as needed.
12. Click **[Save]** and close the **Policy Editor** window.
13. When an event occurs on the agent device, you can open the **Event Investigator** page for that event and run this user-initiated run book action by selecting the action from the **[Tools]** pane:



Configuring a Skylar One Agent on the Device Manager Page

> **NOTE:** To configure agent settings for an agent, you must first add the ***SL Agent*** column to the **Device Manager** page. For more information about adding the ***SL Agent*** column, see *Adding the SL Agent Column to the Device Manager Page*.

> **NOTE:** If you are using the Skylar One Extended Architecture, you can access these settings from the **[Settings]** tab of the **Device Investigator** page for the device on which the agent is installed. For more information, see *Configuring Agent Settings from the Device Investigator Page*.

You can control how an agent runs on a device by configuring the following agent settings on the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface):

- ***Disk Space***. Controls the amount of disk space that the agent can use to store data. If an agent loses connectivity to Skylar One, this disk space will be used to store collected data until the connection to Skylar One is restored.
- ***Data Directory***. Defines the directory in which the agent will store temporary data.
- ***Excludes***. Defines the list of processes and directories to explicitly exclude from monitoring by the agent.
- ***Includes***. Defines the list of processes and directories that must be explicitly monitored by the agent. Use the ***Includes*** field to ensure that specific processes are monitored.

> **NOTE:** If a process or directory is included in both the ***Excludes*** field and the ***Includes*** field, that process or directory will be monitored by the agent.

Adding the "SL Agent" Column to the Device Manager Page

The ***SL Agent*** column allows you to access the configuration settings for the agent on a device. For more information about agent configuration settings, see *Configuring Agent Settings on a Device*. By default, the ***SL Agent*** column is not displayed in the **Device Manager** page.

To add the ***SL Agent*** column to the **Device Manager** page:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface).
2. Click **[Actions]**, and then select *Device Manager Preferences*. The **Edit Device Manager Preferences** modal appears.

3. In the ***Device Manager Columns*** field, control-click *Agent*.
4. Click **[Save]**.

Configuring Agent Settings on a Device

To configure agent settings, you must first add the ***SL Agent*** column to the **Device Manager** page. For more information about adding the ***SL Agent*** column, see *Adding the SL Agent Column to the Device Manager Page*.

To configure agent settings on a device:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface).
2. Find the device for which you want to edit agent settings. In the ***SL Agent*** column, click the gear icon ( ) for the device. The **Agent Configuration** page appears.
3. Supply values in the following fields:
   - ***Disk Space***. Enter the amount of disk space that the agent can use to store data. If the agent loses connectivity to Skylar One, this disk space will be used to store collected data until the connection to Skylar One is restored.
   - ***Data Directory***. Enter the directory in which the agent will store temporary data.
   - ***Excludes***. Enter a semi-colon delimited list of processes and directories to explicitly exclude from monitoring by the agent.
   - ***Includes***. Enter a semi-colon delimited list of processes and directories that must be monitored by the agent. Use the ***Includes*** field to ensure that specific processes are monitored.

   > **NOTE:** If a process or directory is included in both the ***Excludes*** field and the ***Includes*** field, that process or directory will be monitored by the agent.

4. Click **[Save]**.

Chapter 4

Monitoring Logs Using the Skylar One Agent

Overview

This chapter describes how to use the Skylar One agent to monitor logs with Log File Monitoring policies. Skylar One supports multiple methods for ingesting log data, which you can use to generate events.

This chapter covers the following topics:

What is a Log File Monitoring Policy? *79*

Viewing the List of Log File Monitoring Policies *79*

Creating a Log File Monitoring Policy *80*

Aligning a Log File Monitoring Policy to Devices *81*

Editing a Log File Monitoring Policy *83*

Deleting Log File Monitoring Policies *83*

Viewing the List of Log File Monitoring Policies and Aligned Devices *84*

Creating an Event Policy for Agent Logs *85*

Creating an Event Policy for Agent Logs in the Classic User Interface *87*

What is a Log File Monitoring Policy?

A Log File Monitoring policy specifies:

- A file or Windows log on the host device that an agent will monitor
- The logs from the file or Windows log that an agent will send to Skylar One

You can create, edit, and delete Log File Monitoring policies from the **Log File Monitoring Policies** page (System > Manage > Log File Monitoring Policies). After creating a Log File Monitoring policy, you must align the policy to one or more devices either from the **Log File Monitoring** page or by using a Device Template.

You can view logs collected by the Skylar One agent on the **Logs** panel on the **Device Investigator** page for the device on which the agent is installed. The same logs also appear on the **[Logs]** tab in the **Device Properties** and **Device Summary** pages for that device. You can define event policies that specify how logs collected by an agent will trigger events.

Log File Monitoring policies can be included in a PowerPack. For information about including a Log File Monitoring Policy in a PowerPack, see the ***PowerPacks*** manual.

NOTE: When running the agent as a dedicated user on Windows or Linux systems, the user requires Read access to any directories being monitored. To monitor system events on Windows, the user must be in the Event Log Readers group. See *Installing a Windows Agent* for more information.

Viewing the List of Log File Monitoring Policies

The **Log File Monitoring Policies** page (System > Manage > Log File Monitoring Policies) displays a list of all Log File Monitoring policies. From this page, you can also create, edit, and delete Log File Monitoring policies.

TIP: You can sort the list by a column's values by clicking on its column heading. You can also filter the items on the page by typing or selecting filter options in one or more of the filters found above the columns. For more information, see the topic on *Filtering Inventory Pages*.

For each Log File Monitoring Policy, the page displays:

- ***Name***. Name of the Log File Monitoring policy.
- ***Policy ID***. Unique numeric ID, automatically assigned by Skylar One to each Log File Monitoring policy.
- ***Source Type***. The source of the logs on the monitored device. Possible values are:

    - *File*. The agent will monitor a file on the file system of the device(s).
    - *Event Log*. The agent will monitor the Windows log on the device(s).
- ***Source***. The full path of the log file or the name of the log that the agent will monitor.

