



---

# ServiceNow CMDB SyncPack

Version 4.0.0

---

# Table of Contents

<b>Introduction to the ServiceNow CMDB SyncPack</b> .....	<b>5</b>
What Can I Do with this SyncPack? .....	6
Contents of the ServiceNow CMDB SyncPack .....	7
Skylar Automation Applications .....	7
Skylar Automation Applications (Internal) .....	8
<b>Installing ServiceNow CMDB SyncPack</b> .....	<b>9</b>
Prerequisites for the ServiceNow CMDB SyncPack .....	10
Administrator Access .....	10
Port Access .....	10
Architecture Overview for ServiceNow SyncPacks .....	11
Downloading, Importing, and Installing the SyncPack .....	11
Downloading the SyncPack .....	12
Importing the SyncPack .....	12
Installing the SyncPack .....	13
Installing or Upgrading in an Offline Deployment .....	13
Allowing Cross-Scoped Access in ServiceNow .....	14
Installing the "ScienceLogic Skylar One: CMDB Automation" Application in ServiceNow .....	16
Requesting and Installing the Certified Application .....	16
Creating a ServiceNow Group .....	16
Creating a ServiceNow User .....	17
Installing and Activating ServiceNow Plugins .....	18
Installing the ScienceLogic Domain Separation (Global) Update Set in ServiceNow .....	18
Overview of the Update Set .....	19
Limitations of the Identification Engine .....	19
Installing the Update Set .....	19
Using ServiceNow Domain Separation with Skylar Automation .....	20
User Setup .....	21
Example 1 .....	21
Example 2 .....	22
<b>Configuring Device Sync for the CMDB SyncPack</b> .....	<b>23</b>
Creating and Aligning a Configuration Object .....	24
Creating a Configuration Object .....	24
Aligning a Configuration Object .....	26
Creating an OAuth2 Credential Record in ServiceNow .....	27

Syncing Devices from Skylar One to ServiceNow .....	28
Merged Devices in Skylar One .....	28
Using Other Data Sources with Merged Devices .....	29
Running a Device Sync .....	29
Using a Jinja2 Template .....	36
Example: A Basic Template for Device Attributes .....	36
Example: An Advanced Template for Device Sync .....	37
Example: Another Advanced Template for Device Sync .....	38
Example: Advanced Template for Device Class, Sub Category, and Model .....	38
Filtering Device Sync by Device Group .....	38
Using GraphQL to Create the Custom Filter .....	39
JSON Code Template .....	40
Adding Device Class Mappings .....	41
Persistently Saving Device Class Mappings with the API .....	43
Checking for Missing Device Mappings .....	45
Device Attribute Mappings .....	48
Default Device Attribute Mappings .....	49
Skylar One Device Attributes Available for Syncing .....	50
Adding New Device Attributes to ServiceNow .....	52
Configuring Customer CI Relation Overrides .....	52
Viewing Reports for Device Sync .....	56
Log Messages for the "Generate Required CI Relations for ServiceNow" Application .....	57
<b>Configuring Additional Syncs for the CMDB SyncPack .....</b>	<b>59</b>
Syncing Business Services .....	60
Syncing Business Services from Skylar One to ServiceNow .....	60
Syncing Business Services from ServiceNow to Skylar One .....	64
Aspects of Syncing Business Services from ServiceNow to Skylar One .....	64
Syncing CI Attributes from ServiceNow to Skylar One .....	68
Creating a Link in a Skylar One Device to a ServiceNow CI .....	72
Adding Related Items and Lists to the CI Record in ServiceNow .....	73
Deleting Devices .....	75
Syncing File Systems from Skylar One to ServiceNow .....	76
Syncing Installed Software between Skylar One and ServiceNow .....	80
Syncing Interfaces from Skylar One to ServiceNow .....	82
Scheduling Skylar Automation Applications .....	89

Log Messages for the "Generate Required CI Relations for ServiceNow" Application .....	92
<b>Troubleshooting the CMDB SyncPack .....</b>	<b>95</b>
Initial Troubleshooting Steps .....	96
Skylar Automation .....	96
ServiceNow .....	96
Resources for Troubleshooting .....	96
Useful Skylar Automation Ports .....	96
Helpful Docker Commands .....	97
Viewing Container Versions and Status .....	97
Restarting a Service .....	97
Stopping all Skylar Automation Services .....	97
Restarting Docker .....	98
Diagnosis Tools .....	98
Retrieving Additional Debug Information (Debug Mode) .....	98
Troubleshooting CMDB Sync .....	100
Issues Creating CIs in ServiceNow .....	100
Check the Device Remediation Table .....	101
Enabling Debugging of the CI Payload .....	101
<b>Mappings between Skylar One, ServiceNow, and Other Applications .....</b>	<b>103</b>
Overview .....	105
VMware vCenter .....	105
Default Customer Relationship Overrides .....	106
Specific Class Breakdown .....	107
VMware vCenter Folder .....	108
VMware Datastore .....	108
VMware Virtual Machine Instance .....	109
VMware Cluster .....	109
ESX Server .....	110
ESX Resource Pool .....	110
vCenter Network .....	110
Distributed Virtual Switch .....	111
Distributed Virtual Portgroup .....	111

---

# Chapter

# 1

## Introduction to the ServiceNow CMDB SyncPack

---

### Overview

This chapter gives an overview of the "ServiceNow Configuration Management Database (CMDB)" SyncPack, which lets you integrate Skylar One with the ServiceNow Configuration Management Database (CMDB).

For more information about how Skylar One integrates with ServiceNow, including uses cases, certifications, and customer feedback, see [ServiceNow Integration](#) at the ScienceLogic website.

This chapter covers the following topics:

<i>What Can I Do with this SyncPack?</i> .....	6
<i>Contents of the ServiceNow CMDB SyncPack</i> .....	7

---

## What Can I Do with this SyncPack?

The "ServiceNow Configuration Management Database (CMDB)" SyncPack is the ScienceLogic integration with the ServiceNow Configuration Management Database (CMDB) Module. This SyncPack maintains and enhances the ServiceNow CMDB by discovering device information bi-directionally between Skylar One and ServiceNow and by automatically maintaining ServiceNow Configuration Item (CI) relationships.

This SyncPack includes the following applications:

- **Device Sync.** The "Sync Devices from Skylar One to ServiceNow" application syncs devices and virtual device relationships from Skylar One to ServiceNow. You can sync devices based on organization and collector group, and you can use "mappings" to connect a Skylar One device class to a ServiceNow CI class. You can also collect manufacturer/model attributes from asset records aligned with devices in Skylar One and sync that information with ServiceNow. For more information, see [Syncing Devices from Skylar One to ServiceNow](#).
- **Business Service Sync.** The "Sync Business Services from ServiceNow to Skylar One" application syncs services that were defined in ServiceNow with business services in Skylar One, recreating the service structure in Skylar One, where you can see the relationships between all the components. For more information, see [Syncing Business Services](#).
- **CI Attribute Sync.** The "Sync CI Attributes from ServiceNow to Skylar One" application imports CI attributes from ServiceNow to the relevant asset and attribute fields in Skylar One. The CI Attribute Sync supports assets, asset configuration, asset maintenance, location, production statuses, and custom attributes. For more information, see [Syncing CI Attributes from ServiceNow to Skylar One](#).
- **File System Sync.** The "Sync File Systems from Skylar One to ServiceNow" application reads file systems discovered in Skylar One and then maps them to a parent CI record in ServiceNow. For more information, see [Syncing File Systems from Skylar One to ServiceNow](#).
- **Interface Sync.** The "Sync Interfaces from Skylar One to ServiceNow" application collects interface data from ServiceNow and Skylar One and runs multiple CI syncs for each interface to be synced. For more information, see [Syncing Interfaces from Skylar One to ServiceNow](#).
- **Installed Software Sync.** The "Sync Installed Software from Skylar One to ServiceNow" reads all software packages from Skylar One and creates new CIs in ServiceNow. Then, the "Sync Installed Software from Skylar One to ServiceNow" then reads all available software packages from Skylar One and the devices aligned to that software by region and syncs them with ServiceNow. For more information, see [Syncing Installed Software between Skylar One and ServiceNow](#).

**NOTE:** Due to lack of data in Skylar One, software that is synced using the "Sync Installed Software from Skylar One to Service Now" application might conflict with the output of discovery from ServiceNow or other sources that populate software information in ServiceNow.

---

# Contents of the ServiceNow CMDB SyncPack

This section lists the Skylar Automation applications that are in the "ServiceNow CMDB" SyncPack.

**NOTE:** Starting with version 3.5.0 of this SyncPack, the "Sync Advanced Topology from Skylar One to ServiceNow" application was removed. The functionality in that application was added to the "Sync Devices from Skylar One to ServiceNow" and the "Sync Interfaces from Skylar One to ServiceNow" applications.

## Skylar Automation Applications

The following applications are included with the "ServiceNow CMDB" SyncPack:

- **Create Custom Attributes and ServiceNow Custom Link in Skylar One.** Creates custom device attributes in Skylar One, which Skylar One uses to create a custom link in Skylar One that redirects to the ServiceNow CI. In Skylar One the link appears as a *ServiceNow* link in the **Tools** menu on the **Device Investigator** window for the corresponding device.
- **Delete Devices from Skylar One.** Lets you delete devices in a specific Skylar One Virtual Collector Group if those devices have not been modified in Skylar One for a specified amount of time that is set in the application.
- **Generate Required CI Relations for ServiceNow.** Pulls device class mappings from the "Sync Devices from Skylar One to ServiceNow" and the "Sync CI Attributes from ServiceNow to Skylar One" applications to prevent you from having to add a separate set of class mappings. The application also lists any missing relationships in the Step Log in the Skylar Automation user interface. Please note that this application is a report used by Skylar Automation, and it does not send any data to ServiceNow.
- **Report: Identify Unmapped Devices Classes.** Pulls the class mappings from Device Sync and Attribute Sync and compares the mappings with the full list of device classes of discovered devices in Skylar One. The application generates a report on the **Reports** page that lists missing mappings, and if any device classes are unmapped, the application generates an event in the target Skylar One system.
- **Sync Business Services from Skylar One to ServiceNow.** Reads Business Services, IT Services, and Device Services in Skylar One and syncs them with business services in ServiceNow. This application creates and updates services, but it does not delete services.
- **Sync Business Services from ServiceNow to Skylar One.** Syncs services that were defined in ServiceNow with Business Services in Skylar One. By syncing services from Service Now to Skylar One, you can see the relationships between the service components, the application components, and the infrastructure components in Skylar One.
- **Sync CI Attributes from ServiceNow to Skylar One.** Reads CI attributes from ServiceNow and maps those attributes to asset and attribute fields in Skylar One. This application uses the mappings and additional attributes options from the "Sync Devices from Skylar One to ServiceNow" application. This application can also sync the location and production state attributes from ServiceNow to Skylar One.

- **Sync Device Groups from Skylar One to ServiceNow.** Collects all device groups and group IDs from Skylar One and posts device group data to ServiceNow. To prevent errors when running this application or a device sync, make sure that the device group names are not already being used by existing groups in ServiceNow.
- **Sync Devices from Skylar One to ServiceNow.** Syncs devices and their properties and relationships from Skylar One to ServiceNow.
- **Sync File Systems from Skylar One to ServiceNow.** Reads file systems discovered in Skylar One and then maps them to a parent CI record in ServiceNow.
- **Sync Installed Software from Skylar One to ServiceNow.** Reads all available software packages from ServiceNow and the devices aligned to that software by region and syncs them with Skylar One. Run this application after running the "Sync Software Packages from Skylar One to ServiceNow" application.
- **Sync Interfaces from Skylar One to ServiceNow.** Collects network interface data from ServiceNow and Skylar One, and then runs multiple CI syncs for each interface to be synced.
- **Sync Software Packages from Skylar One to ServiceNow.** Reads all software packages from and creates new CIs in ServiceNow. Run this application before running the "Sync Installed Software" application.

## Skylar Automation Applications (Internal)

To view the internal Skylar Automation applications, click the Filter icon (☰) on the **Applications** page and select *Show Hidden Applications*. Internal applications are hidden by default. The following applications are "internal" applications that should not be run directly, but are automatically run by applications from the previous list:

- **Bulk Delete Devices.** Deletes devices from Skylar One.
- **Create ServiceNow CI.** Creates a new ServiceNow CI with a mappings dictionary, but does not attempt to look up new CIs.
- **Post Attribute DB Calls to Skylar One.** Posts attribute database calls to Skylar One.
- **Post Attribute Rest Calls to Skylar One.** Posts attribute REST calls to Skylar One.

## Installing ServiceNow CMDB SyncPack

---

### Overview

This chapter explains how to install and configure the various applications used by the "ServiceNow CMDB" SyncPack.

The following workflow covers how to install and configure this SyncPack:

1. *Review the prerequisites and background information for this SyncPack.*
2. *In Skylar Automation, download, import, and install the "ServiceNow CMDB" SyncPack.*
3. *In ServiceNow, enable cross-scoped access (if needed).*
4. *In ServiceNow, install the "ScienceLogic Skylar One: CMDB & Incident Automation" application (also called the "Scoped Application").*
5. *In ServiceNow, create a ServiceNow Group and then create a ServiceNow User .*
6. *In ServiceNow, install and activate the relevant plugins.*
7. *In ServiceNow, enable the ServiceNow Identification and Reconciliation Module.*
8. *In ServiceNow, install the "ScienceLogic Domain Separation (Global)" update set (for domain-separated ServiceNow instances only).*

---

# Prerequisites for the ServiceNow CMDB SyncPack

This section describes the prerequisites for the ServiceNow SyncPacks. For more information about the specific software versions required by a ServiceNow SyncPack, see the [release notes](#) for that SyncPack.

