ScienceLogic

ServiceNow Service Graph Connector Synchronization PowerPack

Version 1.0.0, rev 1

Table of Contents

Chapter

1

Introduction to the ServiceNow Service Graph Connector Synchronization PowerPack

Overview

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

You can use this Synchronization PowerPack to sync your monitored devices in SL1 to your ServiceNow CMDB using the ServiceNow Service Graph Connector. Instead of adding classes and field mappings in PowerFlow, you can set up those classes and field mappings in ServiceNow.

> WARNING: This Synchronization PowerPack is not compatible with versions of the *ServiceNow Incident* Synchronization PowerPack older than version 4.0.0.

> NOTE: You cannot upgrade to this Synchronization PowerPack from the *ServiceNow CMDB* Synchronization PowerPack.

> NOTE: The label "SyncPack" is used in place of "Synchronization PowerPack" in the PowerFlow user interface.

This chapter covers the following topics:

Synchronization PowerPack Architecture and Workflow

The following diagram details the workflow and architecture for the bi-directional sync of SL1 Devices with ServiceNow Configuration Items (CIs):
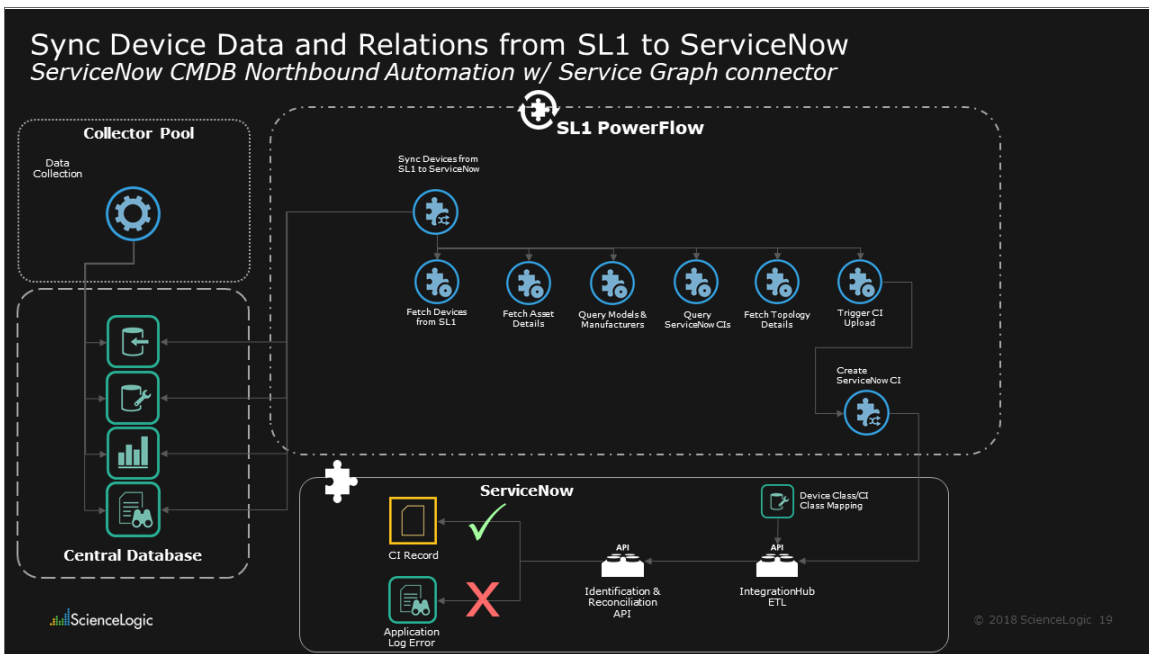
SL1 and ServiceNow Terminology

The following table lists the different names for the shared elements in SL1 and ServiceNow:

SL1	ServiceNow
Asset, Custom Attribute	Asset (ITAM)
Device	CI (Configuration Item)
Discovery Session	Service Request, Catalog Request
Event	Incident, Event, or Case (depending on the Synchronization PowerPack 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

Dependencies for this Synchronization PowerPack

The *Service Graph Connector* Synchronization PowerPack requires that you install the following Synchronization PowerPacks on your PowerFlow system:

- *Base Steps* Synchronization PowerPack 1.3.1 or later
- *ServiceNow Base* Synchronization PowerPack 3.3.0 or later

> **TIP:** For more information about the *Base Steps* Synchronization PowerPack, see the **SL1 PowerFlow Platform** manual. You can download Synchronization PowerPacks from the [PowerPacks page](#) at the SL1 Support Site.

The *Service Graph Connector* Synchronization PowerPack also requires version 1.30 or later of *CMDB CI Class Models* in ServiceNow. You can download the models from the [ServiceNow Store](#).

Prerequisites for ServiceNow Synchronization PowerPacks

This section describes the prerequisites for the ServiceNow Synchronization PowerPacks. For more information about the specific software versions required by a ServiceNow Synchronization PowerPack, see the release notes for that Synchronization PowerPack.

To install any of the ScienceLogic ServiceNow Synchronization PowerPacks, you must have administrator access to both SL1 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 Synchronization PowerPacks at the same time, you should upload *all* of the Synchronization PowerPacks first, and then install them to address any dependencies between the Synchronization PowerPacks.

The following table lists the port access required by PowerFlow and this Synchronization PowerPack:

Source IP	PowerFlow Destination	PowerFlow Source Port	Destination Port	Requirement
PowerFlow	SL1 API	Any	TCP 443	SL1 API Access
PowerFlow	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.

Contents of the Synchronization PowerPack

This section lists the contents of the *ServiceNow Service Graph Connector* Synchronization PowerPack.

PowerFlow Applications

The following PowerFlow applications are included in this Synchronization PowerPack:

- ***Cache SL1 Device Classes***. Reads all existing SL1 Device Classes and writes them to a cache. You must run this application before you run the "Sync Devices from SL1 to ServiceNow via SGC" application.
- ***Post to SGC***. Posts the payload to ServiceNow. This is an internal application, which mean you will not run this directly. This application gets triggered by the "Sync Devices from SL1 to ServiceNow via SGC" application.
- ***Sync Devices from SL1 to ServiceNow via SGC***. Syncs devices and their organizations and relationships from SL1 to ServiceNow. For more information, see *Syncing Devices from SL1 to ServiceNow*.