> **TIP:** For Linux or Unix operating systems, use "/" in the file paths. For Windows you can use "\" in the file paths, but a double slash "\\" will escape the "\" to ensure the file path ends up with a legitimate slash in it.

- ***Filter***. The regular expression that the agent uses to determine whether a log message is sent to Skylar One.
- ***Subscribers***. The number of devices with which the policy is aligned.
- ***Edited By***. Skylar One user who created or last edited the Log File Monitoring policy.
- ***Last Edited***. Date and time the Log File Monitoring policy was created or last edited.

Creating a Log File Monitoring Policy

To create a Log File Monitoring policy:

1. Go to the **Log File Monitoring Policies** page (System > Manage > Log File Monitoring Policies).
2. Click **[Create]**. The **Log Monitoring Policy** modal appears.
3. Supply values in the following fields:
    - ***Name***. Enter a name for the policy.
    - ***Type***. Select the source of the logs on the monitored device. Choices are:
        - *File*. The agent will monitor a file on the file system of the device(s).
        - *Event Log*. The agent will monitor the Windows log on the device(s).
    - ***File Path***. If you selected *File* in the ***Type*** field, this field is displayed. Enter the full path of the file to monitor. You can use a * to match multiple files, such as **/var/log/em7/*.log**.
    - ***Source***. If you selected *Event Log* in the ***Type*** field, this field is displayed. Select the Windows log to monitor. Choices are:
        - *application*
        - *system*
        - *security*
        - *other*
    - ***Description***. If you selected *other* in the ***Source*** field, this field is displayed. Type the name of the event log source type.

- ***Limit***. The maximum log messages the agent sends to Skylar One per minute. If the number of matching logs exceeds this value, the agent will stop sending logs to the platform for the remainder of the minute. The limit resets at the beginning of the next minute.

  For example, suppose you set this field to *10,000*. Suppose the agent monitors a device that has 30,000 log messages. The agent will retrieve 10,000 logs and then wait until the beginning of the next minute. The agent will then retrieve the next 10,000 logs and then wait until the beginning of the next minute. The agent will continue to retrieve 10,000 logs per minute until it has retrieved all the logs from the device.

- ***Filter***. Specify a regular expression that will be used to evaluate the log messages in the specified file or Windows log. If a log message matches this regular expression, the agent will send that log message to Skylar One. If a log message does not match this regular expression, the agent will not send that log message to Skylar One.

> **NOTE:** For Windows event logs, the Skylar One Agent adds the **Event ID** to the value in the *Message* portion of the Windows log before applying the value in the ***Filter*** field. The agent does not apply the value in the ***Filter*** field to the *Instance ID* or any other property of a Windows event log entry.

> **TIP:** Avoid adding a leading ".*" in a filter, such as ".ERROR", as that character might increase the time it takes the agent to execute the filter, and on busy Skylar One systems, that character in the filter can negatively impact the CPU.

4. Click **[Save]**.
5. Before you can use this Log File Monitoring policy, you will need to align the policy with one or more devices. For more information, see *Aligning a Log File Monitoring Policy to Devices*.

Aligning a Log File Monitoring Policy to Devices

You can align Log File Monitoring policies to devices either from the **Log File Monitoring** page or by using a Device Template.

This section describes how to align a Log File Monitoring policy from the **Log File Monitoring** page. It also describes how to use a one-off Device Template to align a Log File Monitoring policy. For more information on Device Templates, including the other methods you can use to create, save, and apply Device Templates, see the ***Device Groups and Device Templates*** manual.

To align Log File Monitoring policies to one or more devices from the ***Log File Monitoring*** page:

1. Go to the **Log File Monitoring** page (Registry > Monitors > Logs).
2. Click **[Create]**. The Log File Monitor modal appears.
3. In the Log File Monitor modal, supply values in the following fields:

- ***Device***. Select a device to align with the Log File Monitoring policy.
- ***Log Policy***. Select the Log File Monitoring policy to align with the selected device. Only policies that are appropriate for the selected device will appear. For example, if you chose a Linux device in the ***Device*** field, the ***Log Policy*** field will not show policies of the *Event Log* type.

4. If desired, click the names of the following fields to enable and edit them. These fields allow you to override settings of the policy you selected in the ***Log Policy*** field *for the device selected in the* ***Device*** *field*:
   - ***File Path***. Enter the full file path or the file name to monitor. This field appears only if the type of the policy is *File*.
   - ***Limit***. The maximum log messages the agent sends to Skylar One per minute. If the number of matching logs exceeds this value, the agent will stop sending logs to the platform for the remainder of the minute. The limit resets at the beginning of the next minute. For example, suppose you set this field to *10,000*. Suppose the agent monitors a device that has 30,000 log messages. The agent will retrieve 10,000 logs and then wait until the beginning of the next minute. The agent will then retrieve the next 10,000 logs and then wait until the beginning of the next minute. The agent will continue to retrieve 10,000 logs per minute until it has retrieved all the logs from the device.
   - ***File***. Specify a regular expression that will be used to evaluate the log messages in the specified file or Windows log. If and only if a log message matches this regular expression, the agent will send the log message to Skylar One.
5. Click **[Save]**.

To align Log File Monitoring policies to one or more devices *using a Device Template*:

1. Go to the **Device Manager** page (Devices > Classic Devices, or Registry > Devices > Device Manager in the classic user interface).
2. Select the checkboxes for the devices with which you want to align Log File Monitoring policies.
3. In the ***Select Action*** drop-down list, select *MODIFY by Template*.
4. Click **[Go]**. The **Device Template Editor** modal appears.
5. Click the **[Logs]** tab.
6. Click the Add New Log Policy Sub-Template icon (+).
7. Supply values in the following fields:
   - ***Align Log Monitoring Policy With***. Select the devices to which the Log File Monitoring policy will be applied.
   - ***Log Monitoring Policy***. Select the Log File Monitoring policy you want to align with the selected devices.
8. Optionally, you can override one or more settings from the Log File Monitoring policy specifically for the selected devices. To do this, click the field label for each setting you want to override to enable the fields and supply a value in those fields. For a description of each field, see the *Creating a Log File Monitoring Policy* section.
9. Repeat steps 6 and 7 for each Log File Monitoring policy you want to align with the devices you selected in step 2.

