



ServiceNow Service Graph Connector SyncPack

Version 2.0.0

Table of Contents

Introduction to the ServiceNow Service Graph Connector SyncPack	4
What Can I Do with this SyncPack?	5
Contents of the SyncPack	5
Skylar Automation Applications	5
Steps	6
Installing the Service Graph Connector SyncPack	7
Prerequisites for ServiceNow SyncPacks	8
Skylar One and ServiceNow Terminology	9
Downloading and Installing the SyncPack	9
Importing the SyncPack	10
Installing the SyncPack	10
Installing the "Service Graph Connector for ScienceLogic" Application in ServiceNow	11
Configuring Applications for the SyncPack	13
Workflow for Configuring the SyncPack	14
Configuring ServiceNow	14
Configuring Skylar Automation	14
Configuring ServiceNow	14
Configuring the Service Graph Connector for ScienceLogic	14
Creating a ServiceNow Group	16
Creating a ServiceNow User	19
Configuring Skylar Automation	20
Creating and Aligning a Configuration Object	20
Creating a Configuration Object	21
Aligning a Configuration Object	24
Creating an OAuth2 Credential Record in ServiceNow	26
Managing Organizations	27
Syncing Devices from Skylar One to ServiceNow	27
Syncing File Systems from Skylar One to ServiceNow	31
Syncing Interfaces from Skylar One to ServiceNow	33
Scheduling PowerFlow Applications	36
Known Issues	38
Adding or Checking Hosting Dependent Relationships	38
Known Issue with VMware Sync	38
Additional Known Issues	39

Troubleshooting the SyncPack	40
Initial Troubleshooting Steps	41
Skylar One Skylar Automation	41
ServiceNow	41
Troubleshooting Specific to this SyncPack	41
Enabling Verbose Logging	42
Using the Results of Verbose Logging	42
Narrowing the Import Set	42
Import Table Endpoints	43
Resources for Troubleshooting	43
Useful Skylar Automation Ports	43
Helpful Docker Commands	43
Viewing Container Versions and Status	43
Restarting a Service	44
Stopping all Skylar Automation Services	44
Restarting Docker	44
Diagnosis Tools	44
Retrieving Additional Debug Information (Debug Mode)	45
ServiceNow API Endpoints for the Service Graph Connector	48
Staging Tables	49
Staging Table Names	49
URL Format	49
Example Payload	49
Example Response	50

Chapter

1

Introduction to the ServiceNow Service Graph Connector SyncPack

Overview

The "ServiceNow Service Graph Connector" SyncPack works with the ServiceNow Service Graph Connector method instead of the ServiceNow Identification and Reconciliation Engine (IRE) used by the existing "ServiceNow CMDB" SyncPack.

This SyncPack uses the "Service Graph Connector for ScienceLogic" certified application in ServiceNow.

This SyncPack is not compatible with versions of the "ServiceNow Incident" SyncPack older than version 4.0.0. You cannot upgrade to this SyncPack from the "ServiceNow CMDB" SyncPack.

This chapter covers the following topics:

<i>What Can I Do with this SyncPack?</i>	5
<i>Contents of the SyncPack</i>	5

What Can I Do with this SyncPack?

The "ServiceNow Service Graph Connector" SyncPack works with the ServiceNow Service Graph Connector method instead of the ServiceNow Identification and Reconciliation Engine (IRE) used by the existing "ServiceNow CMDB" SyncPack.

You can use this SyncPack to sync your monitored devices in Skylar One to your ServiceNow CMDB using the ServiceNow Service Graph Connector. Instead of adding classes and field mappings in Skylar Automation, you can set up those classes and field mappings in ServiceNow.

You can also use this SyncPack to sync file systems and their organizations and relationships from Skylar One to ServiceNow, and to sync network interfaces and their organizations and relationships from Skylar One to ServiceNow.

NOTE: Currently, all data is synced only from Skylar One to ServiceNow.
--

This SyncPack includes the following applications, which you can use to synchronize data between ServiceNow and Skylar One:

- **Device and Organization Sync.** The "Sync Devices from Skylar One to ServiceNow via SGC" application syncs devices and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing Devices from Skylar One to ServiceNow](#).
- **File System Sync.** The "Sync File Systems from Skylar One to ServiceNow via SGC" application syncs file systems and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing File Systems from Skylar One to ServiceNow](#).
- **Interface Sync.** The "Sync Interfaces from Skylar One to ServiceNow via SGC" application syncs network interfaces and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing Interfaces from Skylar One to ServiceNow](#).

In addition, this SyncPack uses a Guided Setup process in ServiceNow to help you quickly and easily set up these syncs.

Contents of the SyncPack

This section lists the contents of the "ServiceNow Service Graph Connector" SyncPack.

Skylar Automation Applications

The following Skylar Automation applications are included in this SyncPack:

- **Cache Skylar One Orgs and Device Classes.** Reads all existing Skylar One Device Classes and writes them to a cache. You must run this application before you run the "Sync Devices from Skylar One to ServiceNow via SGC" application.

- ***Sync Devices from Skylar One to ServiceNow via SGC.*** Syncs devices and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing Devices from Skylar One to ServiceNow](#).
- ***Sync File Systems from Skylar One to ServiceNow via SGC.*** Syncs file systems and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing File Systems from Skylar One to ServiceNow](#).
- ***Sync Interfaces from Skylar One to ServiceNow via SGC.*** Syncs network interfaces and their organizations and relationships from Skylar One to ServiceNow. For more information, see [Syncing Interfaces from Skylar One to ServiceNow](#).

Steps

The following steps are included in this SyncPack:

- Post CIs to ServiceNow
- Post Relationships to ServiceNow
- Process Skylar One Devices
- Pull and Process Advanced Topology from Skylar One for SGC
- Pull and Process File Systems from Skylar One
- Pull and Process Interfaces and IPs/subnets from Skylar One for SGC
- Pull Devices from Skylar One
- Query and Cache Skylar One Device Class info

TIP: You can use the "ServiceNow SyncPack" configuration object from the *ServiceNow BaseSyncPack* as a template for a configuration object for this SyncPack.

NOTE: This SyncPack requires you to download and install the new "Service Graph Connector for ScienceLogic" application. For more information, see [Installing the "Service Graph Connector for ScienceLogic" Application in ServiceNow](#).

Chapter

2

Installing the Service Graph Connector SyncPack

Overview

This chapter describes how to install the "ServiceNow Service Graph Connector" in Skylar Automation. This chapter also covers how to install the "Service Graph Connector for ScienceLogic" certified application in ServiceNow.

This chapter covers the following topics:

<i>Prerequisites for ServiceNow SyncPacks</i>	8
<i>Skylar One and ServiceNow Terminology</i>	9
<i>Downloading and Installing the SyncPack</i>	9
<i>Installing the "Service Graph Connector for ScienceLogic" Application in ServiceNow</i>	11

Prerequisites for ServiceNow SyncPacks

This section describes the prerequisites for the ServiceNow SyncPacks. For more information about the specific software versions required by a ServiceNow SyncPack, see the latest release notes for this SyncPack on the [SyncPack Manuals and Release Notes](#) page.

To install any of the ScienceLogic ServiceNow SyncPacks, you must have administrator access to both Skylar One and ServiceNow. Specifically, you will need:

- ScienceLogic administrator access to the Administration Portal
- ServiceNow administrator access

TIP: If you want to upload and install multiple ServiceNow SyncPacks at the same time, you should upload *all* of the SyncPacks first, and then install them to address any dependencies between the SyncPacks.

The following table lists the port access required by Skylar Automation and this SyncPack:

Source IP	Skylar Automation Destination	Skylar Automation Source Port	Destination Port	Requirement
Skylar Automation	Skylar One API	Any	TCP 443	Skylar One API Access
Skylar Automation	ServiceNow API	Any	TCP 443	ServiceNow API Access

NOTE: ScienceLogic highly recommends that you disable all firewall session-limiting policies. Firewalls will drop HTTPS requests, which results in data loss.

Skylar One and ServiceNow Terminology

The following table lists the different names for the shared elements in Skylar One and ServiceNow:

Skylar One	ServiceNow
Asset, Custom Attribute	Asset (ITAM)
Device	CI (Configuration Item)
Discovery Session	Service Request, Catalog Request
Event	Incident, Event, or Case (depending on the SyncPack you are using)
Alert	Event
Organization	Company, Domain
Schedule, Maintenance Schedule	Change Request, Change Schedule
Topology, Relationships, Dynamic Component Mapping and Relationships (DCM+R)	Dependency View, Affected CIs

Downloading and Installing the SyncPack

A SyncPack file has the **.whl** file extension type. You can download the SyncPack file from the ScienceLogic Support site.

NOTE: If you are installing or upgrading to the latest version of this SyncPack in an offline deployment, see [Installing or Upgrading in an Offline Environment](#) to ensure you install any external dependencies.

To locate and download the SyncPack:

1. Go to the ScienceLogic Support Center at <https://support.sciencelogic.com/s/>.
2. Go to the **SyncPacks** page (Skylar Automation > SyncPacks).
3. In the **Search** field, search for the SyncPack and select it from the search results. The **Release Version** page appears.
4. On the **[Files]** tab, click the down arrow next to the SyncPack version that you want to install, and select *Show File Details*. The **Release File Details** page appears.
5. Click the **[Download File]** button to download the SyncPack.

After you download the SyncPack, you can import it to your Skylar Automation system using the Skylar Automation user interface.