Steps

The following steps are included in this Synchronization PowerPack:

- FetchParentDevices

- ProcessSL1Devices

- Pull and Process DCM+R Relationships for SGC

- PullAndProcessSL1Orgs

- PullDevicesFromSL1

- Query and Cache SL1 Device Class Info

- SL1SnowDeviceCompare

> **TIP:** You can use the "ServiceNow SyncPack" configuration object from the *ServiceNow Base*Synchronization PowerPack as a template for a configuration object for this Synchronization PowerPack.

> **NOTE:** This Synchronization PowerPack 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*.

Installing the Synchronization PowerPack

Overview

This chapter describes how to install the *ServiceNow Service Graph Connector* in PowerFlow. This chapter also covers how to install the "Service Graph Connector for ScienceLogic" Application in ServiceNow and the other dependencies required by this Synchronization PowerPack.

This chapter covers the following topics:

Downloading and Installing the Synchronization PowerPack

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

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

To locate and download the Synchronization PowerPack:

1. Go to the [ScienceLogic Support Site](#).
2. Click the **[Product Downloads]** tab and select *PowerPack*.
3. In the **Search PowerPacks** field, search for the Synchronization PowerPack and select it from the search results. The **Release Version** page appears.
4. On the **[PowerPack Versions]** tab, click the name of the Synchronization PowerPack version that you want to install. The **Release File Details** page appears.
5. Click the **[Download File]** button or click the name of the **.zip** file containing the **.whl** file for this Synchronization PowerPack to start downloading the file.

> **NOTE:** Synchronization PowerPacks do not require a specific license. After you download a Synchronization PowerPack, you can import it to your PowerFlow system using the PowerFlow user interface.

> **NOTE:** If you are installing or upgrading to the latest version of this Synchronization PowerPack in an offline deployment, see "Installing or Upgrading in an Offline Environment" in the Synchronization PowerPack release notes to ensure you install any external dependencies.

Importing the Synchronization PowerPack

To import a Synchronization PowerPack in the PowerFlow user interface:

1. On the **SyncPacks** page of the PowerFlow user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.
2. Click **[Browse]** and select the **.whl** file for the Synchronization PowerPack you want to install.

> **TIP:** You can also drag and drop a **.whl** file to the **Import SyncPack** page.

3. Click **[Import]**. PowerFlow registers and uploads the Synchronization PowerPack. The Synchronization PowerPack is added to the **SyncPacks** page.

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


Installing the Synchronization PowerPack

NOTE: You must import and install the *ServiceNow Base* Synchronization PowerPack before uploading and installing any of the other ServiceNow Synchronization PowerPacks.

To activate and install a Synchronization PowerPack in the PowerFlow user interface:

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

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

TIP: If you do not see the PowerPack 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 Synchronization PowerPack is activated, the **SyncPacks** page displays a green check mark icon (✅) for that Synchronization PowerPack. 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 check mark icon (✅) or the exclamation mark icon (❗) in the **Activated** column for that Synchronization PowerPack. For a successful installation, the "Activate & Install SyncPack" application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

4. If you have other versions of the same Synchronization PowerPack on your PowerFlow system, you can click the **[Actions]** button (⋮) for that Synchronization PowerPack 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 Synchronization PowerPack. 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 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* Synchronization PowerPack.

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 Synchronization PowerPack

Overview

This chapter describes how to configure Device Sync between SL1 and ServiceNow using the ServiceNow Service Graph Connector.

This chapter covers the following topics:

Workflow for Configuring the Integration with the ServiceNow Service Graph Connector

The following workflows describe how to configure ServiceNow and PowerFlow to work with the Synchronization PowerPack.

Configuring the ServiceNow Instance

1. *Configure the Service Graph Connector for ScienceLogic.*
2. *Create a ServiceNow Group.*
3. *Create a ServiceNow User*.

Configuring the PowerFlow System

1. *Install the ServiceNow Service Graph Connector Synchronization PowerPack and its dependencies.*
2. *Create a configuration object in the PowerFlow user interface.*
3. *Align the new configuration object with the following PowerFlow applications*:

   - "Cache SL1 Device Classes"
   - "Sync Devices from SL1 to ServiceNow via SGC"
   - "Post to SGC"

4. *Run* or *schedule* the following applications, in the following order:

   - "Cache SL1 Device Classes"
   - "Sync Devices from SL1 to ServiceNow via SGC"