10. If you want to save this Device Template for future use, select the ***Save When Applied & Confirmed*** checkbox and enter a name for the Device Template in the ***Template Name*** field.
11. Click **[Apply]**. The **Setting Confirmation** page is displayed.
12. Click **[Confirm]**. The aligned Log File Monitoring policy will appear on the **Registry > Monitors > Logs** page (Registry > Monitors > Logs).

Unaligning Log File Monitoring Policies from Devices

To delete Log File Monitoring Policies, you must first unalign the policy from any devices. You can unalign a Log File Monitoring policy by from the **Log File Monitoring** page.

To unalign devices from a Log File Monitoring policy:

1. Go to the **Log File Monitoring** page (Registry > Monitors > Logs)
2. Select the checkboxes for the devices from which the policy must be unaligned.
3. In the ***Select Action*** drop-down menu, choose *Delete Log File Monitors*.

> **NOTE:** This action does not delete the Log File Monitoring *policy*.

4. Click **[Go]** to unalign the Log File Monitoring policy from the devices.

Editing a Log File Monitoring Policy

To edit a Log File Monitoring policy:

1. Go to the **Log File Monitoring Policies** page (System > Manage > Log File Monitoring Policies).
2. Click the wrench icon (🔧) for the Log File Monitoring Policy you want to edit. The **Log Monitoring Policy** modal appears.
3. Edit the value in one or more fields. For a description of each field, see the *Creating a Log File Monitoring Policy* section.
4. Click **[Save]**.

Deleting Log File Monitoring Policies

> **NOTE:** Before you delete a Log File Monitoring Policy, you must unalign that policy from all devices. *See Unaligning Log File Monitoring Policies* for more information.

To delete one or more Log File Monitoring policies:

1. Go to the **Log File Monitoring Policies** page (System > Manage > Log File Monitoring Policies).
2. Select the checkboxes for the **Log File Monitoring Policies** you want to delete.

3. In the ***Select Action*** drop-down list, select *DELETE Log File Monitoring Policies*.
4. Click **[Go]**.

Viewing the List of Log File Monitoring Policies and Aligned Devices

The **Log File Monitoring** page (Registry > Monitors > Logs) displays a list of existing relationships between devices and Log File Monitoring policies. From the **Log File Monitoring** page, you can also align and unalign devices and Log File Monitoring policies.

For each aligned Log File Monitoring policy and device, the page displays:

- ***Name***. The name of the Log File Monitoring policy.
- ***Device Name***. The name of the device aligned to the Log File Monitoring policy.
- ***ID***. The unique numeric ID of the Log File Monitoring policy. The ID is automatically assigned by Skylar One.
- ***Source Type***. The source of the logs in the monitored device. The possible values are:
  - *File*. The agent monitors a file on the file system of the device. Usually, this is used to monitor Linux log files.
  - *Event Log*. The agent monitors to Windows log on the device.
- ***Source***. The full path of the log file or the name of the Windows log that the agent monitors.
- ***Filter***. The regular expression the agent uses to determine if a log should be sent to Skylar One.
- ***Limit***. The maximum log messages the agent sends to Skylar One per minute. If the number of matching logs exceeds this value, the agent will stop sending logs to the platform for the remainder of the minute. The limit resets at the beginning of the next minute. For example, suppose you set this field to *10,000*. Suppose the agent monitors a device that has 30,000 log messages. The agent will retrieve 10,000 logs and then wait until the beginning of the next minute. The agent will then retrieve the next 10,000 logs and then wait until the beginning of the next minute. The agent will continue to retrieve 10,000 logs per minute until it has retrieved all the logs from the device.
- ***Edited By***. The user who created or last edited the alignment between the device and Log File Monitoring policy.
- ***Last Edited***. The date and time the alignment between the device and Log File Monitoring policy was created or last edited.

TIP: You can sort the list by a column's values by clicking on its column heading. You can also filter the items on the page by typing or selecting filter options in one or more of the filters found above the columns. For more information, see the topic on *Filtering Inventory Pages*.

Creating an Event Policy for Agent Logs

To trigger events based on log messages collected by the agent, you must create an event policy on the **Event Policies** page, and then associate that event policy with a Log File Monitoring policy.

> **NOTE:** If you are using the classic user interface for Skylar One, see *Creating an Event Policy for Agent Logs in the Classic User Interface*.

To create an event policy that triggers based on log data collected by the agent:

1. Go to **Event Policies** page (Events > Event Policies).
2. On the **Event Policies** page, click the **[Create Event Policy]** button. The **Event Policy Editor** page appears, displaying the **[Basic]** tab.
3. On the **[Basic]** tab, define the following fields:
   - ***Event Policy Name***. Enter a name for the event policy
   - ***Enable Event Policy***. Select this checkbox to enable the event policy.
   - ***Event Source***. Select *ScienceLogic Agent* as the source for the event. When you do so, the fields below this field will change based on your selection.
   - ***Log Policy***. Select the Log File Monitoring Policy the agent will use to collect the log message.
   - ***Type of Match***. Use this dropdown to select *String* or *Regular Expression*.
   - ***Match String***. A string used to correlate the event with a log message. Can be up to 512 characters in length. To match this event policy, the text of a log message or alert must match the value you enter in this field. Can be any combination of alpha-numeric and multi-byte characters. Skylar One's expression matching is case-sensitive. This field is required for events generated with a source of Syslog, API, and Email.
   - ***Second Match String (Optional)***. A secondary string used to match against the originating log message. Can be up to 512 characters in length. Can be any combination of alpha-numeric and multi-byte characters. To match this event policy, the text of a log message or alert must match the value you enter in this field and the value you entered in the ***Match String*** field. This field is optional.
   - ***Event Auto Expiration***. If selected, enter the time in which an active event will be cleared automatically if there is no reoccurrence of the event in the fields that appear. You can enter time in minutes or hours.
   - ***Multiple Matches Required to Trigger Event***. If selected, enter the number of alerts and the time in which an event requires multiple triggers to occur in the fields that appear. You can enter time in minutes or hours.
   - ***Detection Weight***. If two event definitions are very similar, this field specifies the order in which Skylar One should match messages against the similar event definitions. The event definition with the lowest weight will be matched first. This field is most useful for events that use expression matching. Options range from 0 (first) - 20 (last).