## Administrator Access

To install this SyncPack, you must have administrator access to both Skylar One and ServiceNow. Specifically, you will need:

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

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

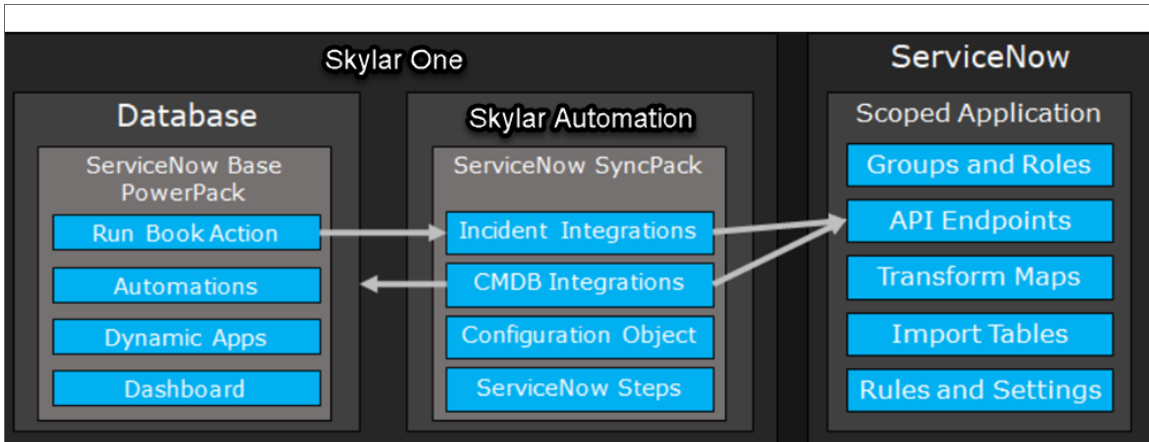
## Port Access

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

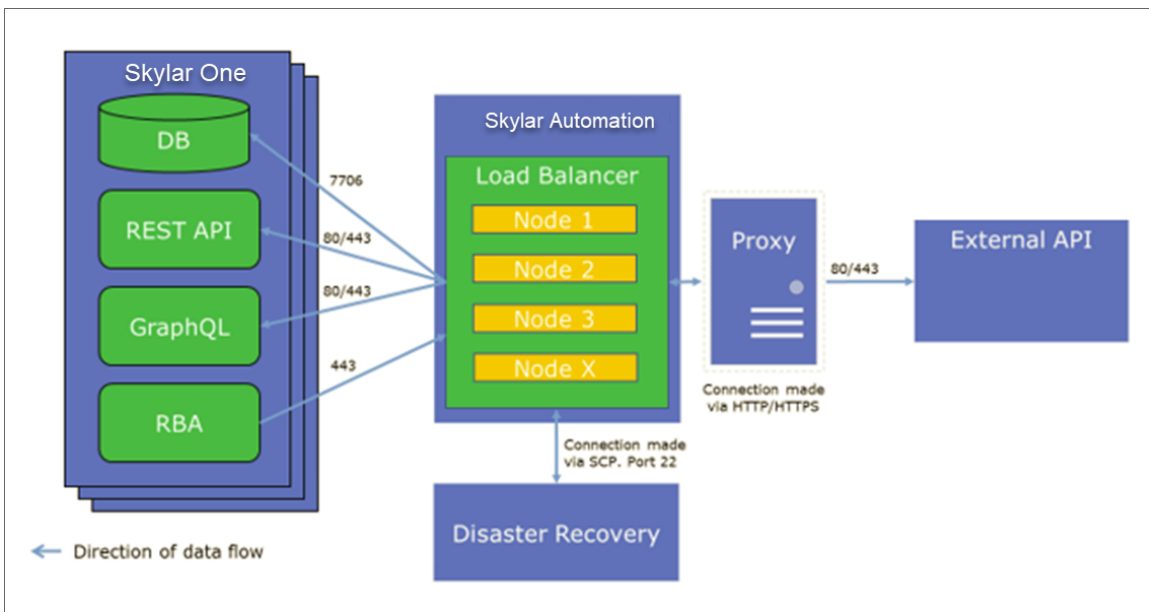
Source	Port	Requirement
Skylar One API	443	Skylar One API Access
ServiceNow API	443	ServiceNow API Access
Skylar One Database	7706	Skylar One Database Access

## Architecture Overview for ServiceNow SyncPacks

The following diagram details the various elements that are contained in Skylar One and the Skylar Automation system, and how Skylar Automation sits between the core Skylar One platform and an external data platform:



The following diagram provides an example of the high-level architecture of a Skylar Automation system with High Availability, Disaster Recovery, and a proxy configured:



## Downloading, Importing, and Installing the SyncPack

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

**IMPORTANT:** If you are planning to *upgrade* from an older version of this SyncPack to version 3.5.x, see [Upgrading to Version 3.5.x](#).

## Downloading the SyncPack

**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

You must import and install the "ServiceNow Base" SyncPack before uploading and installing any of the other ServiceNow SyncPacks.

If you are upgrading to version 3.5.0 or later of this SyncPack, be sure to install and activate version 1.5.0 or later of the "Base Steps" SyncPack. Older versions of the "Base Steps" SyncPack might cause installation errors for 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. However, any mappings you added to the **attribute\_mappings** section for the "Sync Devices from Skylar One to ServiceNow" application *are* 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 or Upgrading in an Offline Deployment

Use the following procedure if you are installing or upgrading this SyncPack in an offline deployment.

This release of the SyncPack requires the following external files for an offline deployment:

- **Jinja2-2.11.1-py2.py3-none-any.whl**, available at [this location](#) at pypi.org.
- **MarkupSafe-1.1.1-cp37-cp37m-manylinux1\_x86\_64.whl**, available at [this location](#) at pypi.org.

To upload the required external files:

1. After downloading the two required external files, SCP the files to the master node.
2. Run the following commands on the host:

```
devpi use 'https://<is_username>:<is_password>@<is_hostip>:3141/isadmin/dependencies'
```

```
devpi login <is_username> --password=<is_password>
```

```
cd /tmp/
```

```
devpi upload Jinja2-2.11.1-py2.py3-none-any.whl
```

```
devpi upload MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl -force
```

**NOTE:** If you cannot run these commands on the host, you can instead run them on the pypiserver container.

3. Perform a docker copy to both files (if you are running commands in the pypiserver container):

```
docker cp <file1-location> $(docker ps -q -f name=services_pypiserver):/tmp
```

```
docker cp <file2-location> $(docker ps -q -f name=services_pypiserver):/tmp
```

4. Follow the steps in [Installing the SyncPack](#) to install and activate this SyncPack.

---

## Allowing Cross-Scoped Access in ServiceNow

When using custom tables, you might need to configure cross-scope access for the ScienceLogic plugin. The following examples contain errors that might occur when cross-scope access is required.

Example of an API response:

```
{"results":[{"error":{"message":"com.glide.script.fencing.access.ScopeAccessNotGrantedException: read access to ui_test_hardware not granted","detail":"","status":"failure"}}
```

Example of navigating to a URL directly from a web browser when cross-scope access is required:

**This page contains the following errors:**



error on line 1 at column 1: Document is empty

**Below is a rendering of the page up to the first error.**

In this example, the table requires that you grant access to the ScienceLogic Scope to allow the API call to run correctly. In the above example, the target table is **u\_test\_hardware**.

**NOTE:** A ServiceNow account with System Administrator is required.

To grant access to the ScienceLogic Scope in ServiceNow:

1. Log in to your ServiceNow instance.
2. For newer releases of ServiceNow, click the globe icon () to change the scope. On older releases, click the **Settings** icon () and select the **Developer** tab. The **Developer System Settings** window appears.
3. From the **Application** drop-down list, select *ScienceLogic ServiceNow Integration*.
4. Close the **Developer System Settings** window and navigate to the **Cross scope privileges** page (System Applications > Application Cross-Scope Access). Make sure you are in the "ScienceLogic ServiceNow Application" scope and track these updates in an update set.
5. Click the **[New]** button to create a new record on the **Cross scope privileges** page:
6. Verify that the **Source Scope** and **Application** fields are set to *ScienceLogic ServiceNow Integration*. If they are not, repeats steps 2-3.
7. Complete the following fields:
  - **Target Scope**. Specify the scope of the target table, such as *Global*. Be sure to verify the application to which the table belongs, and use that value as the target scope in this field.
  - **Operation**. Select *Read*.
  - **Target Name**. Specify the name of the target table.
  - **Status**. Select *Allowed*.
  - **Target Type**. Select *Table*.
8. Click the **[Submit]** button.

For more information, see the [Cross-scope privilege record](#) topic in the ServiceNow documentation.

---

# Installing the "ScienceLogic Skylar One: CMDB Automation" Application in ServiceNow

"ServiceNow CMDB" SyncPack version 4.0.0 and later integrates with the standalone "ScienceLogic: CMDB Automation" application in ServiceNow. The previous "ScienceLogic SL1: CMDB & Incident" application is no longer utilized, as its functionality has been migrated to other standalone applications.

## Requesting and Installing the Certified Application

You must first request the "ScienceLogic: CMDB Automation" application from the ServiceNow Store, and then you can install it. You must have a ServiceNow HI Service Account to request this application and download it onto your ServiceNow instance.

To request and install the certified application:

1. Go to the ServiceNow Store at <https://store.servicenow.com> and search for "ScienceLogic".
2. Select the "ScienceLogic: CMDB Automation" application. The detail page for the application appears.
3. Click the **[Get]** button 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 or My Company Applications).
5. Click **[Downloads]** in the menu header or search for "ScienceLogic".
6. Click the version drop-down for the "ScienceLogic Skylar One: CMDB & Incident Automation" application listing to make sure you are using the correct version of the application that is compatible with your version of this SyncPack.
7. Click the **[Install]** button for the 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 left-hand navigation menu to verify that the application was installed. You might need to log out of ServiceNow and log in again to see the updated left-hand navigation menu.

---

## 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 Incident management:

1. In ServiceNow, go to the **Groups** page (System Security > Users and Groups > Groups) and click **[New]**. A **New record** page appears.
2. In the **New record** page, type the group name and any additional information. **Name** is the only required field.
3. Click **[Submit]**.

4. Select the new group from the **Groups** page, and at the bottom of the Group record, select the **[Roles]** tab and click **[Edit]**.
5. Search for `x_sclo_scilogic.Admin` and move it to the **Roles List** column using the arrow buttons.
6. Click **[Save]**. Your ServiceNow Group now has an assigned Role.
7. 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:

1. In ServiceNow, go to the **Users** page (System Security > Users and Groups > Users) and click **[New]**. A **New record** page appears.
2. 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)*.
3. Right-click the gray header and click *Save* to save the user.
4. Select the **[Groups]** tab at the bottom of the record and click the **[Edit]** button.
5. Find the group you created previously and move the group to the right-hand column using the arrow buttons.
6. 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 Skylar Automation configuration object.

---

## Installing and Activating ServiceNow Plugins

This SyncPack requires the "ServiceNow Configuration Management for Scoped Apps (CMDB)" plugin (com.snc.cmdb.scoped), which gives the ScienceLogic scoped application access to the Identification Engine APIs. You will need to install this plugin on your ServiceNow instance.

In addition, ScienceLogic strongly recommends that you install the "CMDB CI Class Models" plugin (sn\_cmdb\_ci\_class). This plugin contains all new class models provided by ServiceNow.

To install and activate plugins in ServiceNow, you must have ServiceNow Instance administrator rights.

To install one or both of the plugins in ServiceNow:

1. In ServiceNow, log in and navigate to **Plugins** (System Definition > Plugins). This page is only available with administrative rights.
2. Search for a plugin by its name or its ID (such as sn\_cmdb\_ci\_class) and select the plugin. The options to **Install** or **Update** appear, depending on the status of the plugin:
  - **Install**. Installs the plugin in ServiceNow if it has not been installed before. After you install the plugin, the button is grayed out and the text changed to **Installed**.
  - **Update**. Shows that newer versions of the plugin are available.
3. Click **Install** or **Update** to complete the installation.

---

## Installing the ScienceLogic Domain Separation (Global) Update Set in ServiceNow

If your ServiceNow environment is **domain-separated**, where the data, processes, and administrative tasks have been organized into logical groupings called domains, you will need to install the latest version of the "ScienceLogic Domain Separation (Global)" update set in ServiceNow. This update set is *not* included in the "ScienceLogic Skylar One: CMDB & Incident Automation" application (also called the Certified application).

**NOTE:** Ask your ScienceLogic contact for access to this update set.

For more information about ServiceNow domain separation, see [Using ServiceNow Domain Separation with Skylar Automation](#).

**IMPORTANT:** If your ServiceNow environment does *not* use domain separation, you can skip this topic.

## Overview of the Update Set

The "ScienceLogic Domain Separation (Global)" update set contains the following items:

- Scripted REST API
- Scripted REST Resource
- Scripted REST Query Parameter
- Scripted REST Query Parameter Association
- Script Include

This update set completely separates the ServiceNow Identification Engine REST resource that is used in the "ScienceLogic ServiceNow Integration" application and all of the required resources and duplicates it in the Global scope.

A Scripted REST Service in the Global application is a direct copy of the application endpoint with a new name: `api/10693/sciencelogic_domain_separation`. This REST Service includes only one Resource: `Device IdentificationEngine POST`. This resource works exactly like the application version, but it points to the new Script Include "SciLoDomainSepUtil". This version of the REST resource takes the same formatted JSON as the Certified application.

The Script Include "SciLoDomainSepUtil" includes all of the functionality needed to run the ServiceNow Identification Engine API.

Additional resources for the ServiceNow API:

- [CMDB Identification and Reconciliation](#)
- [identifyCI\(String jsonString\)](#)
- [createOrUpdateCI\(String source, String input\)](#)
- [Identification engine error messages](#)

**NOTE:** The only resource shared with this update set and the Certified application is the Device Properties page. These properties are located in the Certified application at ScienceLogic > Device > Device Properties.

## Limitations of the Identification Engine

For more information about how the Identification Engine handles incoming payloads in domain-separated systems, see the following ServiceNow Knowledge Base article: [KB0695949](#).

The payload and the user domain must match, or the ServiceNow Identification Engine (IDE) will by default insert the CMDB record. Safeguards within the Skylar Automation Device Sync application were put in place for payloads that have relationships. The application will drop the payload if all Configuration Items do not share the same domain.

## Installing the Update Set

To install the "ScienceLogic Domain Separation (Global)" update set:

1. Ask your ScienceLogic contact for access to this update set.
2. In ServiceNow, navigate to the **Retrieved Update Sets** page (System Update Sets > Retrieved Update Sets).
3. Click the **Import Update Set from XML** link under **Related Links**.
4. Click **[Browse]** and navigate to the update set XML file you downloaded. Select the XML file and click **[Upload]**.
5. After the file is uploaded, the **Retrieved Update Sets** page appears. Click the link for the "ScienceLogic Domain Separation (Global)" update set. The **Retrieved Update Set** page appears.
6. Click **[Preview Update Set]**. After the preview set runs, a status page appears.
7. Ensure that "Success" appears in the **Completion code** field. If "Success" does not appear in the **Completion code** field, contact ScienceLogic Support to assist with reviewing any conflicts that might exist. Do not proceed until those conflicts are resolved and "Success" appears in the **Completion code** field.
8. Click **[Commit]** to commit the fix script after running the preview set.
9. Before you start to sync devices, you must select the **Domain Separation** option on the **Configuration** pane in the "Sync Devices from Skylar One to ServiceNow" application. This option ensures that Skylar Automation gets re-pointed to the API endpoint after you install the "ScienceLogic Domain Separation (Global)" update set. For more information, see [Running a Device Sync](#).

## Using ServiceNow Domain Separation with Skylar Automation

To implement domain separation effectively, the "ServiceNow SL1: Service Catalog Automations" application in ServiceNow is needed to establish connections between matching companies and organizations.

The following topics provide more information about ServiceNow domain separation and how it relates to Skylar Automation.

**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.

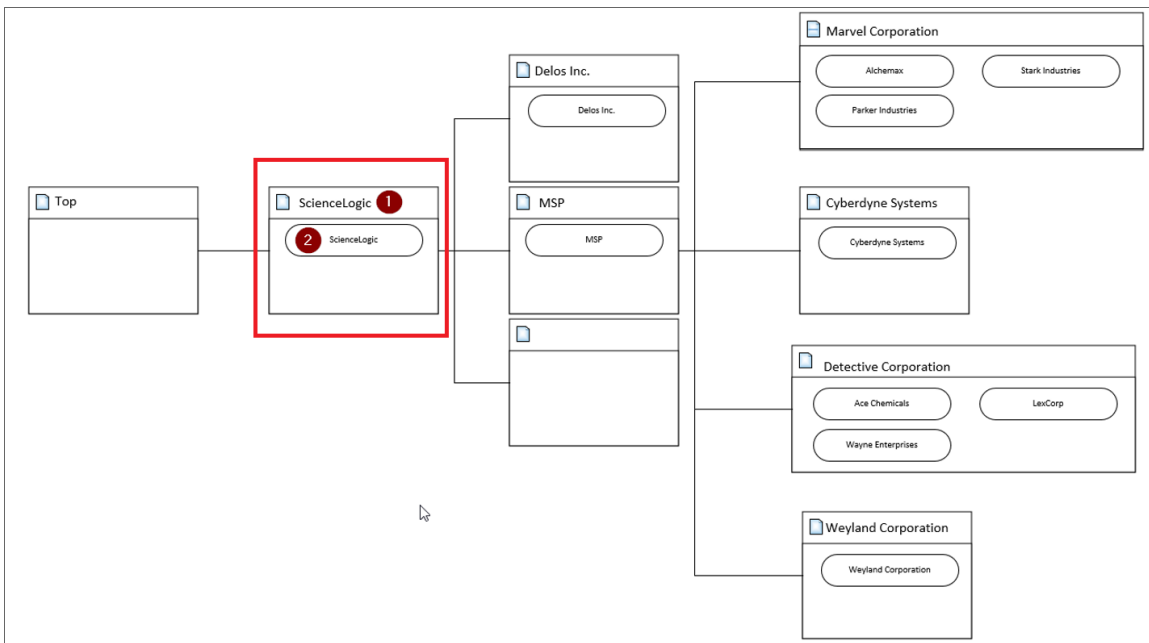
**NOTE:** If you get an error message stating that a device or CI has no **domain\_sys\_id** value, that means that the company for that CI has no domain.