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 Connector for 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]**.
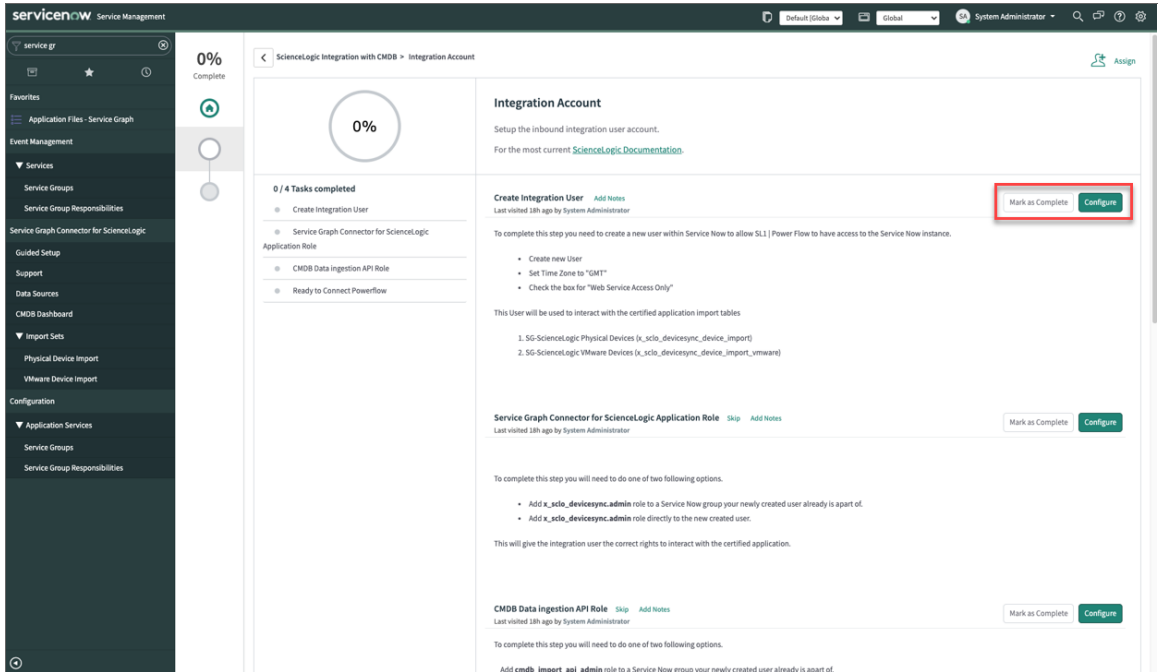
5. After you finish all the tasks in the Integration Account section, go to the **PowerFlow Setup** section and click **[Get Started]**:
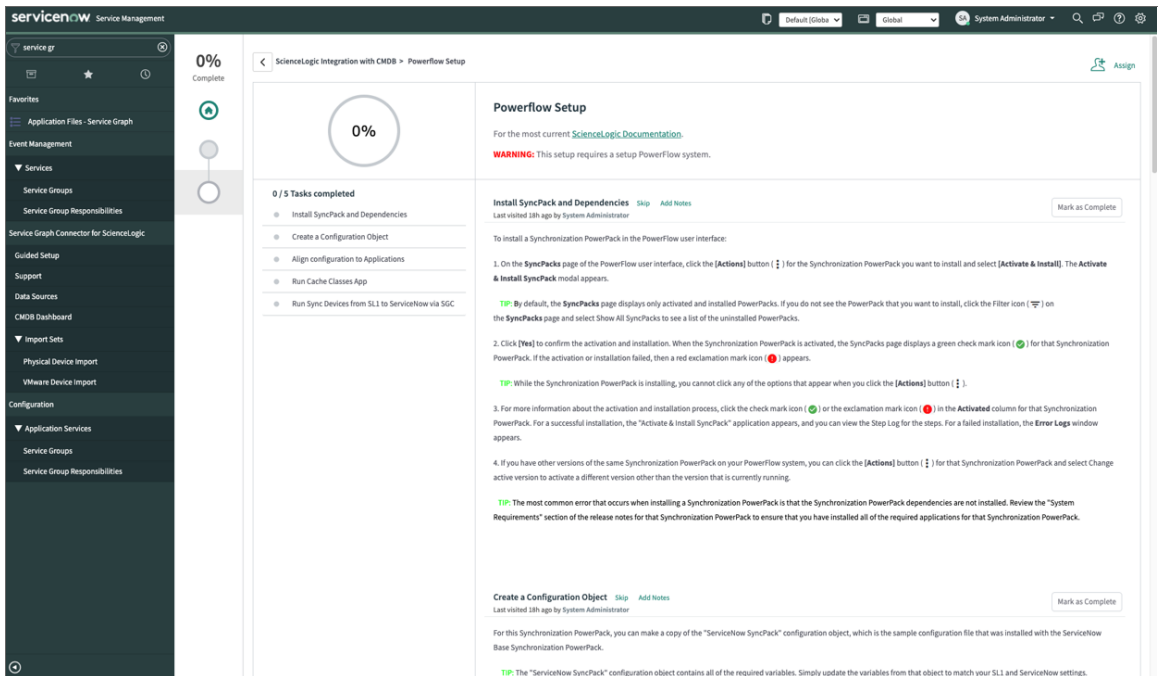
6. These PowerFlow tasks are described in the following sections of this manual:

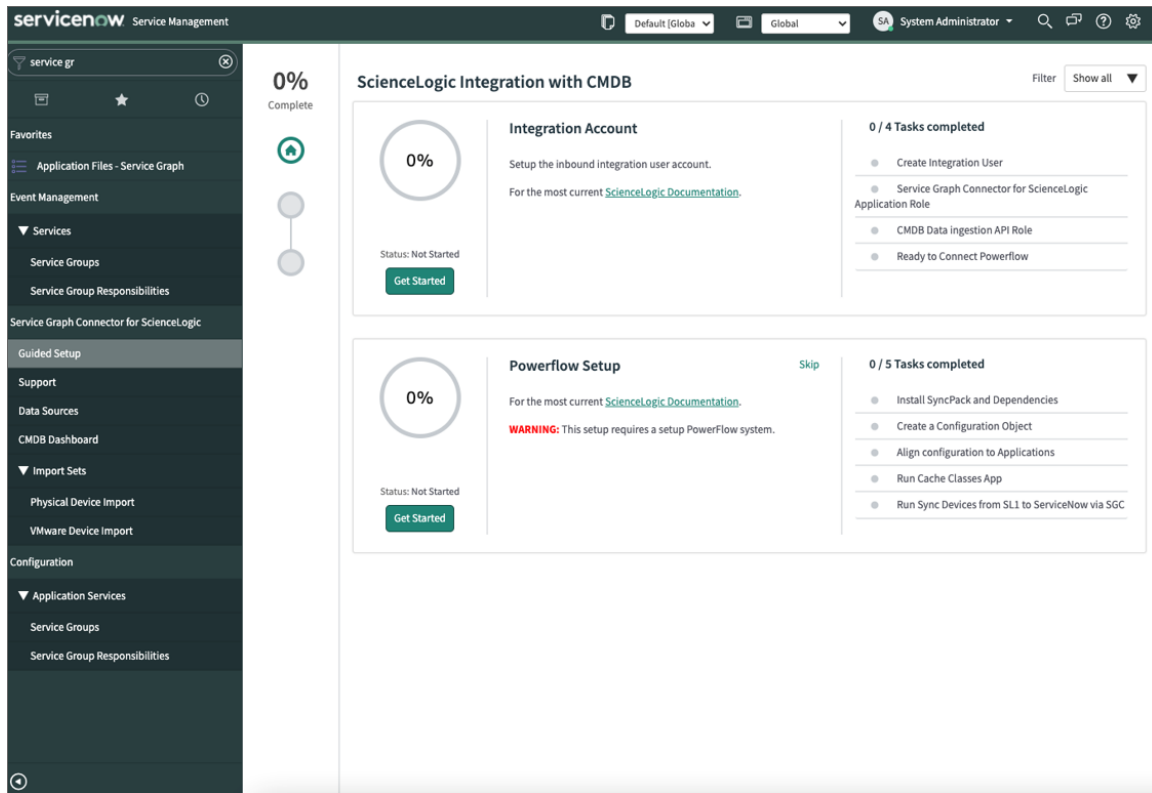- *Install the ServiceNow Service Graph Connector Synchronization PowerPack and its dependencies.*
- *Create a configuration object in the PowerFlow user interface.*
- *Align the new configuration file with the PowerFlow applications for this Synchronization PowerPack.*
- *Run the "Cache SL1 Device Classes" application.*
- *Run the "Sync Devices from SL1 to ServiceNow via SGC" application.*