- ***Multi Match***.By default, Skylar One will match a log message or alert to only one event policy. If a log message or alert matches multiple event polices, Skylar One will use the ***Detection Weight*** setting to determine which event policy the log message or alert will match. If you select the ***Multimatch*** checkbox in all event policies that can match the same log message or alert, Skylar One will generate an event for every event policy that matches that single log message or alert.
- ***Message Match***. If Skylar One has generated an event and then a second log message or alert matches the same event policy for the same entity, Skylar One will not generate a second event, but will increase the ***count*** value for the original event on the **Events** page and in the **Viewing Events** page. By default, this behavior occurs regardless of whether the two log messages or alerts contain the same message. If you select the ***Message Match*** checkbox, this behavior will occur only if the log messages or alerts contain the same message.
- ***Event Message***. The message that appears in the **Event Console** page or the **Viewing Events** page when this event occurs. This field defaults to "%M" for new event policies upon creation. The message can be any combination of alphanumeric and multi-byte characters. Variables include the characters "%" (percent) and "|" (bar). You can also use regular expressions and variables that represent text from the original log message to create the ***Event Message***:
  - To include regular expressions in the event message, surround the regular expression with %R and %/R. For example:

    %RFilename: .*? %/R

    This example would search for the first instance of the string "Filename: " (Filename-colon-space) followed by any number of any characters up to the line break. The %R indicates the beginning of a regular expression. The %/R indicates the end of a regular expression.

    Skylar One will use the regular expression to search the log message and use the matching text in the event message.

    For details on regular expression syntax, see the documentation at http://www.python.org.

  - You can also use the following variables in this field:
    - *%I* (capital "eye"). Depending on the context, this variable contains one of the following:
      - For events with a source of "dynamic", this variable contains the index value from SNMP; this index value will be displayed in the Event Message. For Dynamic Applications, %I maps to the raw index that comes back from SNMP. For example, a walk of the MIB at .1.3.6.1.4.1.999.3.2.1 might return the following OIDs, in which case *%I* would return .1.1, .2.1, and .3.1, respectively:

        ```
        1.3.6.1.4.1.999.3.2.1.1.1
        ```

        ```
        1.3.6.1.4.1.999.3.2.1.2.1,
        ```

        ```
        1.3.6.1.4.1.999.3.2.1.3.1.
        ```

- For events with a source of "syslog" or "trap", this variable contains the value that matches the ***Identifier Pattern*** field in the **[Advanced]** tab.
- For events with a source of "internal", this variable contains the "yName" (sub-entity name) value that matches the ***Identifier Pattern*** field in the **[Advanced]** tab.

- *%M*. The full text of the log message that triggered the event will be displayed in ***Event Message*** field.
- *%V*. Data Value from log file will be displayed in the ***Event Message*** field.
- *%T*. Threshold value from the log file will be displayed in ***Event Message*** field.

- ***Event Severity***. Defines the severity of the event. Choices are:
  - *Healthy*. Healthy events indicate that a device or condition has returned to a healthy state. Frequently, a healthy event is generated after a problem has been fixed.
  - *Notice*. Notice events indicate a condition that does not affect service but about which users should be aware.
  - *Minor*. Minor events indicate a condition that does not currently impair service, but the condition needs to be corrected before it becomes more severe.
  - *Major*. Major events indicate a condition that impacts service and requires immediate investigation.
  - *Critical*. Critical events indicate a condition that can seriously impair or curtail service and requires immediate attention (i.e., service or system outages).
- ***Use Interface Severity Modifier***. If selected, when the event is triggered, Skylar One will check to see if the interface associated with this event has a custom severity modifier. If so, the event will appear in the **Event Console** with that custom severity modifier applied to the severity in the ***Event Severity*** field. For example, if an interface with an ***Event Severity Adjust*** setting of *Sev -1* triggers an event with an ***Event Severity*** of *Major* and that event has the ***Use Interface Severity Modifier*** checkbox selected, the event will appear in the **Event Console** with a severity of *Minor*.

9. Optionally, supply values in the other fields on this page. For more information on the remaining fields, as well as the **[Advanced]** tab, see the *Defining and Editing Event Policies* chapter in the ***Events*** manual.
10. Click **[Save]**.

Creating an Event Policy for Agent Logs in the Classic User Interface

To trigger events based on log messages collected by the agent in the classic user interface for Skylar One, you must create an event policy that is associated with a Log File Monitoring policy.