Importing the SyncPack

NOTE: You must import and install the "ServiceNow Base" SyncPack before uploading and installing this SyncPack.

To import a SyncPack in the Skylar Automation user interface:

1. On the **SyncPacks** page (☺) of the Skylar Automation user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.
2. Click **[Browse]** and select the **.whl** file for the SyncPack you want to install. You can also drag and drop a **.whl** file to the **Import SyncPack** page.
3. Click **[Import]**. Skylar Automation registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in Skylar Automation. For more information, see the following topic.

NOTE: You cannot edit the content package in a SyncPack published by ScienceLogic. You must make a copy of a ScienceLogic SyncPack and save your changes to the new SyncPack to prevent overwriting any information in the original SyncPack when upgrading.

Installing the SyncPack

WARNING: If you are *upgrading* to this version of the SyncPack from a previous version, make a note of any settings you made on the **Configuration** pane of the various Skylar Automation applications in this SyncPack, as these settings are *not* retained when you upgrade.

To activate and install a SyncPack in the Skylar Automation user interface:

1. On the **SyncPacks** page of the Skylar Automation user interface, click the **[Actions]** button (⋮) for the SyncPack you want to install and select *Activate & Install*. The **Activate & Install SyncPack** modal appears.

NOTE: If you try to activate and install a SyncPack that is already activated and installed, you can choose to "force" installation across all the nodes in the Skylar Automation system.

TIP: If you do not see the SyncPack that you want to install, click the Filter icon (≡) on the **SyncPacks** page and select *Toggle Inactive SyncPacks* to see a list of the imported PowerPacks.

2. Click **[Yes]** to confirm the activation and installation. When the SyncPack is activated, the **SyncPacks** page displays a green check mark icon (✓) for that SyncPack. If the activation or installation failed, then a red exclamation mark icon (!) appears.
3. For more information about the activation and installation process, click the highlighted version in the **Installed SyncPack** column for that SyncPack. For a successful installation, the "Activate & Install SyncPack" application appears, and you can view the Step Log for the steps. For a failed installation, go to the hidden "activate & Install SyncPack" application on the **Applications** page and check the step logs.
4. If you have other versions of the same SyncPack on your Skylar Automation system, you can click the **[Actions]** button (⋮) for that SyncPack and select *Change active version* to activate a different version other than the version that is currently running.

Installing the "Service Graph Connector for ScienceLogic" Application in ServiceNow

You must install the "Service Graph Connector for ScienceLogic" application on the ServiceNow instance to enable this SyncPack. You must first request the "Service Graph Connector for ScienceLogic" application from the ServiceNow Store, and then you can install it.

NOTE: You must have a ServiceNow HI Service Account to request this application and download it onto your ServiceNow instance.

To request and install the "Service Graph Connector for ScienceLogic" application:

1. Go to the ServiceNow Store at <https://store.servicenow.com/> and search for "ScienceLogic".
2. Select the "Service Graph Connector for ScienceLogic" application. The detail page for the application appears.
3. Click **Get** and log in with your HI credentials.
4. After the request is approved, log in to ServiceNow as an administrator and navigate to **Application Manager** (System Applications > Applications).
5. Click **Downloads** in the menu header or search for "ScienceLogic".
6. Click the version drop-down for the "Service Graph Connector for ScienceLogic" application listing to make sure you are using the correct version of the application that is compatible with your version of the *ServiceNow Service Graph Connector* SyncPack.
7. Click **Install** for the "Service Graph Connector for ScienceLogic" application. The installation is complete when the button changes to **Installed**.

8. In the filter navigator, search for "ScienceLogic" and locate the application in the navigation menu in the left pane.

NOTE: You might need to log out of ServiceNow and log in again to see the updated left-hand navigation menu.

Chapter

3

Configuring Applications for the SyncPack

Overview

This chapter describes how to configure Device Sync between Skylar One and ServiceNow using the ServiceNow Service Graph Connector.

This chapter covers the following topics:

<i>Workflow for Configuring the SyncPack</i>	14
<i>Configuring ServiceNow</i>	14
<i>Configuring Skylar Automation</i>	20
<i>Known Issues</i>	38

Workflow for Configuring the SyncPack

The following workflows describe how to configure ServiceNow and Skylar Automation to work with the "ServiceNow Service Graph Connector" SyncPack.

Configuring ServiceNow

1. [Configure the Service Graph Connector for ScienceLogic](#)
2. [Create a ServiceNow Group](#)
3. [Create a ServiceNow User](#)

Configuring Skylar Automation

1. [Install the ServiceNow Service Graph Connector SyncPack and its dependencies](#)
2. [Create a configuration object in the Skylar Automation user interface](#)
3. [Align the new configuration object with the following Skylar Automation applications:](#)
 - "Cache Skylar One Orgs and Device Classes"
 - "Sync Devices from Skylar One to ServiceNow via SGC"
 - "Sync File Systems from Skylar One to ServiceNow via SGC"
 - "Sync Interfaces from Skylar One to ServiceNow via SGC"
4. Run or [schedule](#) the following applications, in the following order:
 - "[Cache Skylar One Orgs and Device Classes](#)" and "[Sync Devices from Skylar One to ServiceNow via SGC](#)"
 - "[Sync File Systems from Skylar One to ServiceNow via SGC](#)"
 - "[Sync Interfaces from Skylar One to ServiceNow via SGC](#)"

Configuring ServiceNow

The following topics cover how to set up your ServiceNow instance to work with the "ServiceNow Service Graph Connector" SyncPack.

Configuring the Service Graph Connector for ScienceLogic

In ServiceNow, you can use the Guided Setup process to configure the Service Graph Connector for ScienceLogic.

1. In ServiceNow, go to **Service Graph Connectors > ScienceLogic > Guided Setup**. The ScienceLogic Integration with CMDB page appears.
2. In the **Integration Account** section, click **[Get Started]** and complete the tasks in order.

3. Click **[Configure]** button to configure each task.
4. After you complete each task, click **[Mark as Complete]**.

0% Complete

ScienceLogic Integration with CMDB > Integration Account

Assign

0%

0 / 4 Tasks completed

- Create Integration User
- Service Graph Connector for ScienceLogic Application Role
- CMDB Data Ingestion API Role
- Ready to Connect Powerflow

Integration Account

Setup the inbound integration user account.
For the most current [ScienceLogic Documentation](#).

Create Integration User

Last visited 18h ago by System Administrator

To complete this step you need to create a new user within Service Now to allow SL1 | Power Flow to have access to the Service Now instance.

- Create new User
- Set Time Zone to "GMT"
- Check the box for "Web Service Access Only"

This User will be used to interact with the certified application import tables

- SG-ScienceLogic Physical Devices (x_sclo_devicesync_device_import)
- SG-ScienceLogic VMware Devices (x_sclo_devicesync_device_import_vmware)

Service Graph Connector for ScienceLogic Application Role Skip Add Notes
Last visited 18h ago by System Administrator

To complete this step you will need to do one of two following options.

- Add **x_sclo_devicesync.admin** role to a Service Now group your newly created user already is apart of.
- Add **x_sclo_devicesync.admin** role directly to the new created user.

This will give the integration user the correct rights to interact with the certified application.

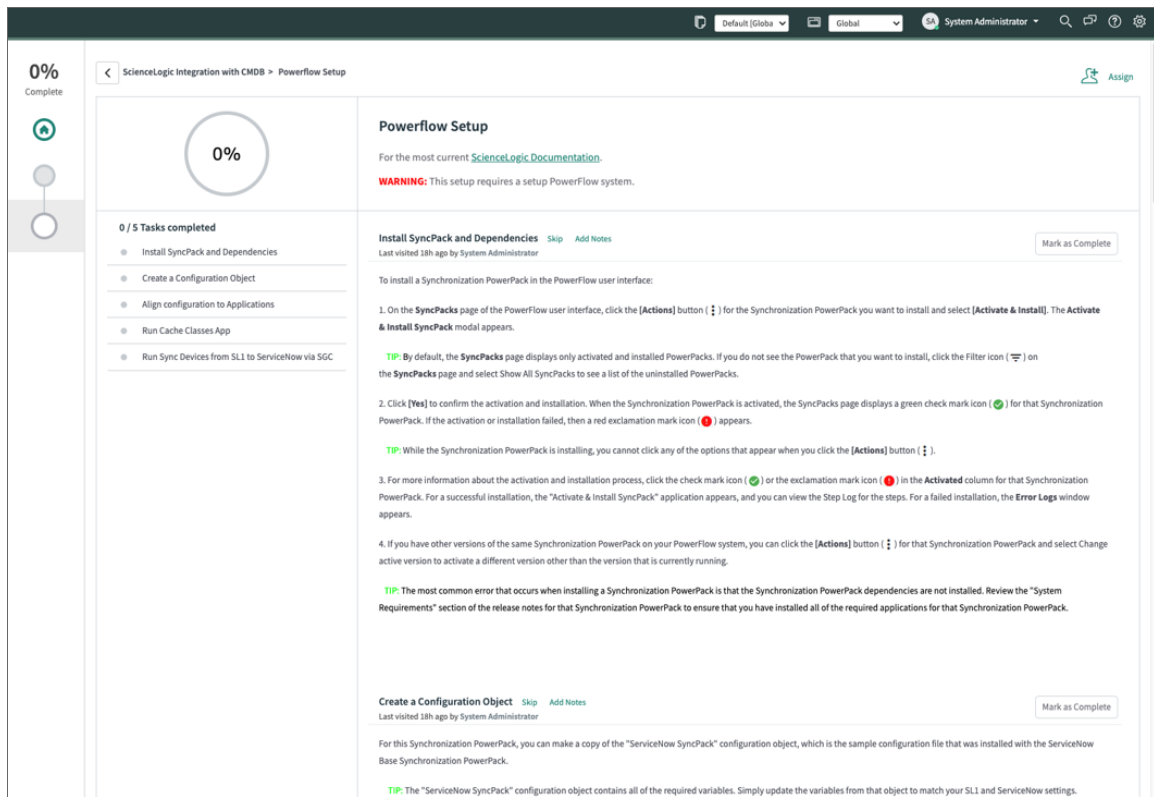
CMDB Data Ingestion API Role Skip Add Notes
Last visited 18h ago by System Administrator

To complete this step you will need to do one of two following options.