7. After you complete each task in PowerFlow, click **[Mark as Complete]** in the **PowerFlow Setup** section in ServiceNow.

8. When are done with all the tasks in the **PowerFlow 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:

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

5. At the bottom of the Group form, select the **[Roles]** tab and click **[Edit:]**
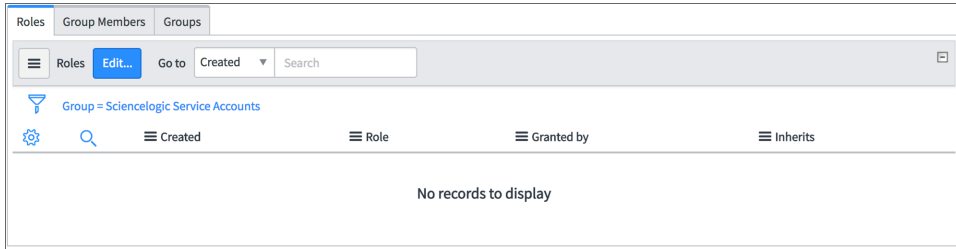
6. Search for *cmdb_import_api_admin* and *x_sclo_devicesync.admin* and move them to the **Roles List** column using the arrow buttons:

Creating a ServiceNow Group
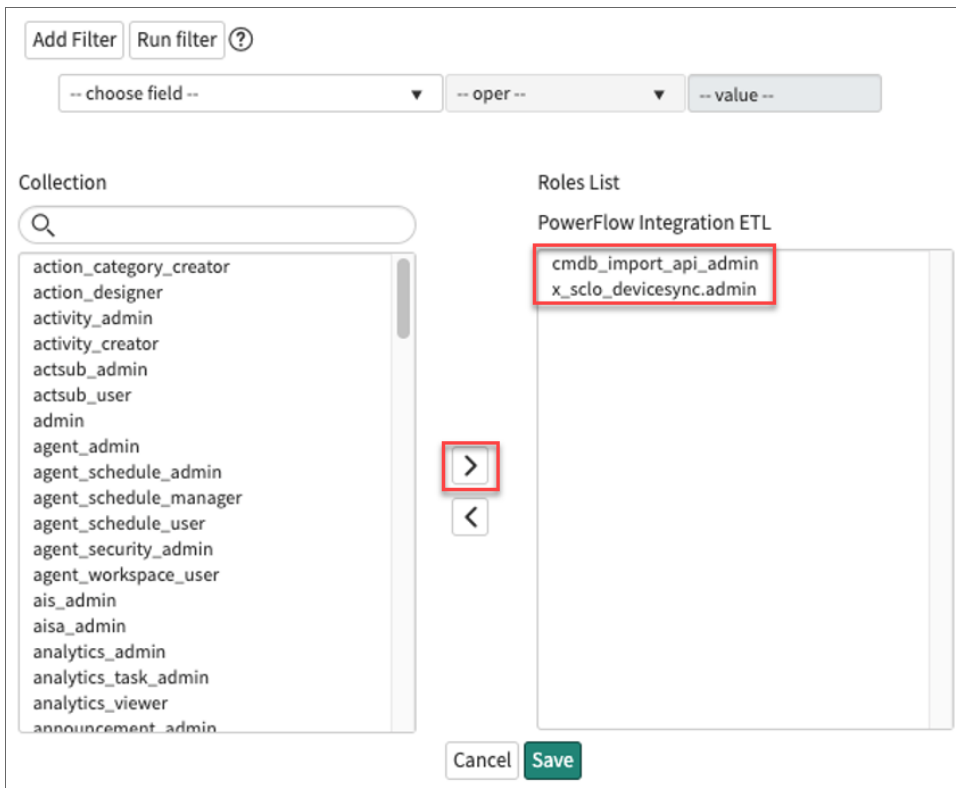
7. Click **[Save]**. Your ServiceNow Group now has two assigned Roles:
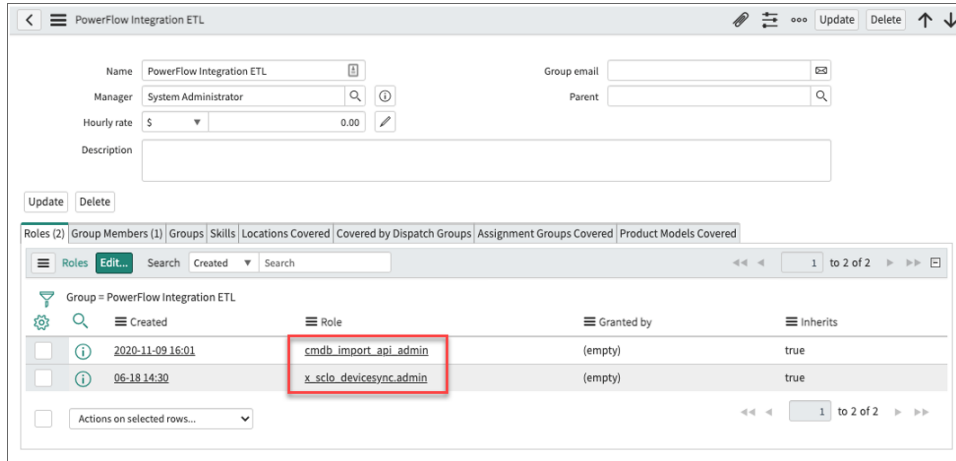
8. Next, create a ServiceNow user to use with this Group. See the following procedure for the details.