To create an event policy in the classic user interface based on log data collected by the agent:

1. Go to **Event Policy Manager** page (Registry > Events > Event Manager in the classic user interface).
2. In the **Event Policy Manager** page, click **[Create]**. The **Event Policy Editor** page appears.
3. In the **Event Policy Editor** page and set of tabs, you can define a new event. The **Event Policy Editor** page contains three tabs:
   - ***Policy***. Define basic parameters for the event.
   - ***Advanced***. Define pattern-matching for the event and also define event roll-ups and suppressions.
   - ***Suppressions***. Suppress the event on selected devices. When you suppress an event, you are specifying that, in the future, if this event occurs again on a specific device, the event will not appear in the **Event Console** page or the **Viewing Events** page for the device.
4. Supply values in the following fields:
   - ***Event Source***. Select *ScienceLogic Agent*.
   - ***Policy Name***. The name of the event. Can be any combination of alphanumeric characters, up to 48 characters in length.
   - ***Operational State***. Specifies whether event is to be operational or not. Choices are *Enabled* or *Disabled*.
   - ***Event Message***. The message that appears in the **Event Console** page or the **Viewing Events** page when this event occurs. Can be any combination of alphanumeric characters.
     - You can use regular expressions that represent text from the original log message to create the ***Event Message***:
       - *%R*. Indicates a regular expression. Surround the regular expression with %R and %/R. For example, %RFilename: .*? %/R would search for the first instance of the string "Filename: " followed by any number of any characters up to the line break. For details on the regular expression syntax allowed by Skylar One, see http://www.python.org/doc/howto/.
       - *%I* ("eye"). This variable contains the value that matches the ***Identifier Pattern*** field in the **[Advanced]** tab.
       - *%M*. The full text of the log message that triggered the event will be displayed in ***Event Message*** field.
       - *%T*. Threshold value from the log file will be displayed in ***Event Message*** field.
   - ***Event Severity***. Defines the severity of the event. Choices are:
     - *Healthy*. Healthy Events indicate that a device or condition has returned to a healthy state. Frequently, a healthy event is generated after a problem has been fixed.
     - *Notice*. Notice Events indicate a condition that does not affect service but about which users should be aware.
     - *Minor*. Minor Events indicate a condition that does not currently impair service, but the condition needs to be corrected before it becomes more severe.
     - *Major*. Major Events indicate a condition that is service impacting and requires immediate investigation.

    - *Critical*. Critical Events indicate a condition that can seriously impair or curtail service and require immediate attention (i.e. service or system outages).
  - ***Use Modifier***. If selected, when the event is triggered, Skylar One will check to see if the interface associated with this event has a custom severity modifier. If so, the event will appear in the **Event Console** with that custom severity modifier applied to the severity in the ***Event Severity*** field. For example, if an interface with an ***Event Severity Adjust*** setting of *Sev -1* triggers an event with an ***Event Severity*** of *Major* and that event has the ***Use Modifier*** checkbox selected, the event will appear in the **Event Console** with a severity of *Minor*.
  - ***Policy Description***. Text that explains what the event means and what possible causes are.
5. Select the **[Advanced]** tab.
6. In the ***Log Policy*** field, select the Log File Monitoring policy that the agent will use to collect the log message.
7. Enter values in the following fields to specify specific text that must appear in the log message for the event policy to trigger:
  - ***First Match String***. A string used to match against the originating log message. To match this event policy, the text of a log message must match the value you enter in this field. Can be any combination of alphanumeric characters. Expression matching in Skylar One is case-sensitive.
  - ***Second Match String***. A secondary string used to match against the originating log message. To match this event policy, the text of a log message must match the value you enter in this field and the value you entered in the ***First Match String*** field. This field is optional.

> **NOTE:** The ***Match Logic*** field specifies whether Skylar One should process ***First Match String*** and ***Second Match String*** as simple text matches or as regular expressions.

8. Optionally, supply values in the other fields on this page. For more information on the remaining fields, as well as the **[Suppressions]** tab, see the ***Events*** manual.
9. Click **[Save]**.

Chapter 5

Skylar One Agent Troubleshooting

Overview

This chapter contains troubleshooting processes that you can use to address issues with a Skylar One agent.

TIP: On a Windows agent, you can run the following diagnostic command to generate a ***.tar.gz** file that contains useful information for troubleshooting issues: `...\SiloAgent.exe --diag`

TIP: On a Linux agent, use `--diag` for the diagnostic option.

To troubleshoot potential issues with Skylar One agents, perform the following procedures, in the following order:

Was the Agent Download Successful? 91

Is the Windows Installation or Upgrade Failing? 91

Is the Agent Process Running? 91

Is the Agent Configuration File Valid? 92

Has Skylar One Discovery Completed? 93

Is the Agent Able to Upload Data? 93

Is Skylar One Receiving Agent Data? 95

Is the Agent Not Reporting Vital Data and Metrics? 95
Can Skylar One Process Agent Data? 95
Is the Number of Processes Inconsistent with Other Applications? 96
Is the Agent Unable to Connect to the Streamer Endpoint? 96
Troubleshooting Examples 99
Additional Troubleshooting Situations and Best Practices 101
Agent Communication with Skylar One 102

Was the Agent Download Successful?

If the Windows agent download failed with a "500 Internal Server Error", restart the **uwsgi** service on the Skylar One system from which you are downloading the agent.

From an administrator command prompt in Windows, run the following command on the Skylar One system:

```
systemct restart uwsgi
```

Is the Windows Installation or Upgrade Failing?

See *Installing a Windows Agent* for more information.

Is the Agent Process Running?

> **NOTE:** When running the agent as a dedicated user on Windows, the "Log on as a service" user right is required for the service to start.

As a first step, always locate the following logs from the Message Collectors when troubleshooting:

- **/var/log/streamer_prime/streamer_prime.log**
- **/var/log/uwsgi/streamer_prime.log**

To determine if the agent process is running:

1. Check the Windows Task Manager or run the "tasklist" or "top" command.
2. Look for **SiloAgent.exe** (Windows) or **scilogd** (Linux):
    - Windows: If **SiloAgent.exe** is not running, check the "Application" event log for events with `source=SiloAgent`.

- Linux: If **scilogd** is not running, check **/var/log/messages** or **/var/log/syslog** for relevant log messages.

If you are using the Skylar One Extended Architecture, determine if the agent was deleted from the **[Agents]** tab instead of uninstalling the agent.

If the agent was deleted in the Skylar One user interface, Skylar One shuts down the agent instead of uninstalling the agent. You should re-install the agent that you deleted in Skylar One.