Add **cmdb.import.api.admin** role to a Service Now group your newly created user already is apart of.

Mark as Complete Configure

5. After you finish all the tasks in the Integration Account section, go to the **Skylar Automation Setup** section and click **[Get Started]**:



6. These Skylar Automation tasks are described in the following sections of this manual:
 - [Install the ServiceNow Service Graph Connector SyncPack and its dependencies.](#)
 - [Create a configuration object in the Skylar Automation user interface.](#)
 - [Align the new configuration file with the Skylar Automation applications for this SyncPack.](#)
 - [Run the "Cache Skylar One Orgs and Device Classes" application.](#)
 - [Run the "Sync Devices from Skylar One to ServiceNow via SGC" application.](#)
7. After you complete each task in Skylar Automation, click **[Mark as Complete]** in the **Skylar Automation Setup** section in ServiceNow.
8. When are done with all the tasks in the **Skylar Automation Setup** section, continue to the next topic, "Creating a ServiceNow Group".

Creating a ServiceNow Group

For best practice and security, create a dedicated ServiceNow account that has restricted access to only the groups, access control lists (ACLs), and roles needed for ScienceLogic integration.

To create a ServiceNow Account for ScienceLogic integration:

1. In ServiceNow, search in the filter navigator for "groups".
2. On the **Groups** page (System Security > Groups), click **[New]**. A **New record** page appears.

3. In the **New record** page, type the group name and any additional information. **Name** is the only required field:

The screenshot shows the 'Group New record' form. The 'Name' field is filled with 'Sciencelogic Service Accounts'. The 'Manager' field is filled with 'System Administrator'. The 'Group email' and 'Parent' fields are empty. The 'Description' field is a large text area. A 'Submit' button is located at the bottom left of the form.

4. Right-click the gray header and click **Save** to save the record.

The screenshot shows the 'Group New record' form with a right-click context menu open over the gray header. The menu options are: Save, Configure, Export, Create Favorite, Copy URL, Copy sys_id, and Reload form. The 'Save' option is highlighted.

5. At the bottom of the Group form, select the **[Roles]** tab and click **[Edit:]**

The screenshot shows the 'Roles' tab in the 'Group Members' section. The 'Edit...' button is highlighted. The table below shows no records to display.

6. Search for *cmdb_import_api_admin* and *x_sclo_devicesync.admin* and move them to the **Roles List** column using the arrow buttons:

The screenshot shows the 'Add Filter' and 'Run filter' buttons at the top. Below them are dropdowns for 'choose field', 'oper', and 'value'. The 'Collection' list on the left includes roles like 'action_category_creator', 'action_designer', 'activity_admin', etc. The 'Roles List' on the right shows 'cmdb_import_api_admin' and 'x_sclo_devicesync.admin' being moved. A red box highlights the right arrow button between the two lists. The 'Cancel' and 'Save' buttons are at the bottom.

7. Click **[Save]**. Your ServiceNow Group now has two assigned Roles:

The screenshot shows the 'PowerFlow Integration ETL' group configuration page. The 'Roles' tab is selected, showing a table with two roles: 'cmdb_import_api_admin' and 'x_sclo_devicesync.admin'. Both roles are highlighted with red boxes. The table has columns for 'Created', 'Role', 'Granted by', and 'Inherits'. The 'Created' column shows dates '2020-11-09 16:01' and '06-18 14:30'. The 'Role' column shows 'cmdb_import_api_admin' and 'x_sclo_devicesync.admin'. The 'Granted by' column shows '(empty)' for both. The 'Inherits' column shows 'true' for both. The 'Actions on selected rows...' dropdown is at the bottom left.

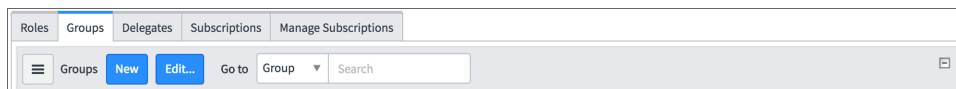
8. Next, create a ServiceNow user to use with this Group. See the following procedure for the details.

Creating a ServiceNow User

NOTE: The ServiceNow user you create in this procedure will *not* be able to log into the ServiceNow user interface with the username and password you give this user. However, you will use the username and password in the relevant configuration objects in the Skylar Automation user interface to run applications. For more information about configuration objects, see [Creating and Aligning a Configuration Object](#).

To create a ServiceNow Account for ScienceLogic Incident management:

1. In ServiceNow, search in the filter navigator for "users".
2. On the **Users** page (System Security > Users), click **[New]**. A **New record** page appears.
3. Complete the following fields:
 - **User ID.** Type a user ID. Required.
 - **First Name.** Type the user's first name.
 - **Last Name.** Type the user's last name.
 - **Password.** Type a password. Required.
 - **Active.** Select this checkbox. Required.
 - **Web Service Access Only.** Select this checkbox. Required.
 - **Time Zone.** Select *GMT*. Required.
 - **Date Format.** Select *System (yyyy-MM-dd)*.
4. Right-click the gray header and click *Save* to save the user.
5. Select the **[Groups]** tab at the bottom of the record and click the **[Edit]** button:



- Find the group you created previously and move the group to the right-hand column using the arrow buttons:

Collection

Search sciencelogic

--None--

Groups List

Service Account

Sciencelogic Service Accounts

>

<

Cancel Save

Name Sciencelogic Service Accounts

- Click **[Save]**. After the user has been added to the group, you can see their Roles and Groups at bottom of the record:

Roles (18) Groups (1) Delegates Subscriptions Manage Subscriptions

Roles Edit... Go to Role Search

1 to 18 of 18

NOTE: As a best practice, you should use a non-administrator ServiceNow user for the Skylar Automation configuration object.

Configuring Skylar Automation

The following topics cover how to set up your Skylar Automation instance to work with the "ServiceNow Service Graph Connector" SyncPack.

Creating and Aligning a Configuration Object

A **configuration object** supplies the login credentials and other required information needed to execute the steps for an application in Skylar Automation. The **Configurations** page (⚙️) of the Skylar Automation user interface lists all available configuration objects for that system.

You can create as many configuration objects as you need. A Skylar Automation application can only use one configuration object at a time, but you can use (or "align") the same configuration object with multiple Skylar Automation applications.

To use this SyncPack, you will need to create one or more configuration objects in the Skylar Automation user interface and align that configuration object to the applications that let you sync data between Skylar One and ServiceNow.

NOTE: Depending on your Skylar One and ServiceNow environments, you might be able to use the same configuration object with other ServiceNow SyncPacks.

NOTE: When either multiple Skylar One stacks or multiple ServiceNow systems are involved with Skylar Automation, you should create an individual configuration object for each Skylar One stack or ServiceNow system. Next, create an individual schedule for each configuration object. Each schedule should use a configuration object that is specific to that single Skylar One stack or ServiceNow system. Creating copies of a Skylar Automation application from a SyncPack for the purpose of distinguishing between domains is not supported, and will result in issues on upgrades.

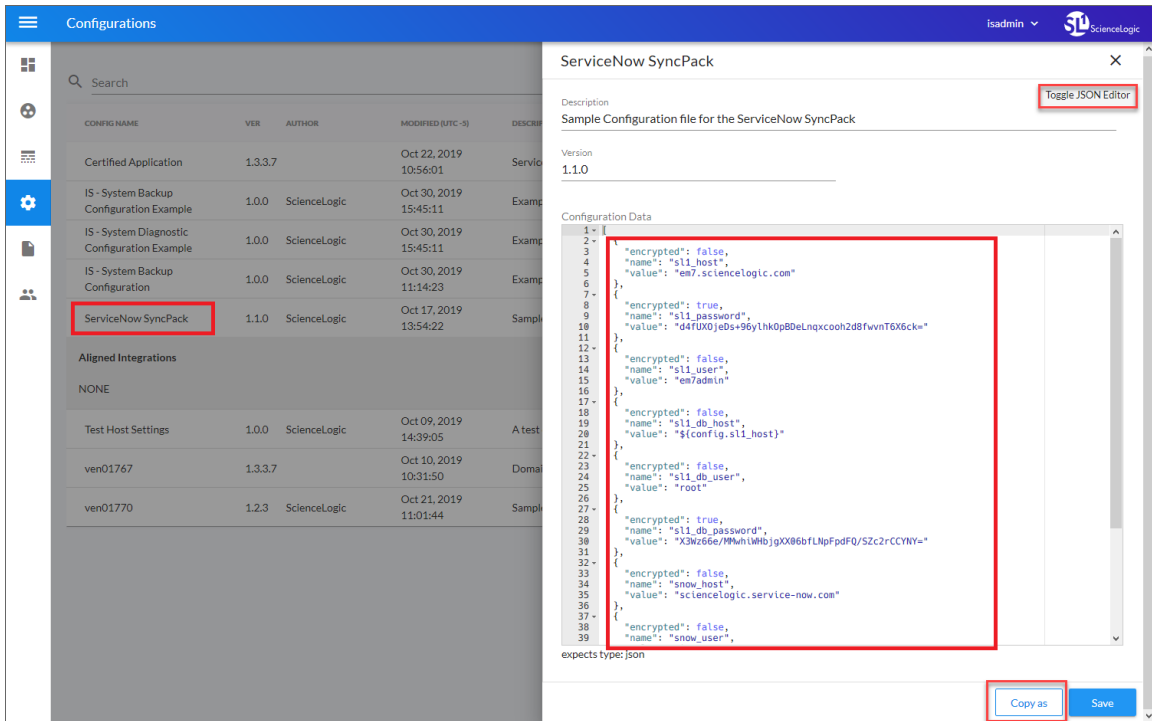
Creating a Configuration Object

For this SyncPack, you can make a copy of the "ServiceNow SyncPack" configuration object, which is the sample configuration file that was installed with the "ServiceNow Base" SyncPack.