# User Setup

Company and domain setup is critical for the domain separation integration to work using the Identification Engine provided by ServiceNow. This solution requires only one user and will require proper setup depending on where the user is located within the domain tree.

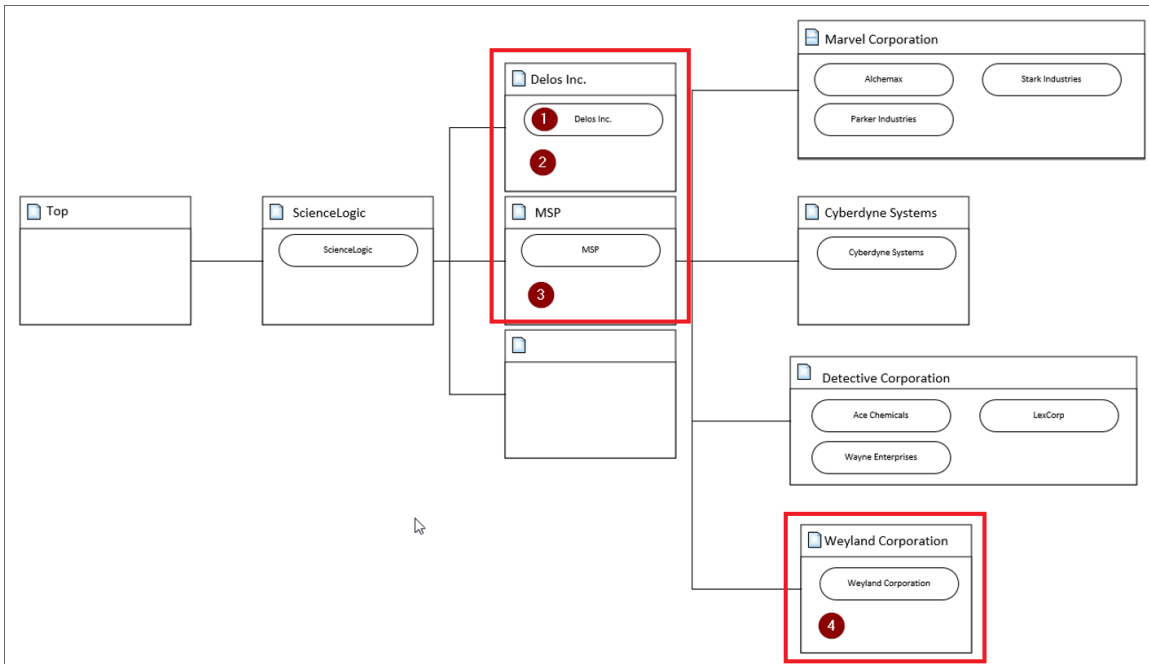
## Example 1

In the following example, **ScienceLogic (1)** is both the domain and the company. The ScienceLogic user service account is associated with **ScienceLogic (2)** company, and it will have access to all child domains. You do not need to set visibility to any domain. This is the best way to set up this user, because placing it in the top domain ensures that it always has access to all children:



## Example 2

In the following example, **Delos Inc. (1)** is the company within the **Delos Inc.** domain. The Skylar Automation service account is associated with the **Delos Inc. (1)** company. The **Delos Inc.** domain has no children domains, and if domain visibility is not assigned, Skylar Automation will not properly update the CMDB. This setup works, but it requires that proper domain visibility is set up for the service account to work correctly.



**NOTE:** Assigning visibility to **MSP (3)** will grant the service account access to all child domains. Assigning visibility to **Weyland Corporation (4)** will only allow access to the **Delos Inc.** domain and the **Weyland** domain; all other domains will not work.

---

# Chapter

# 3

## Configuring Device Sync for the CMDB SyncPack

---

### Overview


This chapter describes the how to configure and run Device Sync in the "ServiceNow CMDB" SyncPack. For more information about the other syncs in this SyncPack, see [Configuring Additional Syncs for the CMDB SyncPack](#).

This chapter covers the following topics:

<a href="#">Creating and Aligning a Configuration Object</a> .....	24
<a href="#">Syncing Devices from Skylar One to ServiceNow</a> .....	28

---

## 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 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:** Each unique stack that includes a Skylar One, Skylar Automation, and a ServiceNow Instance requires an individual configuration object with a unique region.

## 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. For the "ServiceNow SyncPack" configuration object, click the **[Actions]** button (  ) 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.

**IMPORTANT:** 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": "s11_db_host",
  "value": "${config.s11_host}"
},
```

For example:

```
{
  "encrypted": false,
  "name": "s11_db_host",
  "value": "10.2.11.42"
},
```

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. If you want to adjust the version of the ServiceNow IRE endpoint to send CI payloads to, complete the following Configuration Data Values field:
- **snow\_api\_version.** If no value is entered, Skylar Automation will default to use v2. For proper interface handling, ScienceLogic recommends entering v3. The only possible values are v2 or v3.

**NOTE:** In order to use "v3" for the snow\_api\_version, you must install the latest version of the "ScienceLogic ServiceNow Integration (v1.0.81)+ Interface Hotfix" update set in ServiceNow. Ask your ScienceLogic contact for access to this update set.


10. 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 the application you want to use.

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.
2. From the **Configurations** drop-down, select the configuration object you want to use.
3. Click **[Save]** to align that configuration with the application.
4. Repeat this process for every other application you want to use.

**NOTE:** The values for the parameters that display in the **Configuration** pane with a padlock icon (  ) are populated by the values in the configuration object.

## 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 header with 'Update' and 'Delete' buttons. Below is a blue box with OAuth client application details, including fields for Name, Client ID, Client Secret, Refresh Token Lifespan, Access Token Lifespan, Redirect URL, and Enforce Token Restriction. The main form contains fields for Name (PowerFlow Token), Client ID (85abad6bed82c2507f0168b91a1c1bcf), Client Secret (masked), Redirect URL, Logo URL, Public Client (checkbox), Application (Global), Accessible from (All application scopes), Active (checkbox), Refresh Token Lifespan (8,640,000), Access Token Lifespan (3,600), and Login URL. A Comments field is also present. At the bottom, there is an 'Auth Scopes' table with one row for 'useraccount' and an 'Insert a new row...' option.

## Syncing Devices from Skylar One to ServiceNow

The "Sync Devices from Skylar One to ServiceNow" application syncs devices and virtual device relationships from Skylar One to ServiceNow. You can also sync devices based on organization and collector group.

The Device Sync process uses rules or "mappings" that you can define in the "Sync Devices from Skylar One to ServiceNow" application. These mappings connect a **Skylar One device class** to a **ServiceNow CI class**, which determines the CI class that ServiceNow uses when creating the CI in ServiceNow.

**NOTE:** For more information about building **service rules** (containment rules and hosting rules) for devices and CIs, see [Configuring Service Rules for Device Sync](#).

The "Sync Devices from Skylar One to ServiceNow" application can also collect manufacturer and model attributes from asset records aligned with devices in Skylar One and sync that information with ServiceNow. For more information, see [Device Attribute Mappings](#).

## Merged Devices in Skylar One

When the "Sync Devices from Skylar One to ServiceNow" application encounters a merged device in Skylar One, it splits the record into two objects to allow for correct default relationships in ServiceNow.

Starting with version 3.2.0 of this SyncPack, the "Sync Devices from Skylar One to ServiceNow" application syncs *both* the physical and the component device in Skylar One to ServiceNow. When a merged device is encountered in Skylar One, the "Sync Devices" application splits the device in Skylar Automation and creates two CIs in ServiceNow. This action does not impact the source device record in Skylar One.

In ServiceNow, the physical CI includes the relevant asset information. A relationship also exists between the physical CI and the virtual CI. The asset information is directly copied between both CIs, so the data will essentially be duplicated across both devices, and the data will be submitted to two separate tables. The *sl1\_url* will also be the same on both devices, so that both CIs will point to the same device in Skylar One.

## Using Other Data Sources with Merged Devices

If you have other data sources syncing into the ServiceNow CMDB and you have merged devices in Skylar One, ScienceLogic recommends caution when integrating to the CMDB.

Also, ScienceLogic recommends that you ensure that configuration of the Identification and Reconciliation (IRE) within ServiceNow affects all data sources that are integrating into it. In the case of ScienceLogic, this is most apparent when syncing merged devices. Modifications to the IRE to handle merged devices affects all other data sources that sync to those specific class tables. It is your responsibility to understand each data source, how that data source integrates with the ServiceNow CMDB, and how to leverage that knowledge to understand the impact IRE changes may have.

**WARNING:** ScienceLogic cannot be held responsible for any duplicate, lost, or incorrect CI information as a result of merged devices when multiple data sources are involved. This scenario will also affect your Support SLAs, as this practice deviates from recommended best practices.

For more information about the ServiceNow Identification and Reconciliation module, see the ServiceNow documentation: [https://docs.servicenow.com/bundle/orlando-servicenow-platform/page/product/configuration-management/concept/c\\_CompsandProcessIDandReconcil.html](https://docs.servicenow.com/bundle/orlando-servicenow-platform/page/product/configuration-management/concept/c_CompsandProcessIDandReconcil.html).

## Running a Device Sync

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

- **Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes.** Reads all existing Skylar One device classes and ServiceNow CI classes and caches them for the Device Sync. This application uses this data to populate the *mappings* drop-down values in the "Sync Devices from Skylar One to ServiceNow" application. Before version 3.5.0 of this SyncPack, this application was named "Cache ServiceNow CIs and Skylar One Device Classes".

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. To run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

- **Generate Required CI Relations for ServiceNow.** Determines if you are missing any class mappings or service rules that might be required in ServiceNow.

- **Sync Devices from Skylar One to ServiceNow.** Syncs devices and virtual device relationships from Skylar One to ServiceNow.

To sync Skylar One devices with ServiceNow:

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

If you change any of the containment rules or hosting rules in ServiceNow, you will need to run "Cache ServiceNow CIs and Skylar One Orgs, Device Classes" again. For more information, see [Configuring Service Rules for Device Sync](#).

2. Select the "Generate Required CI Relations for ServiceNow" application from the **Applications** page, click **[Configure]**, align a configuration object, and then click **[Run]**.

Skylar Automation uses the Device Class mappings you are going to configure in step 6, so you do not need to set up any mappings on the **Configuration** pane for the "Generate Required CI Relations for ServiceNow" application. Any mappings you add to this application will *overwrite* mappings in the "Sync Devices from Skylar One to ServiceNow" application.

3. When the "Generate Required CI Relations for ServiceNow" application completes, review the log information in the **Step Log** for the "Pull and Process Relations" step. You should see a log message stating that no missing relations were found. For more information, see [Log Messages for the "Generate Required CI Relations for ServiceNow" Application](#).

**NOTE:** If needed, address any missing class mappings or service rules. For more information on service rules, see [Creating a ServiceNow Update Set](#).

- Select the "Sync Devices from Skylar One to ServiceNow" application from the **Applications** page and click **[Configure]**. The **Configuration** pane appears:

Sync Devices From SL1 To ServiceNow

Modify configuration and save. Show JSON Configs

Configuration  
ven70\_sl151\_pf199\_cmdb

sl1_hostname 10.2.11.151	sl1_db_host \${config.sl1_host}	snow_hostname ven01770.service-now.com
sl1_user em7admin	snow_user powerflow_user_cmdb_incidt	sl1_password .....
snow_password .....	sl1_db_user clientdbuser	sl1_db_password .....
region ven70_sl151_pf199	read_timeout 300	Include_Orgs 8
Include_CUGs .....	sl1_chunk_size 500	snow_request_limit 500
selected_devices .....	sl1_url_override em7.scienceologic.com	excluded_devices device_name_a,device_id_232
cache_lookup_chunk_size 1000	discovery_source Other Automated	snow_batch_size 150
snow_chunk_size 150	<input type="checkbox"/> exclude_inactive	<input type="checkbox"/> enable_device_active
<input type="checkbox"/> enable_asset_networks	<input checked="" type="checkbox"/> use_ap2_url	<input checked="" type="checkbox"/> unmerge_devices

Save

- 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.
- read\_timeout.** Specify the maximum amount of time in seconds that the application should wait for a HTTP Read response before timing out. The default is 300 seconds.
- sl1\_chunk\_size.** Specify the number of devices to pull from Skylar One in each chunk. The default is 500 devices per chunk.
- snow\_request\_limit.** Specify the number of CIs fetched from ServiceNow in each request. The default is 500 objects per chunk.
- snow\_encoded\_query.** Specify the filter expression for the ServiceNow GraphQL encoded query.
- disconnect\_field.** Acts as the boolean field on the Configuration Items table to determine "True/False" for the Skylar One monitoring status.

- **ci\_query\_time\_duration.** Determines the lookback window in days for pulling ServiceNow CIs determined by the **last scan** field on the **sys\_object\_source** table. If a CI is not returned on this query, but is still active in Skylar One, a "create" request will be sent for the CI which will update the timestamp.
- **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.
- **cache\_lookup\_chunk\_size.** Specify the number of CI correlation documents to pull from the PowerFlow cache in a chunk. The default is 1000 documents per chunk.
- **discovery\_source.** Specify the ServiceNow Discovery source. The default is "Other Automated".
- **snow\_batch\_size.** Specify the total number of CIs being sent to ServiceNow in each triggered application. The default is 150 objects per batch.
- **snow\_chunk\_size.** Specify the number of CI objects to send in each chunk or in each sub-payload with a batch (specified in the **snow\_batch\_size** field). The default is 150 objects per chunk or sub-payload.
- **max\_count\_for\_relationships.** Specify the maximum number of device IDs to query for sets of relationships in a single chunk from Skylar One.
- **Include\_Orgs.** If you want to include a specific set of Skylar One Organizations in the device 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.
- **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. Optional.

**CAUTION:** A misconfiguration in the **Include Orgs** field or the **Include\_CUGs** fields might change the monitoring flag for one or more devices. In other words, a misconfiguration in this field could switch the Skylar One Monitored flag from "true" to "false", removing that device or devices from the device sync.

- **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.
- **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. A device on the excluded list will still be queried, but it will be dropped during the Skylar Automation sync process. Optional.
- **verify\_snow\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
- **enable\_makes\_and\_models.** Toggle on (blue) this option to enable the makes and models query and sync.

- **exclude\_inactive**. Select this option to prevent syncing devices to ServiceNow that are not enabled, unavailable, or in maintenance. By default, this option 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 option is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template. For more information, see [Using 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 option is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template. For more information, see [Using a Jinja2 Template](#).

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

- **Domain\_Separation**. Select this option if your ServiceNow environment is *domain-separated*, where the data, processes, and administrative tasks have been organized into logical groupings called *domains*. If your ServiceNow instance is domain-separated, the user listed in the **snow\_user** field must be a member of the top domain and have access to *all* of the domains you intend to integrate. Also, ServiceNow should be the "source of truth" for organizations if your environment is domain-separated. If this option is selected, Skylar Automation syncs the ServiceNow company **sys\_id** with the corresponding Skylar One organization.
- **drop\_company**. Select this option if you want to remove the **sys\_id** in existing Companies from the sync, which prevents the "company" field from being sent to ServiceNow. Selecting this option has no effect if you selected the **Domain\_Separation** option for this application. By default, this option is not selected.

**NOTE:** If you select this option, the ServiceNow company **sys\_id** for the CI and the Skylar One organization **crm\_id** for the Skylar One device will not be compared when determining if an update has occurred. Do not select this option if you selected the **Domain\_Separation** option, as ServiceNow requires PowerFlow to send the "company" field in domain-separated environments. Also, do not select this option if you selected the **change\_device\_organizations** option, as changing the company for a CI in ServiceNow requires the "company" field to be sent.

- **change\_device\_organizations**. When enabled, this parameter allows devices to change companies with a Device Sync if the companies are within the same domain in ServiceNow. If you change the Organization/Company and another field or fields, but do not enable this parameter, PowerFlow will not apply any changes to the device. By default, this option is not selected.
- **retry\_jitter**. When selected, instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. By default, this option is not selected.

- **retry\_backoff**. When selected, instead of using a defined interval between retries, Skylar Automation will incrementally increase the interval between retries. By default, this option is not selected.
- **use\_ap2\_url**. Select this option if you want the device URLs in the **Skylar One URL** field in ServiceNow CIs to point to the corresponding **Device Investigator** page in the Skylar One user interface: `/inventory/devices/<Skylar One device ID>/investigator`. This option is enabled by default. If you turn off this option, the device URL points to the classic EM7 user interface: `/em7/index.em7?exec=device_summary&did=<Skylar One device ID>`.
- **unmerge\_devices**. De-select this option if you want to turn off the "unmerge" behavior added in version 3.2.0 of this SyncPack, where PowerFlow splits a merged device record into two objects to allow for correct default relationships in ServiceNow. This option is enabled by default.
- **generate\_report**. Select this option to create a report about the devices that you sync with ServiceNow. Skylar Automation generates a report every time you run the device sync application, and the reports are available on the **Reports** page of the Skylar Automation user interface.
- **Simulation\_Mode**. Select this option if you want to perform a simulated run of this application to show you the potential results of that run. By default, this option is not selected.
- **enable\_advanced\_topology**. Select this option to allow the pull and process advanced topology step to be executed when the sync is run. If disabled, this step will be skipped. This toggle defaults to a value of *True*.
- **enable\_disconnects**. If toggled on (blue), deletes the **sys\_object\_source** record for the ServiceNow CI. The "Sync Devices from SL1 to ServiceNow" application no longer uses a "monitored" field.
- **disconnect\_type**. Specify the disconnect method for disconnecting Skylar One entities from ServiceNow. Enter either *flip\_boolean\_field* or *remove\_sources\_entry* to determine the behavior for disconnects.

- **gql\_filter.** Using JSON, you can optionally define a custom GraphQL filter to apply to the devices 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 **DeviceSearch** and determine how to set up your custom filter. An example of a Device Search query using JSON: `{"name": {"doesNotBeginWith": "jc-is-ma"}}`

**NOTE:** You can use this field to filter devices by Device Group. For more information, see [Filtering Device Sync by Device Group](#).

**CAUTION:** A misconfiguration in this field might change the monitoring flag for one or more devices. For example, a misconfiguration in this field could switch the Skylar One Monitored flag from "true" to "false", removing that device or devices from the device sync. This field is an advanced feature that requires a basic knowledge of the Skylar One GraphQL implementation. For more information, ask your ScienceLogic contact for access to the ScienceLogic GraphQL documentation.

- **customer\_ci\_relation\_overrides.** To override existing relationship linking and directly control the link between Device Classes and attributes, add JSON code to this field. The JSON for this field includes default relationship overrides for VMware instead of direct parent/child relations. For more information, see [Mappings between Skylar One, ServiceNow, and Other Applications](#).
6. In the **Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit the mappings between ServiceNow CI classes and Skylar One device classes. This section is pre-loaded with a large number of default device class mappings. You can map a single ServiceNow CI class with multiple Skylar One device classes.

**NOTE:** The "Sync Devices from Skylar One to ServiceNow" application will *only* sync a device from Skylar One if the device class for that device is mapped to a ServiceNow CI class in the **Mappings** section. The default mappings in this section do not cover all technologies, however, and syncing additional technologies from Skylar One to ServiceNow might require additional research to understand the class structure.

For more information about how to view, edit, and create mappings, see [Editing Mappings in a Skylar Automation Application](#). This topic also covers how to use the **Company Mapping Override** and the **Attribute Mappings** sections, below.

7. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.

8. In the **Attribute Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit the mappings for any other custom device attributes you want to sync between Skylar One (the first column) and ServiceNow (the second column). For more information, including lists of default and available device attribute mappings, see [Device Attribute Mappings](#).

To sync Device Notes between Skylar One and ServiceNow, create a custom attribute for the device note in this section. This option works best if you only want a single value synced over so that value can remain in the notes. All custom attributes for each Skylar One device are automatically synced.

**TIP:** For the **Attribute Mappings** section, you can use a Jinja2 Template for device attribute fields on the Skylar One side (the left column). For more information, see [Using a Jinja2 Template](#).

**NOTE:** When an attribute value is "0" in Skylar One, the corresponding field in ServiceNow might display as empty.

9. Click **[Save]**. The **Configuration** pane closes.
10. Run the "Sync Devices from Skylar One to ServiceNow" application.
11. When the application completes, open the **Step Log** and review the log messages for the "Compare Skylar One Devices and ServiceNow CIs" step to see if any Device or CI records were added, updated, or disconnected from the sync. As needed, select the other steps to review the logs on the **Step Log** for those steps.

ScienceLogic recommends that you schedule a Device Sync to run every 24 hours. For more information, see [Scheduling PowerFlow Applications](#).

## Using a Jinja2 Template

The attribute mappings in Device Sync applications now support Jinja2 Templates, which let you sync complex, concatenated (linked) fields from Skylar One to ServiceNow. For example, you can add these complex values in the Skylar One side of the **attribute\_mappings** section of the **Configuration** pane for the "Sync Devices from Skylar One to ServiceNow" application, and that value is mapped to one or many fields in ServiceNow. For more information about Jinja2 Templates, see the [Template Designer Documentation](#).

In the "Sync Devices from Skylar One to ServiceNow" application, the Skylar One side can be a Template. In the "Sync CI Attributes from ServiceNow to Skylar One" application, the ServiceNow side can be a Template.

### Example: A Basic Template for Device Attributes

This example is included in the "Sync Devices from Skylar One to ServiceNow" application as the first default value in the **attribute\_mappings** section of the **Configuration** pane:

This template, when used on the Skylar One side of the **attribute\_mappings** section, populates the **short\_description** field in ServiceNow:

```
"Description: {{device.device_category}}, Device Class: {{device.device_class}}": [ "short_description" ]
```

In the above example, for a device with a **category: Testing** and a Device Class of **Testing | Testing**, the end result would be **Description: Testing, Device Class: Testing | Testing**, which will be posted to the **short\_description** field in ServiceNow.

The Jinja2 Templates will have access to all properties on the Device.

**NOTE:** Any item that is generated by a template is always a string.

## Example: An Advanced Template for Device Sync

The following example lets you get the active status of Skylar One Devices:

```
{%- set output = [] -%}
{%- if device.active.unavailable == True -%}{%- set output = output +
['Unavailable'] -%}
{%- endif -%}
{%- if device.active.userDisabled == True -%}{%- set output = output +
['User Disabled'] -%}
{%- endif -%}
{%- if device.active.userInitiatedMaintenance == True -%}{%- set output =
output +
['User Initiated Maintenance'] -%}{%- endif -%}
{%- if device.active.userMaintenance == True -%}{%- set output = output +
['User Initiated Maintenance'] -%}{%- endif -%}
{%- if output|length > 0 -%}
{{ ", ".join(output) }}
{%- else -%}
```

```
{{ "Active" }}
{%- endif -%}
```

## Example: Another Advanced Template for Device Sync

The following example shows another way to get the active status of Skylar One Devices:

```
{%- set prettify = {"userInitiatedMaintenance": "User Initiated Main-
maintenance", "systemDisabled":
"System Disabled", "maintenance": "System Maintenance", "unavailable":
"Unavailable",
"userDisabled": "User Disabled"} -%}
{%- set ns = namespace(maint=[]) -%}
{%- for k,v in device.active.items() -%}
{%- if v -%}
{%- set ns.maint = ns.maint + [prettify[k]] -%}
{%- endif -%}
{%- endfor -%}
{{", ".join(ns.maint) if ns.maint else "Active"}}
```

## Example: Advanced Template for Device Class, Sub Category, and Model

The following example shows how to gather device data based on Device Class and Sub Category if available, or Model if Device Class and Sub Category are not available:

```
{%- set output = [] -%}
{%- set output = device.device_class.split('|') -%}
{%- if (output[0]|trim) in ['Checkpoint', 'ScienceLogic', 'Microsoft'] -%}
{%- set output = output[1]|trim -%}
{% else %}
{%- set output = device.model -%}
{%- endif -%}
{{output}}
```

## Filtering Device Sync by Device Group

You can use the *gql\_filter* field on the **Configuration** pane of the "Sync Devices from Skylar One to ServiceNow" application to add a block of JSON code that lets you define a custom filter to apply to the devices in the sync.

**NOTE:** You cannot enter GraphQL code in the *gql\_filter* field in Skylar Automation. The field *only* accepts JSON.

## Using GraphQL to Create the Custom Filter

In this example, you will create a custom GraphQL filter in Skylar One based on Device Groups. This filter will prevent Devices that belong to specific Device Groups from creating CIs in the ServiceNow CMDB.

To create a JSON filter for the *gql\_filter* field:

1. Go to the GraphiQL interface in Skylar One by typing the URL or IP address for Skylar One in a browser, adding */gql* to the end of the URL or IP address, and pressing **[Enter]**.
2. Create a GQL query that uses a search variable, such as the following example:

```
query DeviceFetch($search: DeviceSearch, $order: [ConnectionOrder],
  $first: Int) {
  devices(first: $first, search: $search, after: "", order: $order) {
    pageInfo {
      hasNextPage
      matchCount
    }
    edges {
      cursor
      device: node {
        ...
      }
    }
  }
}
```

3. After the query gives you the results you want, copy the resulting JSON variable definitions from the **Query Variables** section:

```
114 query DeviceFetch($search: DeviceSearch, $order: [ConnectionOrder], $first: Int) {
115   devices(first: $first, search: $search, after: "", order: $order) {
116     pageInfo {
117       hasNextPage
118       matchCount
119     }
120     edges {
121       cursor
122       device: node {
123         ...device

```

QUERY VARIABLES

```
1 {
2   "search": {"and":[{"deviceGroup": {"has": {"or": [{"name": {"contains": "foo"}}, {"name": {"contains": "bar"}}]}]}]},
3   "first": 100
4 }
```

4. Paste that JSON code into the **gql\_filter** portion of Skylar Automation and save the Skylar Automation application.

## JSON Code Template

Instead of using the GraphQL user interface in Skylar One, you can use the following JSON code as a template for filtering based on Device Group; simply paste the code into the **gql\_filter** field on the **Configuration** pane and edit the search criteria and group names as needed:

```
{
  "deviceGroup": {
    "has": {
      "or": [
        {
          "name": {
            "contains": "group1"
          }
        },
        {
          "name": {
            "contains": "group2"
          }
        }
      ]
    }
  }
}
```

Tips for customizing the JSON code:

- To add or remove additional Device Groups, copy or delete the following block of code and edit the group name as needed:

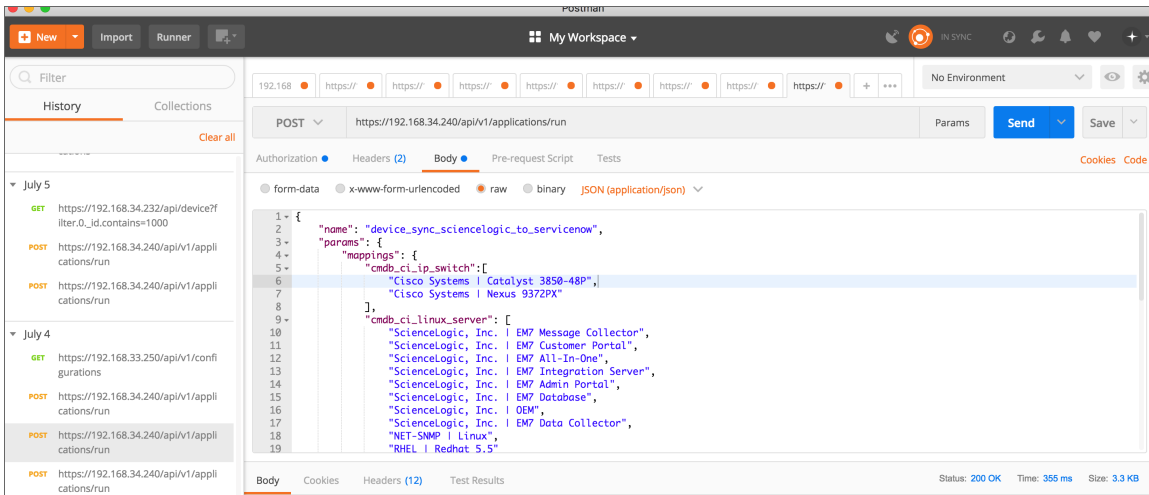
```
{
  "name": {
    "contains": "group2"
  }
}
```

- If you add additional name blocks, add a comma after each name block except for the last name block. If you delete a block, delete the preceding comma.
- You can change the "or" to an "and" if you want to block syncing for devices that are in more than one Device Group.
- The "contains" can be changed to "eq" to make the name an exact match. For example, "contains": "office" would match "The Office," "HQ Office," and "Office235."

## Adding Device Class Mappings

You can dynamically set the device mappings on a per-run basis using the API. You can also *persistently save device mappings with the API*. You can find these mappings in the **mappings** section of the **Configuration** pane for the "Sync Devices from Skylar One to ServiceNow" application.

The following image displays an example of using Postman to send the mapping data to Skylar Automation:



**NOTE:** This example only maps device classes to ServiceNow for VMware, Skylar One devices, and a few Cisco devices. If your environment contains other device classes, you must manually create the mappings.

To add device class mappings using Postman, POST the following JSON file to trigger the required applications in the Skylar Automation user interface to model Skylar One devices to ServiceNow:

```
{  
  
  "name": "device_sync_sciencelogic_to_servicenow",  
  
  "params": {  
  
    "mappings": {  
  
      "cmdb_ci_ip_switch": [  
  
        "Cisco Systems | Catalyst 3850-48P",  
  
        "Cisco Systems | Nexus 9372PX"  
  
      ],  
  
    },  
  
  },  
  
}
```

```
"cmdb_ci_linux_server": [  
  "ScienceLogic, Inc. | EM7 Message Collector",  
  "ScienceLogic, Inc. | EM7 Customer Portal",  
  "ScienceLogic, Inc. | EM7 All-In-One",  
  "ScienceLogic, Inc. | EM7 Integration Server",  
  "ScienceLogic, Inc. | EM7 Admin Portal",  
  "ScienceLogic, Inc. | EM7 Database",  
  "ScienceLogic, Inc. | OEM",  
  "ScienceLogic, Inc. | EM7 Data Collector",  
  "NET-SNMP | Linux",  
  "RHEL | Redhat 5.5"  
],  
"cmdb_ci_esx_resource_pool": ["VMware | Resource Pool"],  
"cmdb_ci_esx_server": [  
  "VMware | ESXi 5.1 w/HR",  
  "VMware | Host Server",  
  "VMware | ESX(i) 4.0",  
  "VMware | ESX(i) w/HR",  
  "VMware | ESX(i) 4.0 w/HR",  
  "VMware | ESX(i)",  
  "VMware | ESX(i) 4.1 w/HR",  
  "VMware | ESXi 5.1 w/HR",  
  "VMware | ESXi 5.0 w/HR",  
  "VMware | ESX(i) 4.1",
```