To re-install the agent that was shut down:

1. Uninstall the agent that you shut down.
2. Delete that agent's configuration from one of the following locations:
   - Windows: **C:\Program Files\ScienceLogic\SiloAgent\conf\scilog.conf**
   - Linux: **/etc/scilog/scilog.conf**
3. Install a new agent.

If the agent was *not* deleted, then the issue could be with the agent. You should generate diagnostics information to share with your ScienceLogic contact.

To generate diagnostics information for an agent:

1. From an administrator command prompt, run one of the following commands:
   - Windows: `C:\Program Files\ScienceLogic\SiloAgent\bin\SiloAgent.exe -diag`
   - Linux: `/usr/bin/scilogd --diag`
2. Share the contents of the newly created diagnostic file in the current directory with your ScienceLogic contact. Depending on your operating system, the file name is:
   - Windows: **scilog-***<current date>***.diag.tgz**
   - Linux: **sl-diag.tar.gz**

Is the Agent Configuration File Valid?

1. Check the agent configuration file in one of the following locations:
   - Windows: **C:\Program Files\ScienceLogic\SiloAgent\conf\scilog.conf**
   - Linux: **/etc/scilog/scilog.conf**
2. Check the configuration item **CollectorID**:
   - If there is *no* **CollectorID** tag, then the agent has not been able to reach the stream or message collector.
   - The **CollectorID** should be a GUID similar to "4179b06ef502129c3023a0f8d58f3c37".
3. Check the configuration item **URLfront**, which is where the agent attempts to get the configuration file:

- Determine if you can ping the **URLfront**.
- If you are using the Skylar One Distributed Architecture, **URLfront** should be the URL of the Message Collector.
- If you are using the Skylar One Extended Architecture, **URLfront** should be the URL of the Streamer service.
- If the URL for **URLfront** is not correct, then re-install the agent.
- If the URL for **URLfront** is correct, then determine if you can ping the host portion of **URLfront**. If you cannot ping, then there are customer firewall or NAT issues.

Has Skylar One Discovery Completed?

To check if Skylar One discovery finished:

1. Check the agent record in the Skylar One Distributed Architecture by using SSH to communicate with the Message Collector and running the following commands:

   ```
   redis-cli -p 6380
   ```

   ```
   keys *
   ```

   ```
   hgetall agent_<GUID>
   ```

   where `<GUID>` is the specific value from **scilog.conf**.

2. Look for a field "did". The following line is the EM7-device-id.
3. If the EM7-device-id is not present, then discovery has not completed.
4. Alternately, you can look on the **Devices** page for the device to compare the shown device-id to the value in the agent record.

Is the Agent Able to Upload Data?

Check the Agent Upload Directory

Check the upload directory for the agent for directories and files in one of the following locations:

- Windows: **C:\Program Files\ScienceLogic\SiloAgent\data**
- Linux: **opt/scilog/data**

If there are many items, then the agent is unable to upload.

If the number of items is *decreasing*, the agent might have an issue. The agent is slowly catching up, but this situation indicates that a previous issue existed.

If the number of items continues to *increase* overall, check the configuration item URL:

- The URL is the location where the agent attempts to upload files.
- Determine if the host portion of the URL is reachable. If the host portion is reachable, the name of the oldest item indicates the approximate time of the issue.

> **NOTE**: To prevent consuming the disk with backed-up data, the agent limits the size and count of items in the upload directory.

A procedural note regarding backed-up data:

For a new installation, the agent reaches out to the streamer for a configuration file. If the configuration file can't reach the streamer, the streamer goes into a slow poll mode, waiting for a good configuration file. In the meantime, the streamer does nothing else (it does not generate data or log files). As a result, even through it looks like there is no backup of data files, in reality, there are no data files.

After the Streamer service receives a valid configuration file:

- After a restart, the agent reaches out to the Streamer service for a new configuration file.
- If the agent can't reach the Streamer, the agent will still generate data files, because it has a valid configuration file from a previous run. In this situation, you will see data files backing up if the Streamer is unreachable.

> **NOTE**: You have the ability to set the `RequireWebCert` to true in the configuration file the streamer sends to the agent so the validation process is successful.

In summary, if you have a valid configuration, you will get data files. If you do not have a backup, Streamer can be reached.

Run the Agent in Debug Mode (Linux)

> **NOTE:** You might need to preface the following commands with `sudo` if you are in root-privileged mode.

1. Stop the agent daemon by running the following command:

   ```
   service scilogd stop
   ```

2. Start the agent from the command line:

   ```
   scilogd -d 2>&1 | tee /tmp/scilogd.log
   ```

3. Let the agent run for about five minutes.
4. Press **Ctrl+C** and examine the output file.

5. Restart the agent by running the following command:

```
service scilogd start
```

Is Skylar One Receiving Agent Data?

If you are using the Skylar One Distributed Architecture:

1. SSH into the Message Collector and run the following command:

```
sudo tail -n 100 /var/log/uwsgi/streamer_prime_uwsgi.log
```

2. Look for lines containing the hostname of the monitored device, such as the following:

```
10.2.16.40 - - [19/Apr/2018:17:04:55 +0000] "POST /SaveData.py/save_
data HTTP/1.1" 200 59 "-" "Windows SiloAgent : aym-win2012r2-0"
```

3. If there are no matching lines, then the Streamer service is not getting data from that agent.

If you are using the Skylar One Extended Architecture:

1. SSH into the Management Node and view the logs.
2. Look for lines containing the HOSTNAME of the monitored device.
3. If there are no matching lines, then the Streamer service is not getting data from that agent.

Is the Agent Not Reporting Vital Data and Metrics?

If the agent is not reporting Windows vitals or PowerShell Dynamic Applications are not being collected:

1. Ensure the agent is running as dedicated user.
2. Ensure the user is enrolled in the "Performance Monitor Users" group to be able to collect the desired metrics

Can Skylar One Process Agent Data?