TIP: The "ServiceNow SyncPack" configuration object contains all of the required variables. Make a copy of the configuration object and update the variables from that object to match your Skylar One and ServiceNow settings.

To create a configuration object based on the "ServiceNow SyncPack" configuration object:

1. In the Skylar Automation user interface, go to the **Configurations** page (⚙️).
2. Click the **[Actions]** button (⋮) for the "ServiceNow SyncPack" configuration object and select **Edit**.
The **Configuration** pane appears:



TIP: Click **[Toggle JSON Editor]** to show the JSON code. Click the button again to see the fields.

3. Click **[Copy as]**. The **Create Configuration** pane appears.

TIP: This step is required. Do *not* use the original configuration object to run Skylar Automation applications.

4. Complete the following fields:
 - **Friendly Name.** Name of the configuration object that will display on the **Configurations** page.
 - **Description.** A brief description of the configuration object.
 - **Author.** User or organization that created the configuration object.
 - **Version.** Version of the configuration object.

5. In the **Configuration Data** field, include the required block of code to ensure that the applications aligned to this configuration object do not fail:

```
{
  "encrypted": false,
  "name": "<sl1_db_host>",
  "value": "${<config.sl1_host>}"
}
```

For example:

```
{
  "encrypted": false,
  "name": "sl1_db_host",
  "value": "10.2.11.42"
}
```

TIP: If you are using IPv6 for IP addresses, wrap the IP string in brackets, such as **https://[2001:db8:3333:4444:5555:6666:7777:8888]**

TIP: Click **[Toggle JSON Editor]** to show the JSON code. Click the button again to see the fields. You can also click **[Add Value]** and add a new name-value pair in the **Configuration Data Values** section.

NOTE: If you are using Skylar One with an External Database (Skylar One Extended architecture or a cloud-based architecture), update the **"value"** of that block of code to be the host of your database. This field accepts IP addresses. For example: **"value": "db.sciencelogic.com"**. If you are *not* using the Skylar One Extended architecture or a cloud-based architecture, you do not need to make any changes to the block of code other than pasting the code into the configuration object.

6. In the **Configuration Data Values** field, update the default variable definitions to match your Skylar Automation configuration.

NOTE: The **region** value is a user-defined variable that identifies your Skylar One instance within ServiceNow.

7. To create a configuration variable in the JSON Editor, define the following keys:

- **encrypted.** Specifies whether the value will appear in plain text or encrypted in this JSON file. If you set this to "true", when the value is uploaded, Skylar Automation encrypts the value of the variable. The plain text value cannot be retrieved again by an end user. The encryption key is unique to each Skylar Automation system. The value is followed by a comma.
 - **name.** Specifies the name of the configuration file, without the JSON suffix. This value appears in the user interface. The value is surrounded by double-quotes and followed by a comma.
 - **value.** Specifies the value to assign to the variable. The value is surrounded by double-quotes and followed by a comma.
8. If you want to use OAuth2 for authentication with ServiceNow, complete the following Configuration Data Values fields:
- **snow_oauth_client_id.** Enter the OAuth2 Client ID from ServiceNow.
 - **snow_oauth_client_secret.** Enter the OAuth2 Client secret from ServiceNow.
 - **snow_oauth_token_url.** Enter the full authentication URL, including host and protocol from ServiceNow. For example, "https://<test-instance-name>.service-now.com/oauth_token.do".
 - **snow_auth_method:** Enter *oauth* or *http_basic*. If no value is provided, *http_basic* will be used for connection.

NOTE: The configuration options listed above are included by default with the sample configuration object provided in the "ServiceNow Base" SyncPack. The configuration options are only required in the configuration object if you plan to use OAuth2 to authenticate. If the values are not present in the configuration object, normal "http_basic" authentication will be used.

9. Click **[Save]**. You can now align this configuration object with one or more applications.

Aligning a Configuration Object

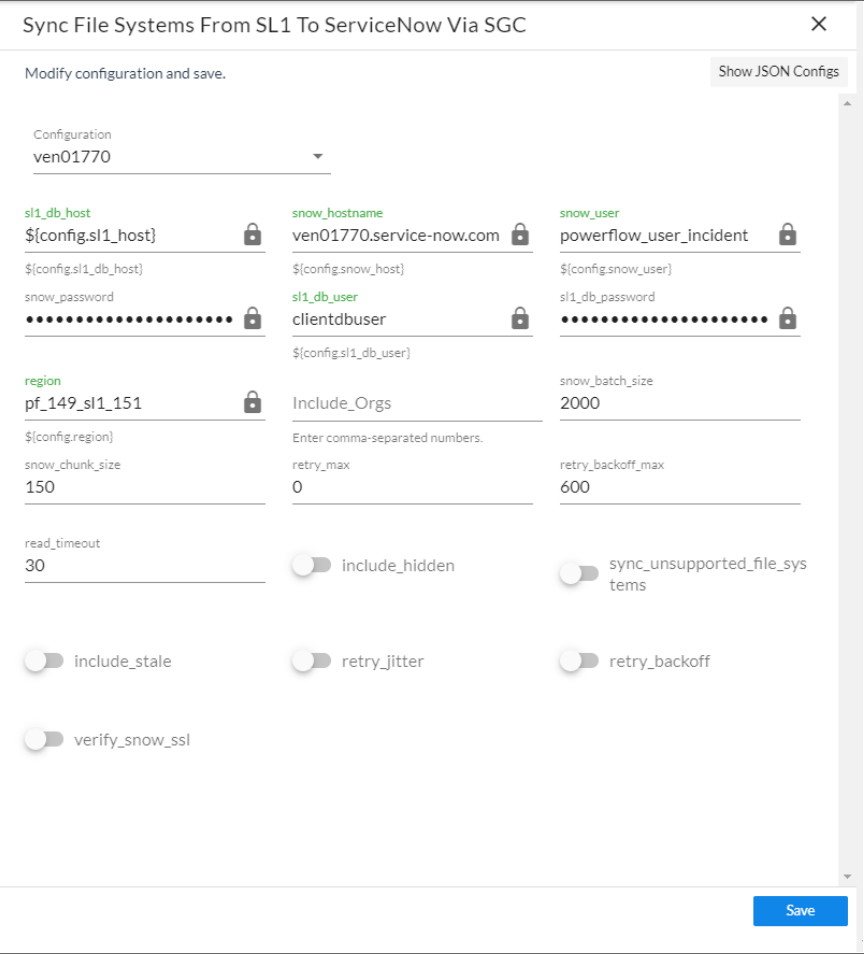
Before you can run the applications in this SyncPack, you must first "align" a configuration object with those applications. The applications you need to align include the following:

- **Cache Skylar One Orgs and Device Classes.** Reads all existing Skylar One Device Classes and writes them to a cache. You must run this application before you run the "Sync Devices from Skylar One to ServiceNow via SGC" application.
- **Sync Devices from Skylar One to ServiceNow via SGC.** Syncs devices and their organizations and relationships from Skylar One to ServiceNow.

TIP: The first time you run both of these applications, ScienceLogic recommends that you run them on Debug Mode to help you identify any misconfigurations.

To align a configuration object with an application:

1. From the **Applications** page of the Skylar Automation user interface, open the relevant application and click **[Configure]** (). The **Configurations** pane for that application appears:



Sync File Systems From SL1 To ServiceNow Via SGC

Modify configuration and save. [Show JSON Configs](#)

Configuration
ven01770

sl1_db_host \${config.sl1_host}	snow_hostname ven01770.service-now.com	snow_user powerflow_user_incident
\$(config.sl1_db_host)	\$(config.snow_host)	\$(config.snow_user)
snow_password	sl1_db_user clientdbuser	sl1_db_password
\$(config.sl1_db_user)		
region pf_149_sl1_151	Include_Orgs Enter comma-separated numbers.	snow_batch_size 2000
\$(config.region)	retry_max 0	retry_backoff_max 600
snow_chunk_size 150		
read_timeout 30	☐ include_hidden	☐ sync_unsupported_file_sys_tems
☐ include_stale	☐ retry_jitter	☐ retry_backoff
☐ verify_snow_ssl		

Save

2. From the **Configuration** drop-down, select the configuration object you want to use.
3. Click **[Save]** to align that configuration with the application.
4. Wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** automatically closes after this message appears.
5. Repeat this process for the remaining Skylar Automation applications in this SyncPack.