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 PowerFlow 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:
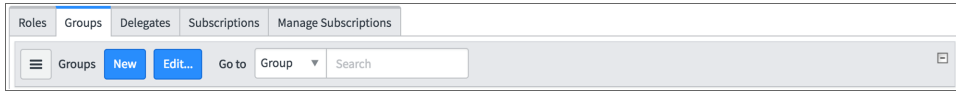
6. Find the group you created previously and move the group to the right-hand column using the arrow buttons:
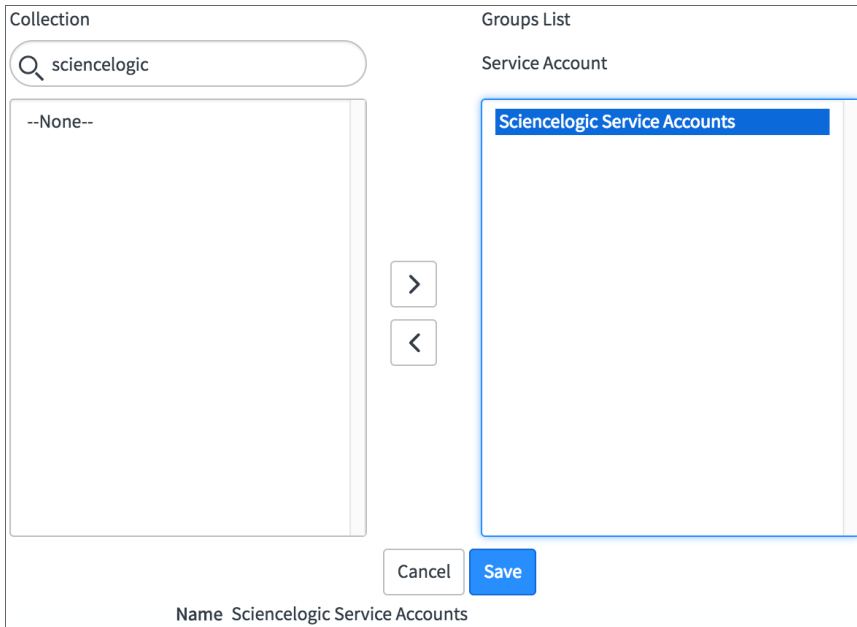
7. Click **[Save]**. After the user has been added to the group, you can see their Roles and Groups at bottom of the record:

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

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 PowerFlow. The **Configurations** page (⚙) of the PowerFlow user interface lists all available configuration objects for that system.

Creating and Aligning a Configuration Object

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

To use this Synchronization PowerPack, you will need to create one or more configuration objects in the PowerFlow user interface and align that configuration object to the applications that let you sync data between SL1 and ServiceNow.

> **NOTE**: Depending on your SL1 and ServiceNow environments, you might be able to use the same configuration object with other ServiceNow Synchronization PowerPacks.

> **NOTE**: When either multiple SL1 stacks or multiple ServiceNow systems are involved with PowerFlow, you should create an individual configuration object for each SL1 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 SL1 stack or ServiceNow system. Creating copies of a PowerFlow application from a Synchronization PowerPack for the purpose of distinguishing between domains is not supported, and will result in issues on upgrades.

Creating a Configuration Object

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

> **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 SL1 and ServiceNow settings.

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

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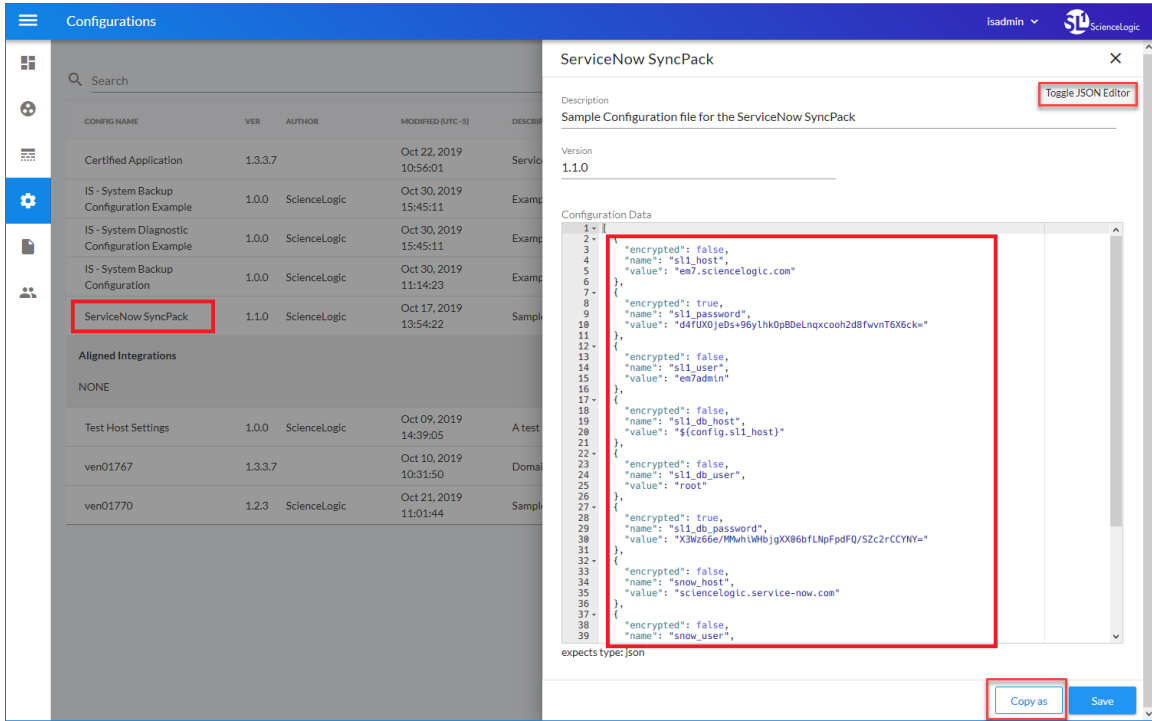
> **TIP:** Click **[Toggle JSON Editor]** to show the JSON code. Click the button again to see the fields.