Check the Message Collector log files or Skylar One Streamer log files:

1. If you are using the Skylar One Distributed Architecture, locate the following files from the Skylar One Message Collector and provide the files to your ScienceLogic contact:
   - **/var/log/uwsgi/streamer_prime_uwsgi.log**
   - **/var/log/streamer_prime/streamer_prime.log**

2. If you are using the Skylar One Extended Architecture, check all logs for *ERR* or *Except*:

```
for p in $(kubectl get pods | cut -f 1 -d ' '); do echo $p; kubectl
logs $p | grep -E "(ERR|Except)"; done
```

3. Search for "ERROR", "Exception", and "HARAKIRI".
4. Contact your ScienceLogic contact with any error messages you find in the log files.

> **NOTE:** If you do not find any error messages, then the issue is most likely with the Dynamic Application that runs on the Data Collector.

Is the Number of Processes Inconsistent with Other Applications?

- On Linux, many outputs from the `ps` command list the kernel threads (the processes listed in square brackets). Because the agent is not in the kernel, it will not list kernel threads.
- Be aware that the agent reports processes that are running as well as processes that started and may have stopped, while `top` or `ps` commands show processes that exist when they are executed.
- Check the agent configuration. Due to back-end space limitations, many configuration combinations can limit what data the agent sends. A combination of parameters to get all processes include the following:
  - **NIPD True**. The agent library can not get into all processes at times, often on install. Non-intercepted process discovery reports processes that are not intercepted via the library.
  - **SLPAggregation**. This parameter takes short-lived processes that exist for less than 80 seconds and rolls information about the processes into the information for their parents. As a result, the short-lived processes will not be seen.

Is the Agent Unable to Connect to the Streamer Endpoint?

Some Skylar One agents on Windows 2012 R2 might have issues connecting with the streamer endpoint if there is not a match with the default TLS ciphers.

To address this issue, add the ssl-ciphers configuration to the existing Kubernetes ConfigMap **nginx-configuration**.

To add ssl-ciphers in on-premises environments:

1. SSH to the Manager Node and enter to the Ansible shell.
2. Run the following commands to install vim:

```
sudo apt update
```

```
sudo apt install vim -y
```

3. Run the following command:

```
kubectl edit configmaps -n ingress-nginx nginx-nginx-ingress-
controller
```

4. In the **nginx-configuration** file, add the missing ciphers after `apiVersion`:

> **NOTE:** Line breaks were added to the following list of ciphers to make the code more readable.

```
data:
  ssl-ciphers: ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:
DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:
DHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256:
ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-
SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:
DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:AES128-GCM-SHA256:
AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:
DES-CBC3-SHA
  ssl-protocols: TLSv1 TLSv1.1 TLSv1.2 TLSv1.3
  use-proxy-protocol: "false"
```

5. Exit with `:wq`. In a few minutes, the agent will be automatically recognized. If the agent is not recognized, try restarting the agent.

To add ssl-ciphers in AWS environments:

1. SSH Bastion/JH and enter to the Ansible shell.
2. Run the following command:

```
kubectl edit configmaps -n ingress-nginx nginx-nginx-ingress-
controller
```

3. In the **nginx-configuration** file, add the missing ciphers after `apiVersion`:

```
data:
  ssl-ciphers: ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:
DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:
DHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256:
ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-
SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:
DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256:AES128-GCM-SHA256:
AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:
DES-CBC3-SHA
  ssl-protocols: TLSv1 TLSv1.1 TLSv1.2 TLSv1.3
  use-proxy-protocol: "true"
```

4. Exit with `:wq`. In a few minutes, the agent will be automatically recognized. If the agent is not recognized, try restarting the agent.

Validating Agent TLS Connections to the Skylar One Streamer Service

As of Skylar One 12.1.1, customers who use the Skylar One Gen 3 agent with on-premises Extended Architecture systems have the option to turn on TLS certificate validation when deploying the Streamer service. This provides additional security to confirm that the agent's connection to Skylar One is valid.

To enable this TLS validation, the extended cluster must be configured with a valid TLS certificate and the "requireTls" setting in the Streamer helm chart must be set to "true" when deploying the Streamer, such as in the following command:

```
helm upgrade --version 1.2.13 streamer sl1/sl1-streamer -f output-
files/steamer-values.yml --set requireTls=true
```

If you update this setting, the Streamer pods will restart and the agent will download the new configuration upon its next communication with the cluster.

CAUTION: This TLS validation is currently disabled by default for on-premises Extended Architecture deployments.

If you want to enable this feature, it is important to first ensure that the Streamer end point that is provided via the URLFRONT installation option is configured with a valid TLS certificate. If the agent is configured to validate the TLS connection but the cluster it is trying to communicate with does not have a valid TLS certificate, the agent will be unable to communicate with that cluster.

If this occurs, you can disable the validation by updating the Streamer deployment to disable the "requireTls" setting, updating the **scilog.conf** file to remove or alter the `RequireWebCert true` line, and then restarting the agent.

NOTE: This feature can be enabled on SaaS Skylar One deployments by submitting a Service Request case to the SRE queue at the ScienceLogic Support Center at https://support.sciencelogic.com/s/ , or by contacting your ScienceLogic customer service manager.