NOTE: The values for **eventDetails** and the other parameters that appear in the **Configuration** pane with a padlock icon () are populated by the configuration object. Do not modify these values.

Creating an OAuth2 Credential Record in ServiceNow

In order to use OAuth2 for authentication with ServiceNow, you must create an OAuth2 credential record in ServiceNow. To configure the ServiceNow credential that will be used by the Connector Instance:

1. Navigate to **System OAuth > Application Registry**. The **Application Registries** page appears.
2. Click **[New]**.
3. Select **Create an OAuth API endpoint for external clients**. A new record appears.
4. Complete the following fields on the new record:
 - **Name**. Type a unique name for the credential. Required.
 - **Client ID**. The Client ID is automatically generated by the ServiceNow OAuth server.
 - **Client Secret**. Leave this empty. ServiceNow will auto-generate this when the record is saved.
 - **Refresh Token Lifespan**. Enter the length of time in seconds the Refresh Token will be valid.
 - **Access Token Lifespan**. Type the length of time in seconds that the Access Token will be valid. ScienceLogic recommends setting this to 3,600 seconds to avoid known issues for longer ServiceNow REST interactions.

NOTE: In a scenario where the **[Access Token Lifespan]** value is shorter than the duration of a Skylar Automation step that makes multiple REST interactions with ServiceNow, the access token will expire and need to be refreshed. As a result, retries were added to several Skylar Automation steps where this issue may occur. This issue will hopefully be addressed in future versions of the Base Steps SyncPack.

5. Under the **Auth Scope** section at the bottom of the page, double click **Insert a new row**.
6. In the search box that appears, click the magnifying glass icon, select the *useraccount* record, and click the checkmark icon to save.
7. Click **[Submit]** to save the new record.

The screenshot shows the 'Application Registries' page for 'PowerFlow Token'. It includes a 'More Info' section with details about Name, Client ID, Client Secret, Refresh Token Lifespan, Access Token Lifespan, Redirect URL, and Enforce Token Restriction. The configuration form contains fields for Name (PowerFlow Token), Client ID (85abad6bed82c2507f0168b91a1c1bcf), Client Secret (masked), Redirect URL, Logo URL, Public Client (unchecked), Application (Global), Accessible from (All application scopes), Active (checked), Refresh Token Lifespan (8,640,000), Access Token Lifespan (3,600), and Login URL. A 'Comments' text area is also present. Below the form is an 'Auth Scopes' table with one row for 'useraccount' and an 'Insert a new row...' option.

Managing Organizations

Two fields are used to select a company. The first non-null value returned will be used for the company. These fields include the following, in order:

- **CRM ID.** The **org_crm_id** supplied is validated using a lookup of the **sys_id** to validate that the **sys_id** match is found on the core company table.
- **Name:** The **org_name** is used in a lookup to find a matching name on the core company table.

Syncing Devices from Skylar One to ServiceNow

To perform a Device Sync between Skylar One and ServiceNow, run the following applications in the Skylar Automation user interface:

- **Cache Skylar One Orgs and Device Classes.** Reads all existing Skylar One Organizations and Device Classes and writes them to a cache.
- **Sync Devices from Skylar One to ServiceNow via SGC.** Syncs devices and their organizations and relationships from Skylar One to ServiceNow.

NOTE: Currently, all data is synced from Skylar One to ServiceNow only.

To sync Skylar One devices with ServiceNow:

1. In the Skylar Automation user interface, select the "Cache Skylar One Orgs and Device Classes" application from the **Applications** page, click **[Configure]** , align a configuration object, and then click **[Run]** .

2. Select the "Sync Devices from Skylar One to ServiceNow via SGC" application from the **Applications** page and click **[Configure]** (⚙️). The **Configuration** pane appears:

Sync Devices From SL1 To ServiceNow Via SGC

Modify configuration and save. Show JSON Configs

Configuration
ven01770

sl1_hostname 10.2.11.151	sl1_db_host \${config.sl1_host}	snow_hostname ven01770.service-now.com
\$(config.sl1_host)	\$(config.sl1_db_host)	\$(config.snow_host)
sl1_user em7admin	snow_user powerflow_user_incident	sl1_password
\$(config.sl1_user)	\$(config.snow_user)	sl1_db_password
snow_password	sl1_db_user clientdbuser	sl1_db_password
\$(config.sl1_db_user)	read_timeout 30	Include_Orgs Enter comma-separated numbers.
region pf_149_sl1_151	sl1_chunk_size 500	sl1_url_override em7.sciencelogic.com
\$(config.region)	selected_devices	lookup_chunk_size 1000
Include_CUGs Enter comma-separated numbers. excluded_devices device_name_a,device_id_232	all_devices_template {{ {"billing_id": device.org.billing_	snow_chunk_size 150
snow_batch_size 2000	retry_max 0	exclude_inactive ☒
retry_backoff_max 600		

3. Complete the following fields, as needed:
 - **Configuration.** Select the configuration object with the relevant Skylar One and ServiceNow credentials to align with this application. You cannot edit fields that are populated by the configuration object. Required.

NOTE: The **region** field (along with other fields related to user names and passwords) is populated by the configuration object you aligned with this application. The region value must match the value in the **Skylar One Region** field in ServiceNow. If you need to update this value, you will need to define the **region** variable in the configuration object that is aligned with this application, or align a different configuration object that has the correct **region** value.

- **read_timeout.** Specify the maximum amount of time in seconds that the application should wait for a response before timing out. The default is 30 seconds. If the Device Sync continually times out, you might want to increase the timeout to 120 seconds, as this sync can be data-intensive.

TIP: Scroll past the *mappings* section to access the next set of fields. For more information about the *mappings* section, go to step 6.

- **Include_Orgs.** If you want to include Skylar One Organizations in the device sync, add the Organization IDs from Skylar One in this field, separated by commas. Leave this field empty to sync all Skylar One Organizations.
- **Include_CUGs.** If you want to include Skylar One Collector Groups (CUGs) in the device sync, add the Collector Group IDs from Skylar One in this field, separated by commas. Leave this field empty to sync all Skylar One Collector Groups.
- **sl1_chunk_size.** Specify the number of devices to include in each chunk sent to in the Skylar One GraphQL query when you run this application. The default chunk size is 500 devices.
- **sl1_url_override.** Update this field if you want to use an URL that is different from the standard Skylar One URL that gets sent to the ServiceNow CI record. Optional.
- **excluded_devices.** Type a list of comma-separated device names or device IDs for any devices that you want to exclude from the device sync. Optional.
- **selected_devices.** If you want to sync a sub-set of all discovered devices, type a comma-separated list of the Device IDs from Skylar One for only the devices that you want to sync. Leave this field empty to sync all Skylar One devices.
- **lookup_chunk_size.** Specify the number of devices in each chunk when pulling the device cache from Couchbase. The default is 1000 devices.
- **max_count_for_relationships.** Defines the maximum number of ids to query for sets of relationships in a single chunk.
- **all_devices_template.** Add a Jinja2 Template to apply to all devices. The result must be a dictionary (Key | Value pair) and will then be included in each JSON device object.
- **snow_chunk_size.** Specify the number of devices per chunk to post to ServiceNow when you run this application. The default chunk size is 2000 devices.

TIP: Scroll down to the bottom of the **Configuration** pane to access the following fields.

- **retry_max.** The maximum number of times Skylar Automation will retry to execute the step before it stops retrying and logs a step failure. For example, if **retry_max** is 3, PowerFlow will retry after 1 second, then 2 seconds, then 4 seconds, and stop if the last retry fails. The default is 0.
- **retry_backoff_max.** The maximum time interval for the **retry_backoff** option, in seconds. For example, This means, if you have **retry_max** set to 15, the delays will be 1, 2, 4, 8, 16, 32, 64, 120, 240, 480, 600, 600, 600, 600, and 600. The default is 600.

- **exclude_inactive.** Select this option to prevent syncing devices to ServiceNow that are disabled, unavailable, or in maintenance. By default, this field is not selected.
- **enable_device_active.** Select this option to enable the **Device Active** block in the device GraphQL query, which contains information about the active state of the Skylar One device. By default, this field is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template.
- **enable_asset_networks.** Select this option to enable the **assetNetworks** block in the device GraphQL query, which returns a list of asset networks. By default, this field is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template.

WARNING: Please note that enabling this option might cause performance issues on the Skylar One side.

- **retry_jitter.** Instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. The default is unselected.
 - **retry_backoff.** Instead of using a defined interval between retries, Skylar Automation will incrementally increase the interval between retries. The default is unselected.
 - **enable_advanced_topology.** Toggle on (blue) this option to enable the advanced topology query when you run this application.
 - **verify_snow_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
 - **gql_filter.** Using JSON, define a custom GraphQL filter to apply to the devices in the sync. To test a GraphQL filter, go to the GraphiQL interface in Skylar One by typing the URL or IP address for Skylar One in a browser, add **/gql** to the end of the URL or IP address, and press **[Enter]**. Search the built in GraphQL docs for **DeviceSearch** and determine how to set up your custom filter. An example of a Device Search query using JSON: `{"name": {"doesNotBeginWith": "jc-is-ma"}}`
4. The **sgc_uri_categories** and the **sgc_category_overrides** sections lets you send specific Skylar One Device Classes to a custom transform target within ServiceNow:
- In the **sgc_uri_categories** section, you can map from a category name or a tag to a ServiceNow import table.
 - In the **sgc_category_overrides** section, you can map from the category names or tags in the first section to Device Classes.