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

> **TIP:** This step is required. Do *not* use the original configuration object to run PowerFlow 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.

Creating and Aligning a 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 SL1 with an External Database (SL1 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 SL1 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 PowerFlow configuration.

> **NOTE:** The *region* value is a user-defined variable that identifies your SL1 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, PowerFlow 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 PowerFlow 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. 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 Synchronization PowerPack, you must first "align" a configuration object with those applications. The applications you need to align include the following:

- *Cache SL1 Device Classes*. Reads all existing SL1 Device Classes and writes them to a cache. You must run this application before you run the "Sync Devices from SL1 to ServiceNow via SGC" application.
- *Sync Devices from SL1 to ServiceNow via SGC*. Syncs devices and their organizations and relationships from SL1 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 PowerFlow user interface, open the relevant application and click [Configure] (⚙). The **Configurations** pane for that application appears:

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 PowerFlow applications in this Synchronization PowerPack.

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.

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 SL1 to ServiceNow

To perform a Device Sync between SL1 and ServiceNow, run the following applications in the PowerFlow user interface:

- *Cache SL1 Device Classes*. Reads all existing SL1 Device Classes and writes them to a cache.
- *Sync Devices from SL1 to ServiceNow via SGC*. Syncs devices and their organizations and relationships from SL1 to ServiceNow.

> NOTE: Currently, all data is being synced from SL1 to ServiceNow only.

To sync SL1 devices with ServiceNow:

1. In the PowerFlow user interface, select the "Cache SL1 Device Classes" application from the **Applications** page, click **[Configure]** (⚙), align a configuration object, and then click **[Run]** (▶).

2. Select the "Sync Devices from SL1 to ServiceNow via SGC" application from the **Applications** page and click **[Configure]** (⚙). The **Configuration** pane appears:

3. Complete the following fields, as needed:

- **Configuration**. Select the configuration object with the relevant SL1 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 *SL1 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 20 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 SL1 Organizations in the device sync, add the Organization IDs from SL1 in this field, separated by commas. Leave this field empty to sync all SL1 Organizations.
- **Include_CUGs**. If you want to include SL1 Collector Groups (CUGs) in the device sync, add the Collector Group IDs from SL1 in this field, separated by commas. Leave this field empty to sync all SL1 Collector Groups.
- **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 SL1 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 SL1 side.

- **sl1_chunk_size**. Specify the number of devices to include in each chunk sent to in the SL1 GraphQL query when you run this application. The default chunk size is 100 devices.
- **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 SL1 by typing the URL or IP address for SL1 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"}}`
- **sl1_url_override**. Update this field if you want to use an URL that is different from the standard SL1 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 SL1 for only the devices that you want to sync. Leave this field empty to sync all SL1 devices.
- **lookup_chunk_size**. Specify the number of devices in each chunk when pulling the device cache from Couchbase. The default is 10 devices.

- **snow_chunk_size**. Specify the number of devices per chunk to post to ServiceNow when you run this application. The default chunk size is 5000 devices.

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

- **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.
- **retry_max**. The maximum number of times PowerFlow 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_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, PowerFlow will incrementally increase the interval between retries. The default is unselected.
- **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.

4. The **sgc_uri_categories** and the **sgc_category_overrides** sections lets you send specific SL1 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 SL1 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.

Known Issues

This section covers known issues with this release of the Synchronization PowerPack.

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 SL1 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 Synchronization PowerPack. At the request of ServiceNow, PowerFlow 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 (PowerFlow-generated)

Virtual Types
Cloud service account (PowerFlow-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".

Scheduling PowerFlow Applications

You can create one or more schedules for a single application in the PowerFlow 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 **Schedule** window appears, displaying any existing schedules for that application:
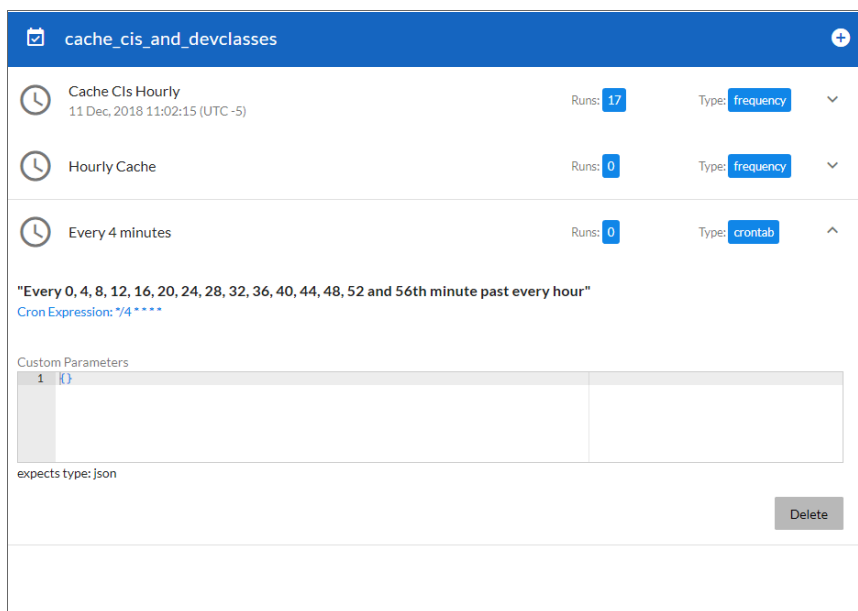
NOTE: If you set up a schedule using a cron expression, the details of that schedule display in a more readable format in this list. For example, if you set up a cron expression of **/4 * * * ***, the schedule on this window includes the cron expression along with an explanation of that expression: "*Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, and 56th minute past every hour*".