Troubleshooting Examples

Example /var/log/streamer_prime/streamer_prime.log for successful discovery

```
2019-01-04T17:07:42.355291+00:00 amateen-em7 journal: SCILO_
SP:6954|logger:log_info:132|INFO|Agent config request received with init
flag set to True. Generated Temp AID: 2ae22a6b4489457abb14373cd3816076.
Request: <WSGIRequest: GET '/api/collector/config/?collector_
key=aEf34$aq3TGSDdf&tenant_id=0&host_name=aym-win2012r2-
1&init=&os=windows&collector_id=0'>
```

```
2019-01-04T17:07:42.619082+00:00 amateen-em7 journal: SCILO_
SP:6954|logger:log_info:132|INFO|Calling Agent version with: <QueryDict:
{'collector_id': ['2ae22a6b4489457abb14373cd3816076'], 'type':
['windows_64'], 'tenant_id': ['0'], 'host_name': ['aym-win2012r2-1'],
'collector_key': ['aEf34$aq3TGSDdf'], 'version': ['115']}>
```

```
2019-01-04T17:07:43.028457+00:00 amateen-em7 journal: SCILO_
SP:16717|logger:log_warning:127|WARNING|System file received from aym-
win2012r2-1
```

```
2019-01-04T17:07:43.032897+00:00 amateen-em7 journal: SCILO_
SP:16717|logger:log_info:132|INFO|Making discovery call for agent
2ae22a6b4489457abb14373cd3816076
```

```
2019-01-04T17:07:43.746284+00:00 amateen-em7 journal: SCILO_
SP:30843|logger:log_warning:127|WARNING|System file received from aym-
win2012r2-1
```

```
2019-01-04T17:07:43.750553+00:00 amateen-em7 journal: SCILO_
SP:30843|logger:log_warning:127|WARNING|Discovery call within time
threshold, sleeping.
```

```
2019-01-04T17:07:46.676827+00:00 amateen-em7 journal: SCILO_
SP:16717|logger:log_info:132|INFO|Update agent request did: 4, oid: 0,
ip: 10.7.6.119, agent id: 2ae22a6b4489457abb14373cd3816076
```

```
2019-01-04T17:07:46.677114+00:00 amateen-em7 journal: SCILO_
SP:16717|logger:log_info:132|INFO|Discovery complete, getting new agent
device id. Downloading new config for device:
2ae22a6b4489457abb14373cd3816076.
```

```
2019-01-04T17:07:47.420509+00:00 amateen-em7 journal: SCILO_
SP:6954|logger:log_warning:127|WARNING|Agent id:
2ae22a6b4489457abb14373cd3816076 being given a return code: 2
```

Example /var/log/uwsgi/streamer.log for successful discovery in the Skylar One Distributed Architecture

```
10.234.196.19 - - [29/Sep/2017:14:04:52 +0000] "POST /api/update_
agent/agent/ HTTP/1.1" 200 89 "-" "python-requests/2.26.0"
```

Save incoming data for a specific device ID in the Skylar One Distributed Architecture

```
PYTHONPATH=/opt/em7/lib/python3:/opt/streamer_prime python3
/opt/streamer_prime/streamer_prime/manage.py agent_save_xml -d <agent
guid> -e true
```

Save incoming data for a specific device ID in the Skylar One Extended Architecture

```
kubectl exec -it $(kubectl get pods -l app=streamer -o jsonpath="{.items
[0].metadata.name}") -- python -m streamer agent_save_data --host_id
<host id> --enable true
```

You can find the host id from the ADS url, such as *https://<sl1_address>/ads/servers/13/system*). You can locate the files in the **/tmp/save_agent_data** directory.

Additional Troubleshooting Situations and Best Practices

The following situations might occur while configuring or working with agents:

Situation	Cause / Resolution
Two device records exist for the same device on the **Devices** page in Skylar One.	This situation occurs when Skylar One first discovered this device with SNMP, and then the agent was installed and started polling that device. This duplication of records also occurs if the agent was installed first, and then you ran an SNMP discovery. To address this issue, you can ***merge*** the device records using the classic user interface. For more information, see the "Managing a Single Device with the Device Administration Panel" chapter in the ***Device Management*** manual.
The SNMP device record has IPv4, but the agent device record has IPv6.	The agent reports all network interfaces to the message collector. The Message Collector uses the first "bound" IP address reported by the agent. To address this issue, you can manually edit the agent device

Situation	Cause / Resolution
	record in the "classic" user interface and update the IP address.
If you uninstall an agent and then run a different installation executable file, you still see the same organization ID for the agent record.	After you uninstall the agent, the **scilog.conf** file is left on the server in case the agent is reinstalled. Skylar One can reuse the same device record and maintain historical performance data for that agent. To address this issue, delete the **scilog.conf** file after you run the uninstallation. If you install this agent again, Skylar One assigns a new organization ID to the agent and creates a new device record.

Agent Communication with Skylar One

This section covers the various codes that are sent to the Skylar One agent when additional actions are needed.

Return Codes

The core method of communication between the Skylar One agent and Skylar One involves the data files the agent uploads to the Streamer. When the agent uploads a file to the Streamer, a number of conditions are checked and the return code given to the agent determines what action should be taken, if any.

Return Code	Meaning
200	No action needed.
409	The data file uploaded by the agent has a problem. The Streamer did not accept it and it should be deleted from the host.
503	Data uploads are occurring more quickly than normal. The agent should refrain from sending its next file until a specified amount of time has passed.
505	The agent should stop streaming.

Sub-codes

The Streamer communicates another set of actions in a JSON response included with the return code to the Skylar One agent.

JSON Response Field	Type	Meaning
send_system_file	boolean	Send a system config file as the next upload from this agent.
get_agent_config	boolean	Retrieve an updated agent config from the Streamer.
get_agent_update	boolean	Download a new version of the agent to update.
get_pd_config	boolean	Retrieve an updated polled data config.
get_log_config	boolean	Retrieve an updated log config.
purge_uploads	boolean	Delete any backed-up data and only upload new files.
set_data_backoff	integer	Do not upload a data file until the set number of seconds have

JSON Response Field	Type	Meaning
		passed.
set_log_backoff	integer	Do not upload a log file until the set number of seconds have passed.

© 2003 - 2026, ScienceLogic, Inc.

All rights reserved.

ScienceLogic™, the ScienceLogic logo, and ScienceLogic's product and service names are trademarks or service marks of ScienceLogic, Inc. and its affiliates. Use of ScienceLogic's trademarks or service marks without permission is prohibited.

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information herein, the information provided in this document may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described herein at any time without notice.

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

International: +1-703-354-1010