NOTE: You can use the default settings in these two sections without making any changes. ScienceLogic recommends that only advanced users edit or create new mappings in these sections. Any Device Classes that are not listed in these sections will be included in the default "Hardware" rule.

5. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.

6. Run the "Sync Devices from Skylar One to ServiceNow via SGC" application. If this is the first time you are running this application, run it a second time to build the internal cache.

Syncing File Systems from Skylar One to ServiceNow

To perform a File System Sync between Skylar One and ServiceNow, run the following applications in the Skylar Automation user interface:

- **Cache Skylar One Orgs and Device Classes.** Reads all existing Skylar One Organizations and Device Classes and writes them to a cache.
- **Sync File Systems from Skylar One to ServiceNow via SGC.** Syncs file systems and their organizations and relationships from Skylar One to ServiceNow.

NOTE: Currently, all data is synced from Skylar One to ServiceNow only.

To sync Skylar One file systems with ServiceNow:

1. In the Skylar Automation user interface, select the "Cache Skylar One Orgs and Device Classes" application from the **Applications** page, click **[Configure]** () , align a configuration object, and then click **[Run]** ().

2. Select the "Sync File Systems from Skylar One to ServiceNow via SGC" application from the **Applications** page and click **[Configure]** (⚙️). The **Configuration** pane appears:

Sync File Systems From SL1 To ServiceNow Via SGC

Modify configuration and save. Show JSON Configs

Configuration: ven01770

sl1_db_host \${config.sl1_host}	sl1_db_hostname ven01770.service-now.com	snow_user powerflow_user_incident
\$(config.sl1_db_host)	\$(config.snow_host)	\$(config.snow_user)
snow_password	sl1_db_user clientdbuser	sl1_db_password
\$(config.sl1_db_user)		
region pf_149_sl1_151	Include_Orgs	snow_batch_size 2000
\$(config.region)	Enter comma-separated numbers.	
snow_chunk_size 150	retry_max 0	retry_backoff_max 600
read_timeout 30	☐ include_hidden	☐ sync_unsupported_file_sys tems
☐ include_stale	☐ retry_jitter	☐ retry_backoff
☐ verify_snow_ssl		

Save

3. Complete the following fields, as needed:
 - **Configuration.** Select the configuration object with the relevant Skylar One and ServiceNow credentials to align with this application. You cannot edit fields that are populated by the configuration object. Required.

NOTE: The **region** field (along with other fields related to user names and passwords) is populated by the configuration object you aligned with this application. The region value must match the value in the **Skylar One Region** field in ServiceNow. If you need to update this value, you will need to define the **region** variable in the configuration object that is aligned with this application, or align a different configuration object that has the correct **region** value.

- ***Include_Orgs***. If you want to include a specific set of Skylar One Organizations in the sync, add the Organization IDs from the Skylar One **Organizations** page (Registry > Accounts > Organizations) in this field, separated by commas. If this field is enabled, Skylar Automation will pull only the Organizations listed in this field; Skylar Automation does not pull all Organizations and then drop those not on the list. Leave this field empty to sync *all* Skylar One Organizations. Optional.
- ***snow_chunk_size***. Specify the number of objects to send in each chunk or in each sub-payload with a batch. The default is 2000 objects per chunk or sub-payload.
- ***retry_max***. The maximum number of times Skylar Automation will retry to execute the step before it stops retrying and logs a step failure. For example, if *retry_max* is 3, PowerFlow will retry after 1 second, then 2 seconds, then 4 seconds, and stop if the last retry fails. The default is 0.
- ***retry_backoff_max***. The maximum time interval for the ***retry_backoff*** option, in seconds. For example, if you have ***retry_max*** set to 15, the delays will be 1, 2, 4, 8, 16, 32, 64, 120, 240, 480, 600, 600, 600, 600, and 600. The default is 600.
- ***read_timeout***. Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 30 seconds.
- ***include_hidden***. Select this option to sync hidden file systems. This filter is only applied on the Skylar One side of the sync. By default, this option is not selected.
- ***sync_unsupported_file_systems***. Select this option to send file systems which have a parent device class of Network Switch or Network Router. By default, this option is not selected.
- ***include_stale***. Select this option to sync stale file systems. This filter is only applied on the Skylar One side of the sync. By default, this option is not selected.

WARNING: Be cautious when applying the ***include*** options, as enabling one or both options might cause unwanted disconnects if the File System Sync is run with these options enabled and then run with the options disabled.

4. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
5. Run the "Sync File Systems from Skylar One to ServiceNow via SGC" application.

Syncing Interfaces from Skylar One to ServiceNow

To perform an Interface Sync between Skylar One and ServiceNow, run the following applications in the Skylar Automation user interface:

- ***Cache Skylar One Orgs and Device Classes***. Reads all existing Skylar One Organizations and Device Classes and writes them to a cache.
- ***Sync Interfaces from Skylar One to ServiceNow via SGC***. Syncs network interfaces and their organizations and relationships from Skylar One to ServiceNow.

NOTE: Currently, all data is synced from Skylar One to ServiceNow only.

To sync Skylar One network interfaces with ServiceNow:

1. In the Skylar Automation user interface, select the "Cache Skylar One Orgs and Device Classes" application from the **Applications** page, click **[Configure]** (⚙️), align a configuration object, and then click **[Run]** (▶️).
2. Select the "Sync Interfaces from Skylar One to ServiceNow via SGC" application from the **Applications** page and click **[Configure]** (⚙️). The **Configuration** pane appears:

Sync Interfaces From SL1 To ServiceNow Via SGC

Modify configuration and save. Show JSON Configs

Configuration
ven01770

sl1_hostname 10.2.11.151	snow_hostname ven01770.service-now.com	sl1_db_host \${config.sl1_host}
\${config.sl1_host}	\${config.snow_host}	\${config.sl1_db_host}
sl1_user em7admin	snow_user powerflow_user_incident	sl1_password
\${config.sl1_user}	\${config.snow_user}	sl1_db_password
snow_password	sl1_db_user clientdbuser	
	\${config.sl1_db_user}	
region pf_149_sl1_151	sl1_chunk_size 500	read_timeout 30
\${config.region}		
Include_Orgs	snow_batch_size 2000	snow_chunk_size 150
Enter comma-separated numbers.		
retry_max 0	retry_backoff_max 600	☒ enabled_only
☐ retry_jitter	☐ retry_backoff	☐ verify_snow_ssl

sql_filter
1 {}

Save

3. Complete the following fields, as needed:

- **Configuration.** Select the configuration object with the relevant Skylar One and ServiceNow credentials to align with this application. You cannot edit fields that are populated by the configuration object. Required.

NOTE: The **region** field (along with other fields related to user names and passwords) is populated by the configuration object you aligned with this application. The region value must match the value in the **Skylar One Region** field in ServiceNow. If you need to update this value, you will need to define the **region** variable in the configuration object that is aligned with this application, or align a different configuration object that has the correct **region** value.

- **sl1_chunk_size.** Specify the number of interfaces to pull from Skylar One in each chunk. The default is 500.
- **read_timeout.** Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 30 seconds.
- **Include_Orgs.** If you want to include a specific set of Skylar One Organizations in the sync, add the Organization IDs from the Skylar One **Organizations** page (Registry > Accounts > Organizations) in this field, separated by commas. If this field is enabled, Skylar Automation will pull only the Organizations listed in this field; Skylar Automation does not pull all Organizations and then drop those not on the list. Leave this field empty to sync *all* Skylar One Organizations. Optional.
- **snow_chunk_size.** Specify the number of objects to send in each chunk or in each sub-payload with a batch. The default is 2000 objects per chunk or sub-payload.
- **max_count_for_relationships.** Defines the maximum number of ids to query for sets of relationships in a single chunk.
- **retry_max.** The maximum number of times Skylar Automation will retry to execute the step before it stops retrying and logs a step failure. For example, if **retry_max** is 3, PowerFlow will retry after 1 second, then 2 seconds, then 4 seconds, and stop if the last retry fails. The default is 0.
- **retry_backoff_max.** The maximum time interval for the **retry_backoff** option, in seconds. For example, if you have **retry_max** set to 15, the delays will be 1, 2, 4, 8, 16, 32, 64, 120, 240, 480, 600, 600, 600, 600, and 600. The default is 600.
- **enabled_only.** Select this option to only sync interfaces where the **administrationStatus** field from Skylar One is set to "Up".
- **enable_advanced_topology.** Toggle on (blue) this option to enable the advanced topology query when you run this application.
- **gql_filter.** Using JSON, you can optionally define a custom GraphQL filter to apply to the interfaces in the sync. To test out a GraphQL filter, go to the GraphiQL interface in Skylar One by typing the URL or IP address for Skylar One in a browser, add **/gql** to the end of the URL or IP address, and press **[Enter]**. Search the built in GraphQL docs for **InterfaceSearch** and determine how to set up your custom filter.

4. The **sgc_uri_categories** and the **sgc_category_overrides** sections lets you send specific Skylar One Device Classes to a custom transform target within ServiceNow:
 - In the **sgc_uri_categories** section, you can map from a category name or a tag to a ServiceNow import table.
 - In the **sgc_category_overrides** section, you can map from the category names or tags in the first section to Device Classes.

NOTE: You can use the default settings in these two sections without making any changes. ScienceLogic recommends that only advanced users edit or create new mappings in these sections.

5. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
6. Run the "Sync Interfaces from Skylar One to ServiceNow via SGC" application.