2. Select a schedule from the list to view the details for that schedule.

3. Click the + icon to create a schedule. A blank **Schedule** window appears:

4. In the **Schedule** window, complete the following fields:

   - *Schedule Name*. Type a name for the schedule.

   - *Switch to*. Use this toggle to switch between a cron expression and setting the frequency in seconds.

     ○ *Cron expression*. Select this option to schedule the application using a cron expression. 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. As you update the cron expression, the **Schedule** window displays the results of the expression in more readable language, such as *Expression: "Every 0 and 30th minute past every hour on the 1 and 31st of every month"*, based on ***/30 * * /30 * ***.

     ○ *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.

5. Click **[Save Schedule]**. The schedule is added to the list of schedules on the initial **Schedule** window. Also, on the **Applications** page, the word "Scheduled" appears in the **Scheduled** column for this application, and the **[Schedule]** button contains a check mark:
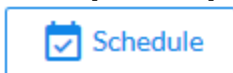
> **NOTE**: After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

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

2. Click the down arrow icon (⌄) to view the details of an existing schedule:
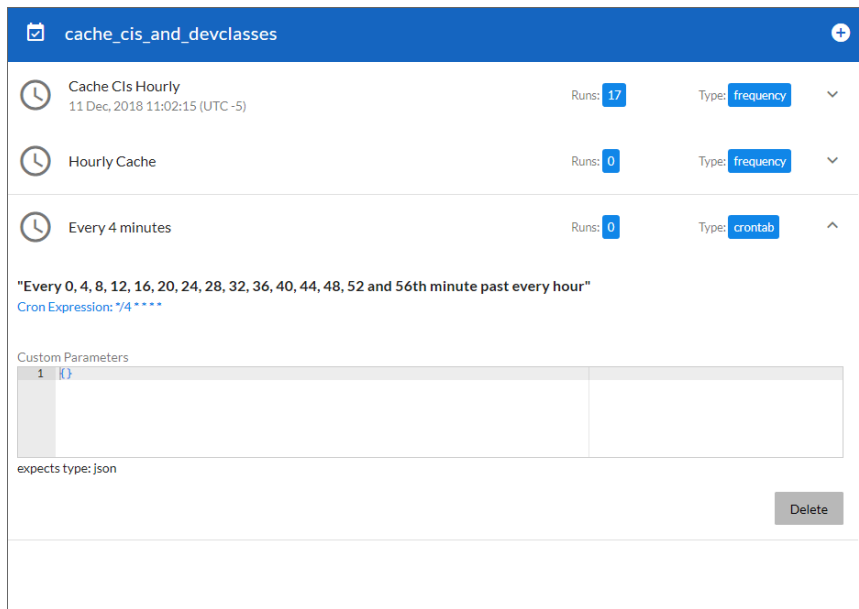
3. To delete the selected schedule, click **[Delete]**. The schedule is removed.

> **NOTE**: When either multiple SL1 stacks or multiple ServiceNow systems are involved with PowerFlow, you should create an individual configuration object for each SL1 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 SL1 stack or ServiceNow system. Creating copies of a PowerFlow application from a Synchronization PowerPack for the purpose of distinguishing between domains is not supported, and will result in issues on upgrades.

The following image shows how you can schedule PowerFlow applications for multiple ServiceNow domains:

device_sync_sciencelogic_to_servicenow

Device Sync Domain A
n/a (UTC -4)

Runs: 63   Type: crontab

"02:00 every day"
Cron Expression: 0 2 * * *

Custom Parameters

```
1  {
2      "configuration": "now_domain_a"
3  }
```

expects type: json

Delete

Device Sync Domain B
n/a (UTC -4)

Runs: 63   Type: crontab

Device Sync Domain C
May 19, 2021 00:00:00 (UTC -4)

Runs: 63   Type: crontab

Chapter

4

Troubleshooting the Synchronization PowerPack

Overview

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

This chapter covers the following topics:

Initial Troubleshooting Steps

PowerFlow 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 PowerFlow is talking. There might be additional configurations or actions enabled on ServiceNow or SL1 that result in unexpected behavior.

> **TIP:** For detailed information about how to perform the steps below, see *Resources for Troubleshooting*.

SL1 PowerFlow

1. Run `docker service ls` on the PowerFlow 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 Synchronization PowerPack version, if applicable.
2. Make a note of whether the user is running a version of the Certified/Scoped application for a Synchronization PowerPack, if relevant.
3. Make a note of the Synchronization PowerPack application that is failing on PowerFlow.
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 Synchronization PowerPack

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

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 SL1 to ServiceNow via SGC" PowerFlow 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 PowerFlow and troubleshooting commands for Docker, Couchbase, and the PowerFlow API.

Useful PowerFlow Ports

- **https://*<IP of PowerFlow>*:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://*<IP of PowerFlow>*:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by PowerFlow workers. Use *guest/guest* for the login.
- **https://*<IP of PowerFlow>*/flower**. Provides access to Flower, a tool for monitoring and administrating Celery clusters.

NOTE: For version 2.0.0 and later of PowerFlow, port 5556 must be open for both PowerFlow and the client.

Helpful Docker Commands

PowerFlow 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 PowerFlow:

Viewing Container Versions and Status

To view the PowerFlow version, SSH to your PowerFlow 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 PowerFlow:

Restarting a Service

Run the following command to restart a single service:

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

Stopping all PowerFlow Services

Run the following command to stop all PowerFlow 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.