Scheduling PowerFlow Applications

You can create one or more schedules for a single application in the Skylar Automation user interface. When creating each schedule, you can specify the queue and the configuration file for that application.

To schedule an application:

1. On the **Applications** page () , click the **[Schedule]** button for the application you want to schedule. The **Scheduler** window appears.
2. In the **Schedule List** pane, click the down arrow icon () next to an existing schedule to view the details for that schedule.
3. In the **Schedule Creator** pane, complete the following fields for the default **Frequency** setting:
 - **Schedule Name**. Type a name for the schedule.
 - **Frequency in seconds**. Type the number of seconds per interval that you want to run the application.
 - **Custom Parameters**. Type any JSON parameters you want to use for this schedule, such as information about a configuration file or mappings.

- To use a cron expression, click the **Switch to Cron Expression** toggle to turn it blue. If you select this option, you can create complicated schedules based on minutes, hours, the day of the month, the month, and the day of the week:

Schedule Creator

Switch to Frequency in Seconds ☒

For more information about schedules, see [Scheduling a PowerFlow Application](#).

Schedule Name

A unique name for the new schedule.

Cron Expression key operators

* any value , list separator - range of values / step values

Minutes
*
Allowed values: 0-59

Hours
*
Allowed values: 0-23

Day of Month
*
Allowed values: 1-31

Month
*
Allowed values: 1-12
Alt. values JAN-DEC

Day of Week
*
Allowed values: 0-6
Alt. values SUN-SAT

Runs app: "Every minute" (UTC -4)
Derived from given values in fields above.

Custom Parameters (Optional)

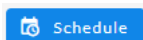
1	{ }
---	-----

Overrides default application variables for ad-hoc execution settings. Any application variable may be specified, and overridden per run here. Special properties like "queue" may also be set here.

Close Save Schedule

As you update the cron expression, the **Schedule** window displays the results of the expression in more readable language, such as *Runs app: "Every 0 and 30th minute past every hour on Sat"*, based on 0,30 in the **Minutes** field and 6 in the **Day of Week** field.

- Click **[Save Schedule]**. The schedule is added to the **Schedule List** pane. Also, on the **Applications** page, the Schedule button now displays with a dark blue background:



NOTE: After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

- On the **Applications** page, click the **[Schedule]** button for the application that contains a schedule you want to delete. The **Scheduler** window appears.

2. Click the down arrow icon (▼) to view the details of an existing schedule.
3. To delete the selected schedule, click the Actions icon (⋮) and select **[Delete]**.

TIP: On the **Scheduler** window for a Skylar Automation application, you can click the **[Copy as]** button from the **Schedule List** pane to make a copy of an existing schedule.

Known Issues

This section covers known issues with this release of the SyncPack.

Adding or Checking Hosting Dependent Relationships

Versions of the ServiceNow [CMDB CI Class Models](#) application older than version 1.30.0 do not include the following two relationships, which are required for correct mapping:

cmdb_ci_vcenter_datastore:

Parent Class	Relationship	Child Class
ESX Resource Pool	Hosted on::Hosts	Logical Datacenter

cmdb_ci_esx_resource_pool:

Parent Class	Relationship	Child Class
VMware vCenter Datastore	Hosted on::Hosts	Logical Datacenter

NOTE: These two relationships are included in version 1.30.0 and later of the "ServiceNow CMDB CI Class Models" application. You can download the application from the [ServiceNow Store](#).

In addition, not all classes are mapped. Only a limited set of classes are handled by default. Currently, if not specified, all classes from Skylar One will be sent to the physical hardware import table.

Known Issue with VMware Sync

Base ServiceNow Discovery of VMware is incompatible with the VMware mapping in this SyncPack. At the request of ServiceNow, Skylar Automation lines up VMware with the cloud management model for mapping, which is different if you do not have the cloud management plugin installed.

In the future, ServiceNow will update Base ServiceNow discovery for VMware to bring it in line with cloud management mapping.

NOTE: For more information about addressing specific issues, see the [Troubleshooting](#) chapter.

Additional Known Issues

The Physical Robust transform map will filter out devices that do not have virtual type of *Physical*:

Virtual Types
Physical
Component
Virtual
DCMR (Skylar Automation-generated)
Cloud service account (Skylar Automation-generated)

Physical Hardware and Hardware within ServiceNow do not line up exactly. There is some "best guess" handling when OIDs do not match to get the Device close to the correct class in ServiceNow.

Also, the OID list within ServiceNow and ScienceLogic do not always match up, which might cause some devices to either end up on the hardware table or a "best guess".

Chapter

4

Troubleshooting the SyncPack

Overview

This chapter includes troubleshooting resources and procedures to use with the *ServiceNow Service Graph Connector* SyncPack.

This chapter covers the following topics:

<i>Initial Troubleshooting Steps</i>	41
<i>Troubleshooting Specific to this SyncPack</i>	41
<i>Resources for Troubleshooting</i>	43

Initial Troubleshooting Steps

Skylar Automation acts as a middle server between data platforms. For this reason, the first steps should always be to ensure that there are no issues with the data platforms with which Skylar Automation is talking. There might be additional configurations or actions enabled on ServiceNow or Skylar One that result in unexpected behavior.

TIP: For detailed information about how to perform the steps below, see [Resources for Troubleshooting](#).

Skylar One Skylar Automation

1. Run `docker service ls` on the Skylar Automation server.
2. Note the Docker container version, and verify that the Docker services are running.
3. If a certain service is failing, make a note the service name and version.
4. If a certain service is failing, run `docker service ps <service_name>` to see the historical state of the service and make a note of this information. For example: `docker service ps iservices_contentapi`.
5. Make a note of any logs impacting the service by running `docker service logs <service_name>`. For example: `docker service logs iservices_couchbase`.

ServiceNow

1. Make a note of the ServiceNow version and SyncPack version, if applicable.
2. Make a note of whether the user is running a version of the Certified/Scoped application for a SyncPack, if relevant.
3. Make a note of the SyncPack application that is failing on Skylar Automation.
4. Make a note of what step is failing in the application, try running the application in debug mode, and capture any traceback or error messages that occur in the step log.

TIP: For more information about specific issues with relationships and mappings, see [Known Issues](#).

Troubleshooting Specific to this SyncPack

This section contains troubleshooting information specific to the ServiceNow Service Graph Connector SyncPack.

Enabling Verbose Logging

To enable verbose logging in ServiceNow:

1. Navigate to System Import Sets > Administration > Robust Import Set Transformers (sys_robust_import_set_transformer).
2. Note that the endpoints used by the ScienceLogic for Service Graph are all prefixed with *SG-ScienceLogic*:
 - SG-ScienceLogic Physical Device Source
 - SG-ScienceLogic VMware Device Source
3. Select each of the endpoints and check the **Verbose** box on the **Robust Import Set Transformer** page to turn on logging within ServiceNow.
4. When you are done troubleshooting, de-select the **Verbose** box for each endpoint.

Using the Results of Verbose Logging

The import table contains a Reference link to the import set associated with the record being imported:

Import Tables	
SG-ScienceLogic Physical Device Source	x_sclo_devicesync_device_import
SG-ScienceLogic VMware Device Source	x_sclo_devicesync_device_import_vmware

When you select the import set associated with the record in question, you can view a related list of Import Set Runs. Each run of a specific import set associated will have its own set of import Set Row Errors and Import Logs that you can review.

Narrowing the Import Set

When you run the "Sync Devices from Skylar One to ServiceNow via SGC" Skylar Automation application, the default payload size sent to ServiceNow is 5000 objects.

If you are having issues, ScienceLogic recommends that you run this application in Debug Mode and copy the JSON of what is being sent to ServiceNow.

Also, you can run the following query to find the device object and any associated virtual type "dcmr" objects, which are parent relationships that should be listed directly after the device in question:

```
{
  "devices": [
    {
      "virtual_type": "PowerFlow",
    },
    {
```

```
    "virtual_type": "dcmr",  
  }  
]  
}
```

Sending this limited version of the payload will make looking at the verbose logging much easier. The endpoints are listed below.

Import Table Endpoints

```
/api/now/import/x_sclo_devicesync_device_import/insertMultiple
```

```
/api/now/import/x_sclo_devicesync_device_import_vmware/insertMultiple
```

Resources for Troubleshooting

This section contains port information for Skylar Automation and troubleshooting commands for Docker, Couchbase, and the Skylar Automation API.

Useful Skylar Automation Ports

- **https://<IP of Skylar Automation>:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://<IP of Skylar Automation>:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by Skylar Automation workers.
- **https://<IP of Skylar Automation>/flower/workers**. Provides access to Flower, a tool for monitoring and administering Celery clusters.
- **https://<IP of Skylar Automation>:3141**. Provides access to the pypiserver service, which you can use to see if SyncPacks have been correctly uploaded to Devpi container.

IMPORTANT: Port 5556 must be open for both Skylar Automation and the client.

Helpful Docker Commands

Skylar Automation is a set of services that are containerized using Docker. For more information about Docker, see the [Docker tutorial](#).