Viewing Logs for a Specific Service

You can use the Docker command line to view the logs of any current running service in the PowerFlow cluster. To view the logs of any service, run the following command:

```
docker service logs -f iservices_<service_name>
```
Some common examples include the following:

```
docker service logs -f iservices_couchbase
```

```
docker service logs -f iservices_steprunner
```

```
docker service logs -f iservices_contentapi
```

> **NOTE:** *Application* logs are stored on the central database as well as on all of the Docker hosts in a clustered environment. These logs are stored at **/var/log/iservices** for both single-node or clustered environments. However, the logs on each Docker host only relate to the services running on that host. For this reason, using the Docker service logs is the best way to get logs from all hosts at once.

Clearing RabbitMQ Volume

RabbitMQ is a service that distributes tasks to be executed by PowerFlow workers. This section covers how to handle potential issues with RabbitMQ.

The following error message might appear if you try to run a PowerFlow application via the API:

```
Internal error occurred: Traceback (most recent call last):\n File \"./content_
api.py\", line 199, in kickoff_application\n task_status = ... line 623, in _on_
close\n (class_id, method_id), ConnectionError)\nInternalError: Connection.open:
(541) INTERNAL_ERROR - access to vhost '/' refused for user 'guest': vhost '/' is
down
```

First, verify that your services are up. If there is an issue with your RabbitMQ volume, you can clear the volume with the following commands:

```
docker service rm iservices_rabbitmq
docker volume rm iservices_rabbitdb
```

If you get a message stating that the volume is in use, run the following command:

```
docker rm <id of container using volume>
```

Re-deploy PowerFlow by running the following command:

```
docker stack deploy -c /opt/iservices/scripts/docker-compose.yml iservices
```

> **NOTE**: Restarting Docker does not clear the queue, because the queue is persistent. However, clearing the queue with the commands above might result in data loss due to the tasks being removed from the queue.

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

```
docker stack deploy -c <compose-file> iservices
```

Dynamically Scaling for More Workers

Run the following command:

```
docker service scale iservices_steprunner=10
```

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnosis Tools

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

- *Docker PowerPack*. This PowerPack monitors your Linux-based PowerFlow server with SSH (the PowerFlow ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your PowerFlow server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor PowerFlow, see the "Using SL1 to Monitor SL1 PowerFlow" chapter in the **SL1 PowerFlow Platform** manual.

- *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 PowerFlow 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 PowerFlow 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/iservices** 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 PowerFlow 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 PowerFlow 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 with "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 PowerFlow user interface:

1. Select the PowerFlow application from the **Applications** page.
2. Hover over the **[Run]** (▶) from and select *Debug Run* from the pop-up menu. PowerFlow executes the application in Debug Mode with a log level of 10.

To run an application in Debug Mode using the API:

1. POST the following to the API endpoint:

   ```
   https://<PowerFlow>/api/v1/applications/run
   ```

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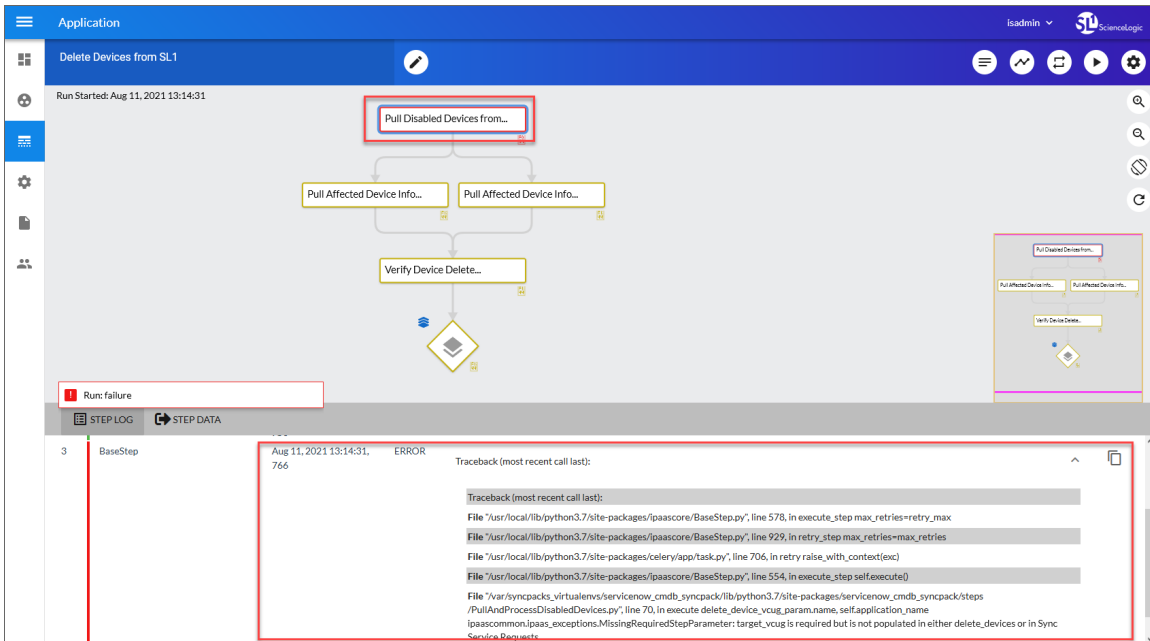
You can also run an application in debug using curl via SSH:

1. SSH to the PowerFlow instance.
2. Run the following command:

```
curl -v -k -u isadmin:em7admin -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 appendix includes the following topics:

Staging Tables

The integration uses the [ImportSet API](#). 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

Name	Table
SG-ScienceLogic Physical Devices	x_sclo_devicesync_device_import
SG-ScienceLogic VMware Devices	x_sclo_devicesync_device_import_vmware

URL Format

Name	Value
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": [
    {
      "sl1_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",
        "sl1_url": "https://10.2.11.154/em7/index.em7?exec=device_summary&did=45",
        "oid": "1.3.6.1.4.1.19567.2.1.10"
    }
  ]
}
```

Example Response

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

ScienceLogic