Use the following Docker commands for troubleshooting and diagnosing issues with Skylar Automation:

Viewing Container Versions and Status

To view the Skylar Automation version, SSH to your instance and run the following command:

```
rpm -qa | grep powerflow
```

To view the individual services with their respective image versions, SSH to your Skylar Automation instance and run the following command:

```
docker service ls
```

In the results, you can see the container ID, name, mode, status (see the *replicas* column), and version (see the *image* column) for all the services that make up Skylar Automation:

```
root@fsunias4lab ~# docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
ommihu35v30i	services_gui	replicated	1/1	repository.auto.sciencelogic.local:5000/is-gui:1.7.0	*:80->80/tcp, *:443->443/tcp
40ve91ciwnh3	services_redis	replicated	1/1	redis:4.0.2	
plm6hljcumif	services_flower	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	*:5555->5555/tcp
hh3pt2181zsf	services_scheduler	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	
htmlcvp6mxx	services_contentapi	replicated	1/1	repository.auto.sciencelogic.local:5000/is-api:1.7.0	*:5000->5000/tcp
eyin5qgudmi	services_rabbitmq	replicated	1/1	rabbitmq:3	
klui9h8jzfs6	services_visual	replicated	2/1	dockerexamples/visualizer:latest	*:8081->8080/tcp
yoy38w8buauw	services_couchbase	replicated	1/1	repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0	*:8091->8091/tcp, *:8092->8092/tcp
3->8093/tcp, *:8094->8094/tcp, *:11210->11210/tcp					
zlbwtmox7uf	services_steprunner	replicated	5/5	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	

Restarting a Service

Run the following command to restart a single service:

```
docker service update --force <service_name>
```

Stopping all Skylar Automation Services

Run the following command to stop all Skylar Automation services:

```
docker stack rm iservices
```

Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

NOTE: Restarting Docker does not clear the queue.

Diagnosis Tools

Multiple diagnosis tools exist to assist in troubleshooting issues with the Skylar Automation platform:

- **Docker PowerPack.** This PowerPack monitors your Linux-based Skylar Automation server with SSH (the Skylar Automation ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your Skylar Automation server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor Skylar Automation, see the [Using Skylar One to Monitor Skylar Automation](#).

- **Flower.** This web interface tool can be found at the /flower endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the Skylar Automation platform. This tool also shows detailed information about each of the tasks that have been executed on the platform. This data includes the UUID, the state, the arguments that were passed to it, as well as the worker and the time of execution. Flower also provides a performance chart that shows the number of tasks running on each individual worker.
- **Debug Mode.** All applications can be run in "debug" mode via the Skylar Automation API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see [Retrieving Additional Debug Information](#).
- **Application Logs.** All applications generate a log file specific to that application. These log files can be found at `/var/log/services` and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.
- **Step Logs.** Step logs display the log output for a specific step in the application. These step logs can be accessed via the Skylar Automation user interface by clicking on a step in an application and bringing up the **Step Log** tab. These step logs display just the log output for the latest run of that step.
- **Service Logs.** Each Docker service has its own log. These can be accessed via SSH by running the following command:

```
docker service logs -f <service_name>
```

Retrieving Additional Debug Information (Debug Mode)

The logs in Skylar Automation use the following **loglevel** settings, from most verbose to least verbose:

- **10.** Debug Mode.
- **20.** Informational.
- **30.** Warning. This is the default settings if you do not specify a loglevel.
- **40.** Error.

WARNING: If you run applications in Debug Mode ("loglevel": 10), those applications will take longer to run because of increased I/O requirements. Enabling debug logging using the following process is the only recommended method. ScienceLogic does not recommend setting "loglevel": 10 for the whole stack with the **docker-compose** file.

To run an application in Debug Mode using the Skylar Automation user interface:

1. Select the Skylar Automation application from the **Applications** page.
2. Hover over the **[Run]** button and select *Custom Run* from the pop-up menu. The **Custom Run** window appears.
3. Select the Logging Level. *Debug* is the most verbose and will take longer to run.

- Specify the configuration object for the custom run in the **Configuration** field, and add any JSON parameters in the **Custom Parameters** field, if needed.
- Click **[Run]**.

To run an application in Debug Mode using the API:

- POST the following to the API endpoint:

```
https://<Skylar Automation_IP>/api/v1/applications/run
```

- Include the following in the request body:

```
{
  "name": "<application_name>",
  "params": {
    "loglevel": 10
  }
}
```

After running the application in Debug Mode, review the step logs in the Skylar Automation user interface to see detailed debug output for each step in the application. This information is especially helpful when trying to understand why an application or step failed:

The screenshot displays the Skylar Automation user interface. At the top, there's a navigation bar with 'Application', 'Help', 'Notifications', 'Isadmin', and the ScienceLogic logo. Below this, a toolbar contains 'Open Editor', 'Reports', 'Timeline', 'Replay', 'Run', and 'Configure'. The main area shows a workflow diagram with four steps: 'Pull Disabled Devices from VCUG in SL1' (highlighted with a red box), 'Pull Affected Device Info from SL1 (MySQL)', 'Verify Device Delete Requests', and 'Delete Devices'. Below the diagram is a 'STEP LOG' section with a table of logs. The table has columns for 'Module', 'Date (UTC-4)', 'Log Level', and 'Message'. The log shows an error in the 'BaseStep' module at 10:48:21,357, with a message indicating a traceback in the 'PullAndProcessDisabledDevices.py' file.

Module	Date (UTC-4)	Log Level	Message
BaseStep	Aug 29, 2023 10:48:21,347	INFO	Executing: Pull Disabled Devices from VCUG in SL1 - steps/PullAndProcessDisabledDevices.py
ipaas_logger	Aug 29, 2023 10:48:21,348	FLOW	Start Pull Disabled Devices from VCUG in SL1
ipaas_logger	Aug 29, 2023 10:48:21,355	FLOW	Step Pull Disabled Devices from VCUG in SL1 still failed after 0 retries
BaseStep	Aug 29, 2023 10:48:21,357	ERROR	Traceback (most recent call last): File "/usr/local/lib/python3.8/site-packages/ipaascore/BaseStep...", line 601, in execute_step self.retry_step(task=task, exc=err, File "/usr/local/lib/python3.8/site-packages/ipaascore/BaseStep.py", line 961, in retry_step task.retry() File "/usr/local/lib/python3.8/site-packages/celery/app/task.py", line 706, in retry raise_with_context(exc) File "/usr/local/lib/python3.8/site-packages/ipaascore/BaseStep.py", line 579, in execute_step self.execute() File "/usr/local/lib/python3.8/site-packages/ipaascore/BaseStep.py", line 51, in inner_execute execute = method(self) File "/var/syncpacks/virtualenvs/service-now_cmdb_syncpack/lib/python3.8/site-packages/service-now_cmdb_syncpack/steps/PullAndProcessDisabledDevices.py", line 69, in execute raise MissingRequiredStepParameter('ipaascommon.ipaas.exceptions.MissingRequiredStepParameter: target_vcue is required but is not populated in either delete_devices or in SyncService...')

You can also run an application in debug using curl via SSH:

1. SSH to the Skylar Automation instance.
2. Run the following command:

```
curl -v -k -u isadmin:<password> -X POST "https://<your_
hostname>/api/v1/applications/run"
-H 'Content-Type: application/json' -H 'cache-control: no-cache' -d
'{"name":
"interface_sync_sciencelogic_to_servicenow","params": {"loglevel":
10}}'
```

Appendix

A

ServiceNow API Endpoints for the Service Graph Connector

Overview

This appendix describes the ServiceNow API Endpoints that were created for the ServiceNow Service Graph Connector.

This chapter covers the following topics:

<i>Staging Tables</i>	49
-----------------------------	----

Staging Tables

The integration uses the ImportSet API at https://developer.servicenow.com/dev.do#!/reference/api/rome/rest/c_importSetAPI. The only interaction between ServiceNow and ScienceLogic systems is to POST to two specific staging tables:

- SG-ScienceLogic Physical Devices
- SG-ScienceLogic VMware Devices

Staging Table Names

<i>Name</i>	<i>Table</i>
SG-ScienceLogic Physical Devices	x_sclo_devicesync_device_import
SG-ScienceLogic VMware Devices	x_sclo_devicesync_device_import_vmware

URL Format

<i>Name</i>	<i>Value</i>
SG-ScienceLogic Physical Devices	/api/now/import/x_sclo_devicesync_device_import/insertMultiple
SG-ScienceLogic VMware Devices	/api/now/import/x_sclo_devicesync_device_import_vmware/insertMultiple

Example Payload

```
{
  "devices": [
    {
      "sll_id": 45,
      "region": "unique_id",
      "native_key": "unique_id+DEV",
      "org_name": "System",
      "org_id": "0",
      "org_crm_id": "",
      "asset_id": 45,
      "serial": "VMware-42 05 48 4f 66 96 d7 dc-f1 cf 4a c2 53 6d ad 10",
      "make": "VMware, Inc.",
      "model": "VMware Virtual Platform",
      "vital_asset_information": "SNMP Location Data: Unknown",
    }
  ]
}
```

```

    "memory": 4045504,
    "cpu_make": "Xeon(R)",
    "cpu_speed": 2199.998,
    "firmware_version": 6.0,
    "device_category": "System.EM7",
    "device_base_class": "ScienceLogic, Inc.",
    "device_sub_class": "EM7 All-In-One",
    "virtual_type": "physical",
    "name": "em7-ao-118",
    "ip": "10.2.11.118",
    "date_added": "2021-06-17 19:13:31",
    "sll_url": "https://10.2.11.154/em7/index.em7?exec=device_sum-
mary&did=45",
    "oid": "1.3.6.1.4.1.19567.2.1.10"
  }
]
}

```

Example Response

```

{
  "import_set_id": "6dff2f26dbaef490d91fae5a489619a2",
  "multi_import_set_id": "e5ff2f26dbaef490d91fae5a489619a4"
}

```

© 2003 - 2025, 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