```

    "VMware | ESXi 5.1",
    "VMware | ESXi 5.0"
  ],
  "cmdb_ci_vcenter_datacenter": ["VMware | Datacenter"],
  "cmdb_ci_vcenter_datastore": ["VMware | Datastore", "VMware |
Datastore Cluser"],
  "cmdb_ci_vcenter_dv_port_group": ["VMware | Distributed Virtual
Portgroup"],
  "cmdb_ci_vcenter_dvs": ["VMware | Distributed Virtual Switch"],
  "cmdb_ci_vcenter_folder": ["VMware | Folder"],
  "cmdb_ci_vcenter_network": ["VMware | Network"],
  "cmdb_ci_vmware_instance": ["VMware | Virtual Machine"],
  "cmdb_ci_vcenter": ["VMware | vCenter", "Virtual Device | Windows
Services"],
  "cmdb_ci_vcenter_cluster": ["VMware | Cluster"]
},
"configuration": "template_snow_integration" #name your configuration
file
}
}

```

## Persistently Saving Device Class Mappings with the API

You can persistently save device class mappings using the API.

1. Use Postman or cURL to do a GET to load the "Sync Devices from Skylar One to ServiceNow" application:

```
GET Skylar Automation_hostname/api/v1/applications/device_sync_sciencelogic_to_servicenow
```

where *Skylar Automation\_hostname* is the IP address or URL for your Skylar Automation system.

**NOTE:** The response should contain the entire JSON output for the application.

2. Copy the entire JSON code and save it to a file named: "device\_sync\_sciencelogic\_to\_servicenow".
3. Open the file you created and locate the object with the "name": "mappings" property in the "app\_variables" list. The "value" property in this object specifies the mappings to use throughout the Skylar Automation applications:

```
"value": {  
  "cmdb_ci_appl_sharepoint": [  
    "VMware | Resource Pool"  
  ],  
  "cmdb_ci_esx_resource_pool": [  
    "VMware | Resource Pool"  
  ],  
  "cmdb_ci_esx_server": [  
    "VMware | ESXi 5.1 w/HR",  
    "VMware | Host Server",  
    "VMware | ESX(i) 4.0",  
    "VMware | ESX(i) w/HR",  
    "VMware | ESX(i) 4.0 w/HR",  
    "VMware | ESX(i)",  
    "VMware | ESX(i) 4.1 w/HR",  
    "VMware | ESXi 5.1 w/HR",  
    "VMware | ESXi 5.0 w/HR",  
    "VMware | ESX(i) 4.1",  
    "VMware | ESXi 5.1",  
    "VMware | ESXi 5.0"  
  ],  
  "cmdb_ci_hyper_v_network": [  
    "VMware | Resource Pool"  
  ],  
}
```

4. Modify the "value" property of the object to use the mappings you want to use.
5. Ensure that the mappings follow the same JSON data structure, or else the sync will not work:

```
{  
  "cmdb_ci_class": [  
    "ScienceLogic Dev Class | ScienceLogic subclass",  
    "Another Silo Dev Class | Another Silo subclass"  
  ]  
}
```

6. After you update the mappings, use the iscli tool to upload the updated application with your new settings. Type the following command at the command line:

```
iscli -uaf device_sync_sciencelogic_to_servicenow -H  
Skylar Automation_hostname -p password
```

where:


- `Skylar Automation_hostname` is the hostname or IP address of the Skylar Automation system.
- `password` is the password you use to log in to the Skylar Automation system.

## Checking for Missing Device Mappings

You can use the "Report: Identify Unmapped Devices Classes" Skylar Automation application to check whether any device mappings are missing in your Skylar Automation server.

This application pulls the class mappings from Device Sync and Attribute Sync and compares the mappings with the full list of device classes of discovered devices in Skylar One. The application generates a report on the **Reports** page that lists missing mappings, and if any device classes are unmapped, the application generates an event in the target Skylar One system.

To configure the "Report: Identify Unmapped Devices Classes" application:

1. If you do not have the "ServiceNow Base" PowerPack version 104 or later on your Skylar One system, search for and download the PowerPack from the **PowerPacks** page on the [ScienceLogic Support Site](#) (Skylar One > PowerPacks), then install the PowerPack in Skylar One.
2. In Skylar One, enable the "ServiceNow CMDB: Un-Mapped Device Classes" Event Policy from the "ServiceNow Base" PowerPack version 104 or later to trap the alert generated by this application.
3. In the Skylar Automation user interface, go to the **Configurations** page.
4. Click the **[Actions]** button (  ) for the configuration object you want to use with this application and select *Edit*. The **Configuration** pane for the object appears.
5. In the **Configuration Data Values** section, click **[Add Value]**. A new name-value line appears.

6. Add a configuration variable named **pf\_host** that has a value of the externally addressable IP or fully qualified domain name (FQDN) of the Skylar Automation cluster or instance.
7. Click **[Save]**.
8. Go to the **Applications** page and select the "Report: Identify Unmapped Devices Classes" application.
9. Click **[Configure]** to open the **Configuration** pane:

Report: Identify Unmapped Device Classes

Modify configuration and save. Show JSON Configs

Configuration  
docs\_config

sl1_hostname 10.64.68.17	sl1_user .....	sl1_password .....
\$(config.sl1_host)	\$(config.sl1_user)	
<b>pf_host</b> 10.2.11.86	read_timeout 20	query_retries 0
\$(config.pf_host)		

10. Complete the following fields, as needed:
  - **Configuration.** Select the configuration object you updated in steps 1-5 to align with this application. Make sure that the **pf\_host** value that is updated on the **Configuration** pane is accurate. You cannot edit fields that are populated by the configuration object. Required.
  - **read\_timeout.** Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 20 seconds.
  - **query\_retries.** Specify the number of times to retry this query if needed. The default chunk size is 0.
11. Click **[Save]**. The **Configuration** pane automatically closes after this message appears.
12. Click **[Run]** to run the application.

13. After the application runs, click the **[Reports]** button and select the relevant *report-missing-classes* report from the **Reports** page. The "missing\_classes" report appears:

The screenshot shows a web interface for a report titled "missing\_classes". The interface includes a navigation menu on the left, a header with "Report" and "isadmin", and a "Delete" button. The main content area displays a "Details" table and a "Missing Device Mappings" table.

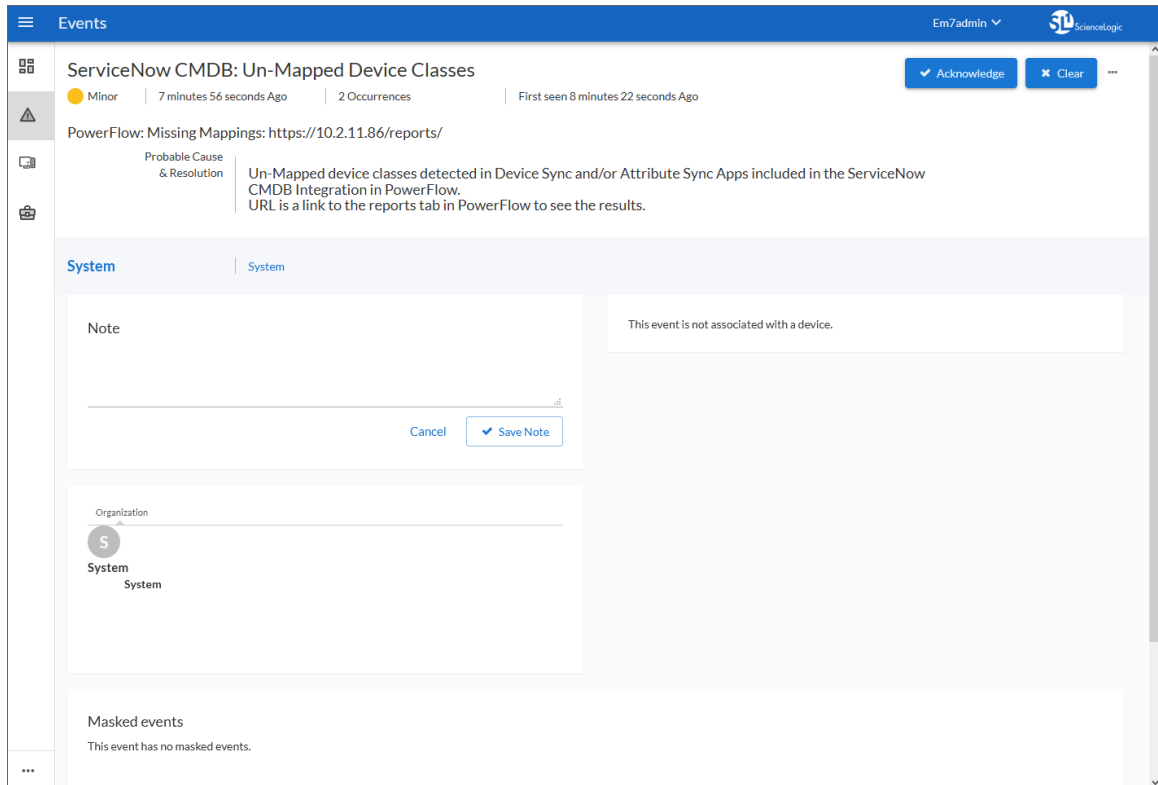
Details	
Application Name	missing_classes
Application ID	15-10-2020 15-49-17
Created (UTC -4)	Oct 15, 2020 11:49:17

Missing Device Mappings	
SL1 to ServiceNow Missing Mappings ↑	ServiceNow to SL1 Missing Mappings
AWS   Service	AWS   Service
Linux   Oracle Linux 7	Linux   Oracle Linux 7
Microsoft   Windows Server 2012 Domain Controller	Microsoft   Windows Server 2012 Domain Controller
Microsoft   Windows Server 2016 Domain Controller	Microsoft   Windows Server 2016 Domain Controller
Ping   ICMP	Ping   ICMP
ScienceLogic   SL1	ScienceLogic   SL1
ScienceLogic   SL1: Compute Cluster	ScienceLogic   SL1: Compute Cluster
ScienceLogic   SL1: Database Cluster	ScienceLogic   SL1: Database Cluster
ScienceLogic   SL1: Storage Cluster	ScienceLogic   SL1: Storage Cluster
ScienceLogic, Inc.   EM7 Admin Portal	ScienceLogic, Inc.   EM7 Admin Portal

**TIP:** The report displays missing mappings for Device Sync (Skylar One to ServiceNow) in the first column, and missing mappings for Attribute Sync (ServiceNow to Skylar One) in the second column.

Also, if any device classes are unmapped and you enabled the "ServiceNow CMDB: Un-Mapped Device Classes" Event Policy in step 2, Skylar One generates an event on the **Events** page in Skylar One:



## Device Attribute Mappings

The "Sync Devices from Skylar One to ServiceNow" application can also collect manufacturer and model attributes from asset records aligned with devices in Skylar One and sync that information with ServiceNow. You can access these mappings in the **Attribute Mappings** section of the **Configuration** pane for that application.

**NOTE:** These manufacturer and model attributes are different from custom attributes in Skylar One.

Skylar Automation only populates the manufacturer and model attributes if the values exist in ServiceNow CIs; Skylar Automation does not create new manufacturer values in ServiceNow. "Sync Devices from Skylar One to ServiceNow" application uses the **sys\_id** field as a reference when syncing manufacturer and model information between Skylar One and ServiceNow.

**WARNING:** Integer fields in ServiceNow have a maximum value of 2147483647. If a value exceeds that value, ServiceNow stores it as 2147483647. This is a MySQL limitation on the maximum value that can be stored in a signed integer variable.

**NOTE:** The "Sync Devices from Skylar One to ServiceNow" application only syncs assets that have a value populated in Skylar One. Skylar One automatically populates fields in an asset record when the record is properly configured. For more information on configuring assets in Skylar One, see [Asset Management](#).

Try one of the following options for the best approach for syncing manufacturer and model attributes, especially if your ServiceNow instance uses Domain Separation:

- Add the ServiceNow user that is listed in the **snow\_user** field on the **Configuration** pane of the "Sync Devices from Skylar One to ServiceNow" application to the Global Domain in ServiceNow.
- Schedule the "Sync Devices from Skylar One to ServiceNow" application to run with different ServiceNow users that belong to each ServiceNow domain, using the **Custom Parameters** field on the schedule window to overwrite the **snow\_user** value.

## Default Device Attribute Mappings

The "Sync Devices from Skylar One to ServiceNow" application contains a set of default device attribute mappings between Skylar One and ServiceNow. You can access these mappings in the **Attribute Mappings** section of the **Configuration** pane for that application.

The following table describes the default device attribute mappings:

Skylar One Device Attribute	ServiceNow CI Attribute
Description	"Description: {{device.device_category}}, Device Class: {{device.device_class}}": [ "short_description"  <b>NOTE:</b> This field requires a Jinja2 Template. For more information, see <a href="#">Using a Jinja2 Template</a> .
asset_tag	asset_tag
cpuCount	cpu_count
cpuMake	cpu_type
cpuSpeed	cpu_speed
dateAdded	first_discovered
diskSize	disk_space
dnsDomain	dns_domain
function	justification
hostname	fqdn, host_name
instance_uuid	account_id
ip	ip_address
manufacturer_sys_id	manufacturer
memory	ram
model_sys_id	model_id
name	name

Skylar One Device Attribute	ServiceNow CI Attribute
operatingSystem	os
purchaseDate	order_date
serial	serial_number
status	hardware_substatus
warrantyExpirationDate	warranty_expiration

## Skylar One Device Attributes Available for Syncing

In the **Attribute Mappings** section of the **Configuration** pane for the "Sync Devices from Skylar One to ServiceNow" application, you can also use the following Skylar One device attributes from Skylar One when syncing attributes with ServiceNow:

- arraySize
- asset\_id
- asset\_tag
- company\_sys\_id
- component\_unique\_id
- cpuCount
- cpuMake
- cpuSpeed
- dateAdded
- depreciationMethod
- depreciationSchedule
- device\_category
- device\_class
- diskCount
- diskSize
- dnsDomain
- dnsName
- domain\_sys\_id
- firmwareVersion
- floor
- function
- hostId
- hostname
- ip

- location
- make
- manufacturer\_sys\_id
- memory
- model
- modelNumber
- model\_sys\_id
- name
- operatingSystem
- org\_id
- org\_name
- owner
- panel
- parent\_device
- parent\_did
- plate
- punch
- purchaseCheck
- purchaseCost
- purchaseDate
- purchaseOrderNumber
- rack
- region
- rfid
- room
- serial
- serviceCheck
- serviceCost
- serviceDate
- serviceDescription
- serviceExpirationDate
- serviceOrderNumber
- servicePolicyNumber
- shelf
- sl1\_id
- sl1\_url

- snow\_ci\_class
- snow\_sys\_id
- status
- vitalAssetInformation
- vitalServiceInformation
- warrantyCheck
- warrantyCost
- warrantyDate
- warrantyDescription
- warrantyExpirationDate
- warrantyOrderNumber
- warrantyPolicyNumber
- zone

## Adding New Device Attributes to ServiceNow

You can also add one or more new attributes to ServiceNow that you can then sync with Skylar One.

To add an attribute in ServiceNow:

1. In ServiceNow, go to the **Tables** page (System Definition > Tables) and select the table to which you want to add a field for a new attribute.
2. From the **Table** page, click the **[New]** button to add a new field on the table. A new record appears.
3. From the **Type** drop-down list, select the data type you want to store, such as *String*. Depending on your selection, additional required fields display. For example, If you selected *String*, then **Column label** should contain the text you want to display in ServiceNow, and **Column name** is the exact column name used by Skylar Automation or the API.
4. Complete the required fields and any other fields as needed, and then click the **Submit** button. The field is added to ServiceNow.

## Configuring Customer CI Relation Overrides

When you are mapping Device Classes and attributes, you might find that Skylar One creates relationship mappings very differently than the way that ServiceNow creates relationships. As a result, ScienceLogic strongly recommends that you use the **customer\_ci\_relation\_overrides** field instead of using ServiceNow to set up those relationships.

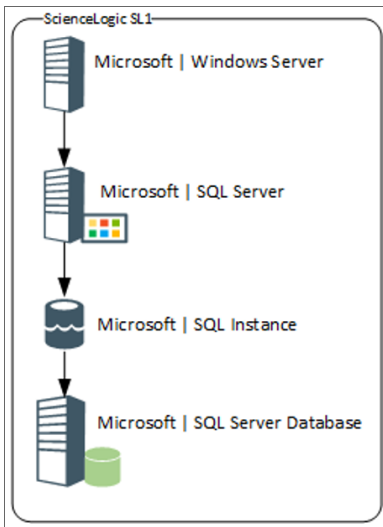
In the "Sync Devices from Skylar One to ServiceNow" Skylar Automation application, you can use the **customer\_ci\_relation\_overrides** field to override the existing relationship linking and directly control the link between Device Classes and attributes. The **customer\_ci\_relation\_overrides** field lets you build dynamic relationships rather than statically setting up relationships within ServiceNow.

**WARNING:** ScienceLogic does not support using ServiceNow to control and set up your device relationships.

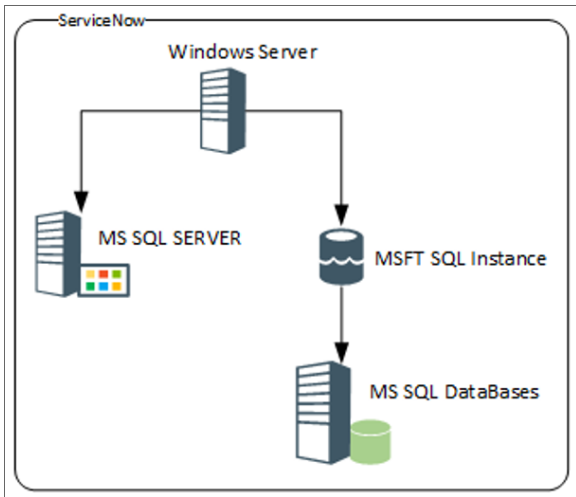
In addition, ScienceLogic strongly recommends that you use the default relationship overrides for VMware, which you can view by clicking **[Show JSON Configs]** from the **Configuration** pane for the "Sync Devices from Skylar One to ServiceNow" Skylar Automation application.

**CAUTION:** This mapping process is intended for advanced users that are familiar with how Skylar One and ServiceNow construct device relationships.

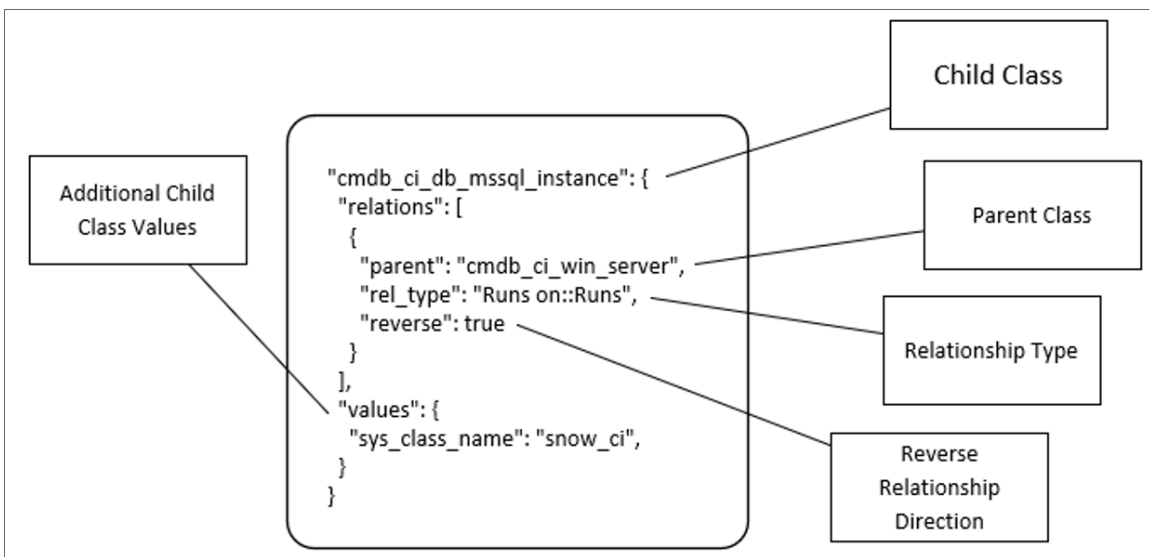
In the following example, the relationship structure in Skylar One is linear :



In ServiceNow, however, the structure is not as linear, and it requires an override (a manual link) between classes to make the relationship link required:



The following image shows the JSON structure formatting that is required for the *customer\_ci\_relation\_overrides* field:



The values in the *customer\_ci\_relation\_overrides* field supersede any of the values configured in the **mappings** section in the **Configuration** pane for the "Sync Device Classes from Skylar One to ServiceNow" Skylar Automation application.

**WARNING:** You must ensure that all classes in the relationship chain in Skylar One are mapped to classes in ServiceNow, or else the chain will break, and Skylar Automation will not correctly apply the overrides.

In the *customer\_ci\_relation\_overrides* field, you can string together multiple relationships as in the following example:


```
{
  "cmdb_ci_db_mssql_instance": {
    "relations": [
      {
        "parent": "cmdb_ci_win_server",
        "rel_type": "Runs on::Runs",
        "reverse": true
      }
    ],
    "values": {"sys_class_name": "snow_ci_class", "instance_name":
"name"}
  },
  "cmdb_ci_db_mssql_database": {
    "relations": [
      {
        "parent": "cmdb_ci_db_mssql_instance",
        "rel_type": "Contains::Contained by",
        "reverse": false
      }
    ],
    "values": {"sys_class_name": "snow_ci_class", "database": "name"}
  },
  "cmdb_ci_db_mssql_server": {
    "relations": [
      {
        "parent": "cmdb_ci_win_server",
        "rel_type": "Runs on::Runs",
        "reverse": true
      }
    ],
  },
}
```

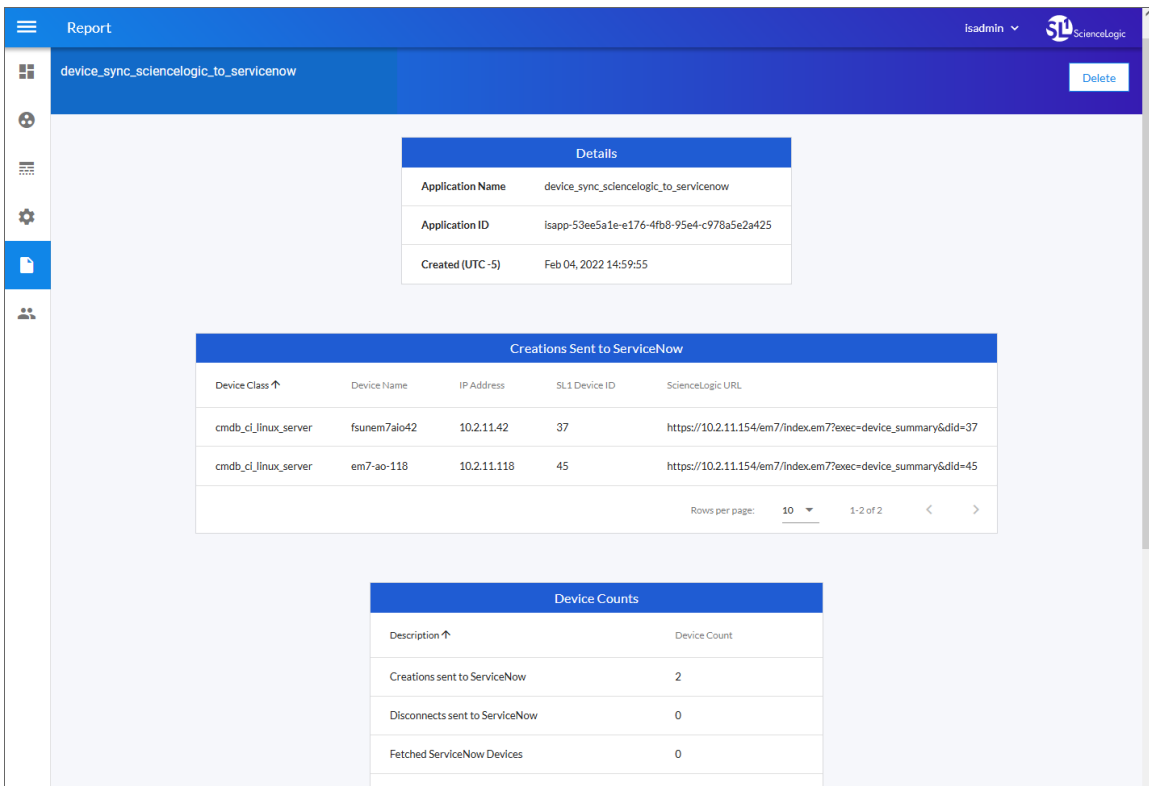
```

    "values": {"sys_class_name": "snow_ci_class", "instance_name":
"name"}
  }
}

```

## Viewing Reports for Device Sync

The "Sync Devices from Skylar One to ServiceNow" application lets you create a report about the devices that you sync with ServiceNow. The report displays on the **Reports** page (  ):



Details	
Application Name	device_sync_sciencelogic_to_servicenow
Application ID	isapp-53ee5a1e-e176-4fb8-95e4-c978a5e2a425
Created (UTC -5)	Feb 04, 2022 14:59:55

Creations Sent to ServiceNow				
Device Class ↑	Device Name	IP Address	SL1 Device ID	ScienceLogic URL
cmdb_ci_linux_server	fsunem7aio42	10.2.11.42	37	https://10.2.11.154/em7/index.em7?exec=device_summary&dld=37
cmdb_ci_linux_server	em7-ao-118	10.2.11.118	45	https://10.2.11.154/em7/index.em7?exec=device_summary&dld=45

Rows per page: 10 1-2 of 2

Device Counts	
Description ↑	Device Count
Creations sent to ServiceNow	2
Disconnects sent to ServiceNow	0
Fetches ServiceNow Devices	0

**TIP:** Enable the report by selecting the *generate\_report* option on the **Configuration** pane for the application.

When the relevant data is present in a device sync, the report displays the following data under the **Details** pane:

- **Creations Sent to ServiceNow.** Device information for devices in Skylar One that will be created in ServiceNow.

- **Device Counts.** Lists the number of created, disconnected, and updated devices, as well as the number of devices pulled from Skylar One and ServiceNow during the sync. If domain separation is on and there are devices removed due to domain separation errors, that count will be added to the table.
- **Deletions Sent to ServiceNow.** Device information for devices in Skylar One that will be deleted in ServiceNow.
- **Post CIs to ServiceNow.** Each instance of the "Post CIs to ServiceNow" application, listing creations, disconnects, errors, skips, and updates made in ServiceNow.
- **Updates Sent to ServiceNow.** Device information for devices in Skylar One to be updated in ServiceNow.

## Log Messages for the "Generate Required CI Relations for ServiceNow" Application

This section describes the different types of log messages you might see in the Step Log when you run the "Generate Required CI Relations for ServiceNow" application. Please note that this application is a report used by PowerFlow, and it does not send any data to ServiceNow.

The following message displays if there are devices in a device tree that do not currently have a CI class mapping assigned.

```
Warning: 2751 Relations with missing mappings detected. Please re-run app
with log level 10 to troubleshoot.
```

In this situation, the device tree cannot be built in ServiceNow. To address this issue, make sure that you have your entire technology tree mapped out in the *mappings* section of the "Sync Devices from Skylar One to ServiceNow" application or in the *mappings* section of the "Generate Required CI Relations for ServiceNow" application.

If you do a Custom Run of the "Generate Required CI Relations for ServiceNow" application in Debug mode (log level 10), the application will create a log that displays the parent and child class, CI, and device ID. For example:

```
Debug: Missing Mapping for Device. Parent: {"class": "VMware | Cluster",
"ci": None, "id": 76}, Child: {"class": "VMware | Host Server", ci:
"cmdb_ci_esx_server", id: 363 }
```

The following message appears if the GraphQL payloads had bad data for parent and or child devices:

```
Warning: 10 bad payloads received from Skylar One. Re-run app in debug to troubleshoot.
```

If you do a Custom Run the application in Debug mode, the application will create a log that displays these payloads.

The following message appears if all relations are mapped:

```
Flow: No missing relations found!
```

The following message appears if there is a parent/child relation between ServiceNow CI classes that does not currently exist in ServiceNow and is required to sync those devices:

```
Flow: Missing Relations: [{"parent": "cmdb_ci_vcenter_folder", "child": "cmdb_ci_esx_server"}, {"parent": "cmdb_ci_vcenter", "child": "cmdb_ci_vcenter_datacenter"}]
```

Refer to the labels in the log (above) to determine which CI class is the parent type and which is the child type. To address this issue, navigate to your ServiceNow instance and create the required service rules based on the recommendations in the **Step Log**.

The following message appears if the application encounters a list of relations that are required, but were successfully found in ServiceNow:

```
Info: Found Relations: [{"parent": "cmdb_ci_vcenter_folder", "child": "cmdb_ci_esx_server"}, {"parent": "cmdb_ci_vcenter", "child": "cmdb_ci_vcenter_datacenter"}]
```

This message lets you verify that your mappings and relations are configured correctly.

---

# Chapter

# 4

## Configuring Additional Syncs for the CMDB SyncPack

---

### Overview

This chapter describes the how to configure and run the various Skylar Automation applications contained in the "ServiceNow CMDB" SyncPack.

For more information about Device Sync, see [Configuring Device Sync for the CMDB SyncPack](#).

For more information about using configuration objects with this SyncPack, see [Creating and Aligning a Configuration Object](#).

This chapter covers the following topics:

<a href="#">Syncing Business Services</a> .....	60
<a href="#">Syncing CI Attributes from ServiceNow to Skylar One</a> .....	68
<a href="#">Syncing File Systems from Skylar One to ServiceNow</a> .....	76
<a href="#">Syncing Installed Software between Skylar One and ServiceNow</a> .....	80
<a href="#">Syncing Interfaces from Skylar One to ServiceNow</a> .....	82
<a href="#">Scheduling Skylar Automation Applications</a> .....	89
<a href="#">Log Messages for the "Generate Required CI Relations for ServiceNow" Application</a> .....	92

---

## Syncing Business Services

Depending on where you have your business services configured initially, you can use one of the two following methods to sync business services between Skylar One and ServiceNow:

- [Syncing Business Services from Skylar One to ServiceNow](#)
- [Syncing Business Services from ServiceNow to Skylar One](#)

To avoid duplicate services, you should only use *one* of the above processes in the Skylar Automation user interface. The Skylar Automation application you use depends on in which application (Skylar One or ServiceNow) you initially configured your services, before syncing.

For example, if you have Business Services, IT Services, and Devices Services set up in Skylar One, you would use the "Sync Business Services from Skylar One to ServiceNow" application, and Skylar One would be the "source of truth" after the sync.

### Syncing Business Services from Skylar One to ServiceNow

The ***Sync Business Services from Skylar One to ServiceNow*** application reads Business Services, IT Services, and Device Services from Skylar One and syncs them with business services in ServiceNow. This application creates and updates services, but it does not delete services.

The "Sync Business Services from Skylar One to ServiceNow" application does not currently support deleting or disconnecting services within ServiceNow. The application logs for the "Post to ServiceNow" step might contain information regarding disconnects, but you can ignore that information.

If a Device Service does not have a parent IT Service, the CMDB Group will be created, but the CI will not be created because of the way PowerFlow pulls each of those items. At the time of the Group creation, PowerFlow does not know if a service has a parent or not. At the time of the CI creation, PowerFlow does not directly pull Device Services; it only pulls Business Services and IT Services and their children.

**WARNING:** Skylar Automation only syncs business services that are aligned with devices that are already synced with ServiceNow. Before setting up business service sync, you must first [sync devices between Skylar One and ServiceNow](#).

To sync Skylar One business services with ServiceNow:

1. In ServiceNow, create an identifier rule for syncing services by typing "CI Identifiers" in the filter navigator and clicking **[New]** on the **Identifiers** page.
2. Complete the following fields:
  - **Name.** Type a relevant name for this rule, such as "Business Service".
  - **Applies to.** Select *cmdb\_ci\_service*.
  - **Independent.** Select this option.
3. Right-click the gray header and click *Save* to save the record.
4. On the **[Identifier Entries]** tab, click **[New]** and add the relevant values from the **Criterion attributes** field for this business service, such as *name*, *service\_classification* and *correlation\_id*.

5. Click **[Submit]**.
6. Repeat steps 4-5 for each identifier you want to add.
7. Go to the **Applications** page of the Skylar Automation user interface and run the "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application. This application reads all existing Skylar One Device Classes, Organizations, ServiceNow CIs, and Companies and writes them to a cache.

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

8. When that application completes, go to the **Applications** page and run the "Sync Devices from Skylar One to ServiceNow" application to sync devices and their properties and relationships from Skylar One to ServiceNow.
9. When that application completes, go to the **Applications** page and select the "Sync Business Services from Skylar One to ServiceNow" application.

- Click **[Configure]** to open the **Configuration** pane:

- Complete the following fields, as needed:

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

**NOTE:** The **region** field is populated by the configuration object you aligned with this application. 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 the application should wait for a response before timing out. The default is 20 seconds.
- **sl1\_chunk\_size.** Specify the number of services to include in each chunk sent to ServiceNow when you run this application. The default chunk size is 500.

- **snow\_request\_limit.** Specify the number of CIs fetched from ServiceNow in each request. The default is 500 objects per chunk.
  - **sl1\_url\_override.** Specify a URL that is different from the standard Skylar One URL that gets sent to the ServiceNow CI record. Optional.
  - **business\_service\_ci\_class.** Specify the ServiceNow CI Class for Business Services. The default is `cmdb_ci_service`.
  - **it\_service\_ci\_class.** Specify the ServiceNow CI Class for IT Services. The default is `cmdb_ci_service_technical`.
  - **business\_service\_classification, it\_service\_classification, and device\_service\_classification.** Use these fields to update the default service classifications. Optional.
  - **relationship\_type.** Specify the relationship type string to use between services. The default is `Depends on::Used by`.
  - **discovery\_source.** Specify the ServiceNow Discovery source. The default is "Other Automated".
  - **verify\_snow\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
  - **snow\_batch\_size.** Specify the total number of CIs being sent to ServiceNow in each triggered application. The default is 150 objects per batch.
  - **snow\_chunk\_size.** Specify the number of CI objects to send in each chunk or in each sub-payload with a batch (specified in the `snow_batch_size` field). The default is 150 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.
  - **Domain\_Separation.** Select this option if your ServiceNow environment is *domain-separated*, where the data, processes, and administrative tasks have been organized into logical groupings called *domains*. If your ServiceNow instance is domain-separated, the user listed in the `snow_user` field must be a member of the top domain and have access to *all* of the domains you intend to integrate. Also, ServiceNow should be the "source of truth" for organizations if your environment is domain-separated.
  - **Simulation\_Mode.** Select this option if you want to perform a simulated run of this application to show you the potential results of that run.
12. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.
  13. Click **[Save]**. The **Configuration** pane automatically closes.
  14. Click **[Run]** to run the "Sync Business Services from Skylar One to ServiceNow" application.

# Syncing Business Services from ServiceNow to Skylar One

The **Sync Business Services from ServiceNow to Skylar One** application syncs services that were defined in ServiceNow with Business Services in Skylar One. When you sync services from Service Now to Skylar One, Skylar Automation recreates the service structure in Skylar One, where you can see the relationships between the service components, the application components, and the infrastructure components. You can also sync any asset fields on the service using an attribute mapping on the **Configuration** pane.

## Aspects of Syncing Business Services from ServiceNow to Skylar One

- This process syncs the business service structure or map from ServiceNow, and the device services created in Skylar One as the result of the sync can contain multiple devices.
- An aggregate service in Skylar One will only be created with services as children and a device service in Skylar One will only be created with devices as children.
- Depending on the actions taken in ServiceNow, an aggregate service can be inserted above or a device service may be inserted below certain services when the aggregate service is created in Skylar One.
- This process creates device services in Skylar One, but it does not create devices in Skylar One.
- Services are not merged in this process, so when you use the "Sync Business Services from ServiceNow to Skylar One" application, the "source of truth" will be set to ServiceNow, not Skylar One.
- If a ServiceNow service has no child services or child devices, that service will not be created in Skylar One.
- If a ServiceNow service was previously synced to Skylar One when it was not empty, but it is now an empty service ServiceNow, the service is deleted in Skylar One.

The following tables in ServiceNow are treated as services in a sync to Skylar One:

- cmdb\_ci\_service
- cmdb\_ci\_service\_technical
- cmdb\_ci\_query\_based\_service
- cmdb\_ci\_service\_discovered
- cmdb\_ci\_service\_auto
- cmdb\_ci\_service\_group
- service\_offering
- cmdb\_ci\_service\_manual
- cmdb\_ci\_service\_calculated

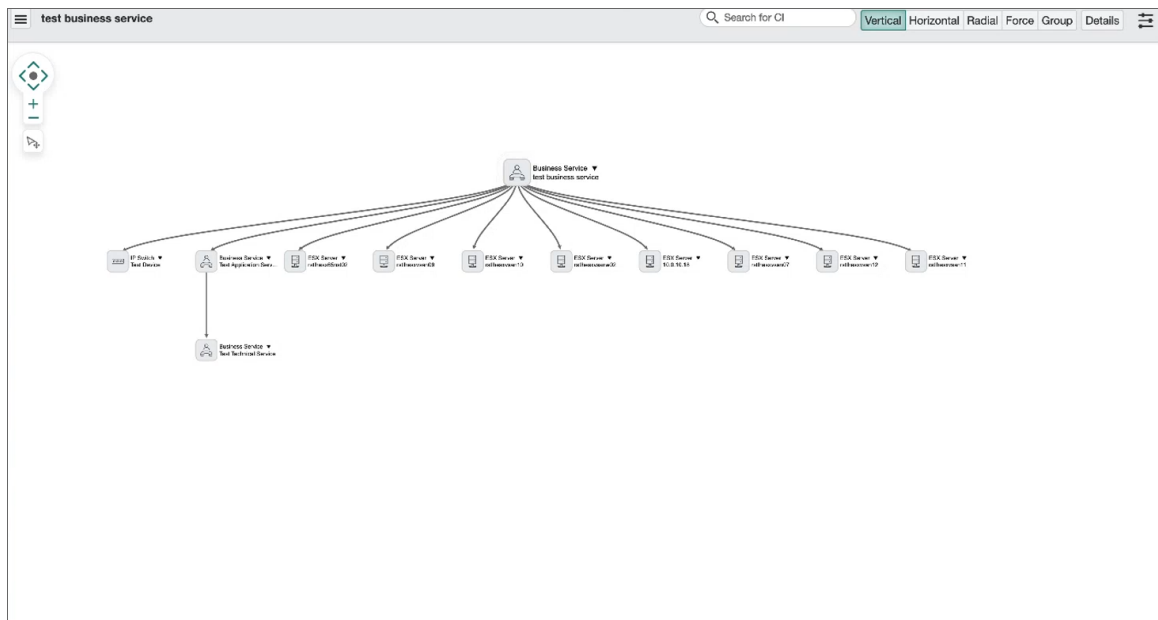
All other tables will be treated as devices.

**NOTE:** The `cmdb_ci_query_based_service` table contains the device services within ServiceNow. For any other service with a CI as a direct child, a device service will be inserted in between this service and the child CI in Skylar One.

**WARNING:** Syncing services from ServiceNow to Skylar One requires Skylar One 10.2.0 or later.

To sync ServiceNow services with Skylar One Business Services:

1. In ServiceNow, configure your business service structure or map so it is ready to be synced to Skylar One:



2. For each service that you want to sync with Skylar One:
  - Ensure that the *Skylar One Region* field in ServiceNow matches the *region* field in Skylar Automation.
  - Set the *Skylar One Monitored* flag to *True*.
3. Go to the **Applications** page of the Skylar Automation user interface and run the "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application. This application reads all existing Skylar One Device Classes, Organizations, ServiceNow CIs, and Companies and writes them to a cache.

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDb" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

4. When that application completes, go to the **Applications** page and run the "Sync Devices from Skylar One to ServiceNow" application to enable Skylar Automation to use the mappings and additional attribute options from Device Sync.
5. When that application completes, go to the **Applications** page and select the "Sync Business Services from ServiceNow to Skylar One" application.
6. Click **[Configure]** (⚙️) to open the **Configuration** pane:

Sync Business Services from ServiceNow to SL1
✕

Modify configuration and save. Show JSON Configs

Configuration  
\_bserv\_sync\_to\_43

<p><small>sl1_hostname</small> 10.2.11.43</p> <p><small>\$(config.sl1_host)</small></p>	<p><small>snw_hostname</small> ven01770.service-now.com</p> <p><small>\$(config.snw_host)</small></p>	<p><small>sl1_user</small> em7admin</p> <p><small>\$(config.sl1_user)</small></p>
<p><small>snw_user</small> is4user1</p> <p><small>\$(config.snw_user)</small></p>	<p><small>sl1_password</small> ●●●●●●●●●●●●●●●●●●●●</p>	<p><small>snw_password</small> ●●●●●●●●●●●●●●●●●●●●</p>
<p><small>region</small> unique_id</p> <p><small>\$(config.region)</small></p>	<p><small>read_timeout</small> 20</p>	

attribute\_map

Select/type an option

+

maps to:

Search options

+

Add Mapping

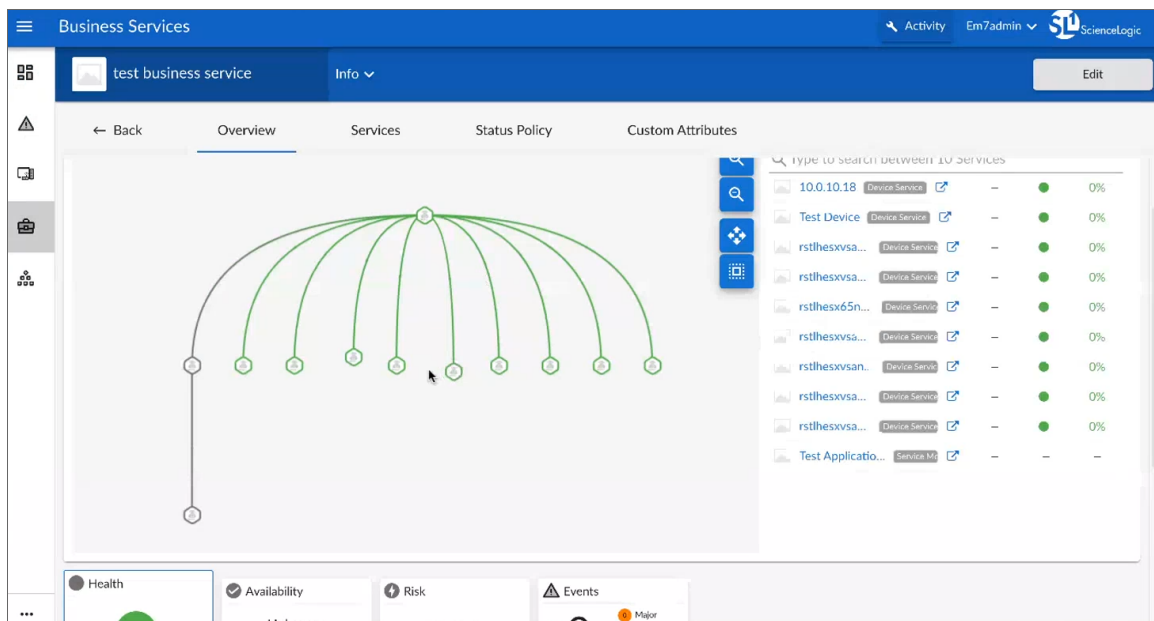
Save

7. Complete the following fields, as needed:

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

**NOTE:** The **region** field is populated by the configuration object you aligned with this application. 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 the application should wait for a response before timing out. The default is 20 seconds.
  - **verify\_snw\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
  - **attribute\_map.** In this section, click **[Add Mapping]** if you want to create a mapping for any other asset fields that you want to sync between Skylar One (the first column) and ServiceNow (the second column).
8. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.
9. Click **[Save]**. The **Configuration** pane automatically closes.
10. Click **[Run]** to run the application. When the application completes, the ServiceNow business services will be available on the **Business Services** page of Skylar One:



# Syncing CI Attributes from ServiceNow to Skylar One

The "Sync CI Attributes from ServiceNow to Skylar One" application imports CI attributes from ServiceNow to the relevant asset and attribute fields in Skylar One. The CI Sync supports assets, asset configuration, asset maintenance, location, production statuses, and custom attributes.

The "Sync CI Attributes from ServiceNow to Skylar One" application can sync the display value and **sys\_id** of **Reference** fields, such as location, as well as the value and label of **Choice List** fields, such as operational\_status. These values can be accessed by appending **\_label** to the desired field name.

## Reference Example:

```
"location": "240f6630db993300dc44f00fbf96196f"
```

```
"location_label": "Corporate Headquarters"
```

## Choice List Example:

```
"operational_status": "1",
```

```
"operational_status_label": "Operational",
```

The following image shows the **Location** table, and the **Display** column shows the **Name** marked as **true**. Only one field on the table can be marked as **true**, and that is the field that will be returned to Skylar Automation :

Column label	Type	Reference	Max length	Default value	Display
Row	Integer	(empty)	40		true
Set	Reference	Import Set	32		false
State	String	(empty)	40	pending	false
Comment	String	(empty)	1,000		false
Updates	Integer	(empty)	40		false
Error	Reference	Import Set Row Error	32		false
Target record	Document ID	(empty)	32		false
Target table	Table Name	(empty)	80		false
Transform Map	Reference	Table Transform Map	32		false
Updated by	String	(empty)	40		false
Updated	Date/Time	(empty)	40		false
City	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false
Country	String	(empty)	40		false
Name	String	(empty)	40		false
State	String	(empty)	40		false
Street	String	(empty)	40		false
Sys ID	Sys ID (GUID)	(empty)	32		false

**NOTE:** When this application runs, if no mappings are provided, Skylar Automation queries the "Sync Devices from Skylar One to ServiceNow" application and uses the mappings from that application.

To sync CI attributes from ServiceNow to Skylar One:

1. Go to the **Applications** page of the Skylar Automation user interface and run the "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application. This application reads all existing Skylar One Device Classes, Organizations, ServiceNow CIs, and Companies and writes them to a cache.

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

2. When that application completes, go to the **Applications** page and run the "Sync Devices from Skylar One to ServiceNow" application to enable Skylar Automation to use the mappings and additional attribute options from Device Sync.
3. When that application completes, go to the **Applications** page and run the "Sync CI Attributes from ServiceNow to Skylar One" application.

- Click **[Configure]**. The **Configuration** pane appears:

- 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. 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
- chunk\_size.** Specify the number of devices to include in each chunk sent to ServiceNow when you run this application. The default chunk size is 500 devices.

- **Include\_Orgs.** If you want to include Skylar One Organizations in the sync, add the Organization IDs from Skylar One in this field, separated by commas. Leave this field empty to sync all Skylar One Organizations. This option filters on the ServiceNow CI sync as well as the Skylar One Device sync.
- **Include\_CUGs.** If you want to include Skylar One Collector Groups (CUGs) in the 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.
- **snow\_request\_limit.** Specify the number of CIs fetched from ServiceNow in each request. The default is 500 objects per 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.
- **verify\_snow\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
- **sync\_empty\_fields.** Select this option if you want to include empty fields on ServiceNow CIs when you run this application. This option is disabled by default.

**NOTE:** Skylar One does not return extended custom attributes for devices if no value is present. As a result, if the value for the custom attribute is empty in ServiceNow, Skylar Automation will continually send updates for those devices, because the empty value written to is not returned in subsequent runs of the application. When the `sync_empty_fields` option is selected, the application sends a 0 or "", depending on the attribute type for the custom attributes in the mappings for all devices.

- **snow\_attributes\_created.** Select this option to show that custom attributes for ServiceNow `sys_id` and `ci_class` already existing in Skylar One.
- **retry\_jitter.** When selected, instead of using a defined interval between retries, Skylar Automation will retry the step execution at random intervals. The default is unselected.
- **retry\_backoff.** When selected, instead of using a defined interval between retries, Skylar Automation will incrementally increase the interval between retries. The default is unselected.
- **gql\_filter.** Specify a custom GraphQL filter to apply to the devices in the sync. An example of a Device Search GraphQL query is: `{"name": {"doesNotBeginWith": "jc-is-ma"}}`

6. In the **Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit the mappings between ServiceNow CI classes and Skylar One device classes. You can map a single ServiceNow CI class with multiple Skylar One device classes.

If no mappings are provided in this section, the application will only pull CI classes provided in the "Sync Devices from Skylar One to ServiceNow" application. If you add mappings in this section, only CI classes that were included in this section will be returned from ServiceNow.

For more information about how to view, edit, and create mappings, see [Editing Mappings in a Skylar Automation Application](#). This topic also covers how to use the **Company Mapping Override** and the **Attribute Mappings** sections, below.

7. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.

8. In the **Attribute Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit mappings for any other custom attributes you want to sync between Skylar One (the first column) and ServiceNow (the second column). For more information, including lists of default and available device attribute mappings, see [Device Attribute Mappings](#).

To sync a user-defined field on the ServiceNow CI record to the device notes table in Skylar One, create a custom attribute mapping for that field in this section. This option works best if you only want a single value synced over so that value can remain in the notes.

**TIP:** For the **Attribute Mappings** section, you can use a Jinja2 Template for device attribute fields on the Skylar One side (the left column). For more information, see [Using a Jinja2 Template](#).

**NOTE:** When an attribute value is "0" in Skylar One, the corresponding field in ServiceNow might display as empty.

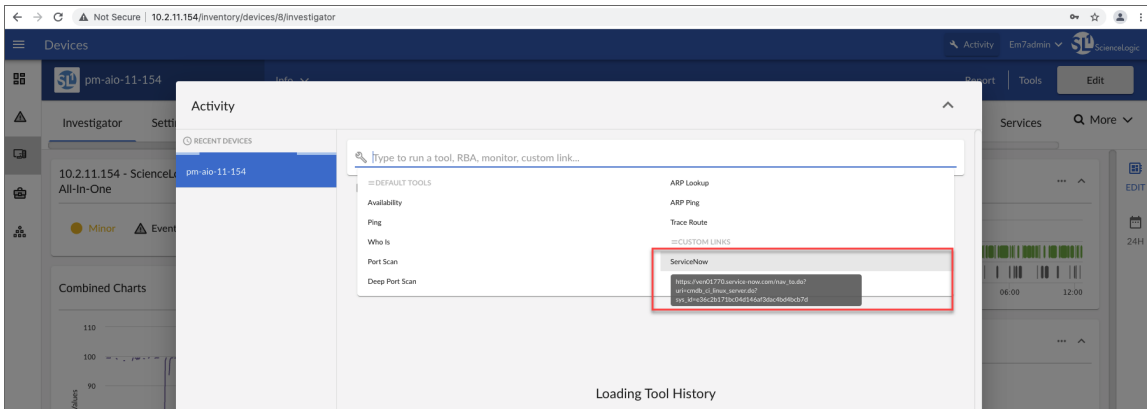
9. Click **[Save]**. The **Configuration** pane automatically closes.
10. Click **[Run]** to run the "Sync CI Attributes from ServiceNow to Skylar One" application.

## Creating a Link in a Skylar One Device to a ServiceNow CI

To help you manage your Skylar One devices and their corresponding ServiceNow CIs, you can create a link in a Skylar One Device that goes directly to the synced CI in ServiceNow.

The "Create Custom Attributes and ServiceNow Custom Link in Skylar One" PowerFlow application creates **servicenow\_sys\_id** and **servicenow\_ci\_class** custom device attributes in Skylar One, and then Skylar One uses those two attributes to create a custom link in Skylar One titled *ServiceNow* that redirects to the ServiceNow CI.

In Skylar One, the link appears as a custom link on the **Tools** menu for the corresponding device:



**NOTE:** The application only creates the custom attributes if the attributes do not already exist. Additionally, if the custom link functionality is disabled by default on Skylar One, the application will enable the feature before creating the necessary attributes and link.

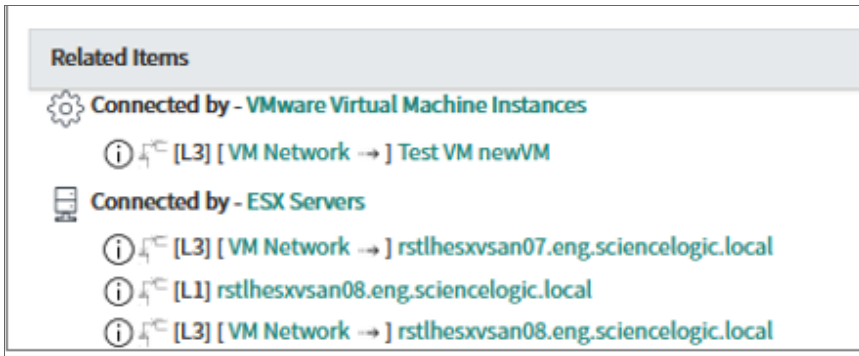
To create a link for the Skylar One device:

1. In the PowerFlow user interface, go to the **Applications** page and open the "Sync CI Attributes from ServiceNow to Skylar One to ServiceNow" application.
2. Click the **[Configure]** button. The **Configuration** pane appears.
3. Enable the **snow\_attributes\_created** toggle and click **[Save]**.
4. Click **[Run]** to run the application.
5. On the **Applications** page, open the "Create Custom Attributes and ServiceNow Custom Link in Skylar One" PowerFlow application, align a configuration object if needed, and click **[Run]** to run the application.

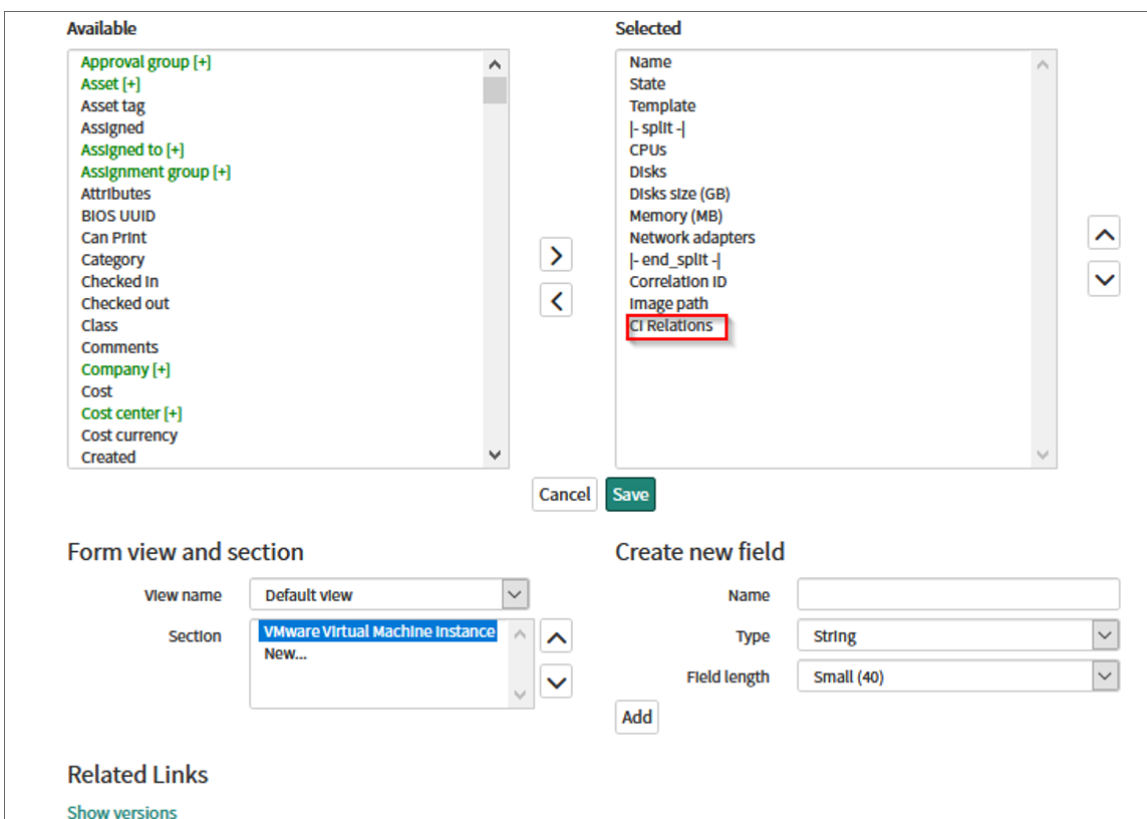
**NOTE:** You only need to run the "Create Custom Attributes and ServiceNow Custom Link in Skylar One" application once.

## Adding Related Items and Lists to the CI Record in ServiceNow

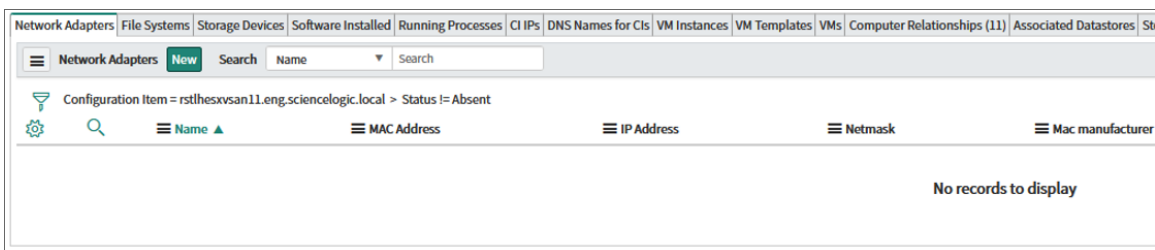
In ServiceNow you can add Related Items and Related Lists to CI records so you can see all related records in ServiceNow. The following image shows the **Related Items** pane on a CI record for a VMware Virtual Machine Instance:



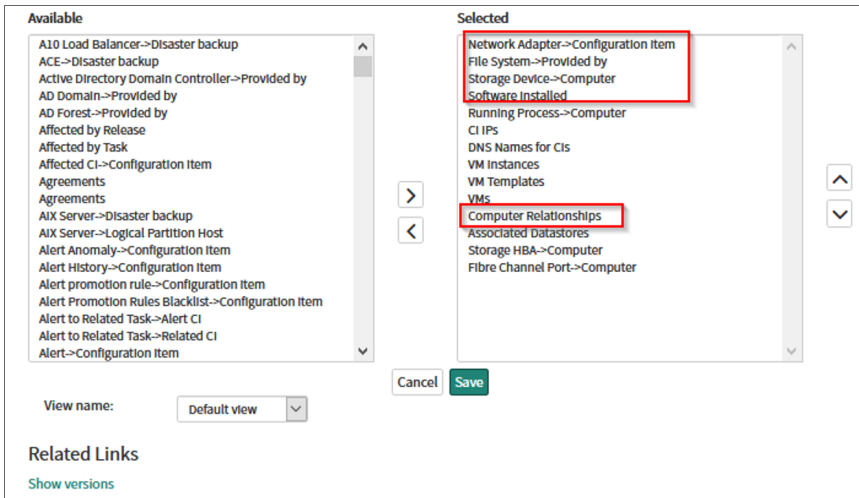
The following image shows how to add Related Items for a VMware Virtual Machine instance by selecting the *CI relations* option.



The following image shows the related lists for network adapters and files systems:



The items highlighted by the red boxes below are the most commonly used items:



For more information, see the ServiceNow Documentation at [https://docs.servicenow.com/bundle/paris-platform-user-interface/page/use/using-forms/concept/c\\_RelatedLists.html](https://docs.servicenow.com/bundle/paris-platform-user-interface/page/use/using-forms/concept/c_RelatedLists.html).

## Deleting Devices

The "Delete Devices from Skylar One" application lets you delete devices in a specific Virtual Collector Group (VCG) if those devices have not been modified in Skylar One for a specified time, such as one day or five days. You can update this time in the *max\_age* configuration value, which is described below.

To delete devices from a Skylar One Virtual Collector Group:

1. In the Skylar Automation user interface, run the "Sync Service Requests from ServiceNow to Skylar One" application to pull a list of decommissioned devices that you no longer want to monitor.

**IMPORTANT:** The "Sync Service Requests from ServiceNow to Skylar One" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Service Catalog" SyncPack version 2.0.0. In order to run the application, you must have version 2.0.0 of the "ServiceNow Service Catalog" SyncPack installed.

1. On the **Applications** page, select the "Delete Devices from Skylar One" application and click **[Configure]** on the application detail page. The **Configuration** page appears:

Delete Devices from SL1

Modify configuration and save.

Configuration  
ven01770

sl1_hostname 10.2.11.154	sl1_db_host \${config.sl1_host}	sl1_user em7admin
sl1_password ●●●●●●●●●●●●●●●●	sl1_db_user root	sl1_db_password ●●●●●●●●●●●●●●●●
max_age 0	target_vcug 2	read_timeout 20

3. Complete the following fields, as needed:
  - **Configuration.** Select the relevant configuration object to align with this application. You cannot edit fields that are populated by the configuration object. Required.
  - **read\_timeout.** Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 20 seconds.
  - **max\_age.** Specify how long (in days) that you want to keep the devices in the VCUG before deleting the devices. The default is 0 days. If this setting is 0, all devices in the VCUG will be deleted as soon as this application runs. If this setting is null, the application will fail. If all device children are in the same VCUG, the application will delete the target device and all of its children.
  - **target\_vcug.** Specify the ID of the Skylar One Virtual Collection Group (VCUG) you created to hold the devices on the **Collector Group Settings** page (System > Settings > Collector Groups). Set this value to -1 if you want this applications to use the **target\_vcug** value from the "Sync Service Requests from ServiceNow to Skylar One" application.

**WARNING:** If you specify a value to **target\_vcug** here, the "Delete Devices from Skylar One" application will use that value instead of the **target\_vcug** value from the "Sync Service Requests from ServiceNow to Skylar One" application.

4. Click **[Save]**. The **Configuration** pane automatically closes.
5. Click **[Run]** to run the application.

---

## Syncing File Systems from Skylar One to ServiceNow

You can map and sync file systems in much the same way you sync devices between Skylar One and ServiceNow. The "Sync File Systems from Skylar One to ServiceNow" application reads file systems discovered in Skylar One and then maps them to a parent CI record in ServiceNow.

For a file system to be synced from Skylar One to ServiceNow, the class of the file system's parent device must be included in the mappings parameters for the "Sync Devices from Skylar One to ServiceNow" application. File systems with parent CIs that have **className** values of **cmdb\_ci\_ip\_switch** or **cmdb\_ci\_ip\_router** will be sent to ServiceNow, but by default they will not be created in ServiceNow due to missing relationship information.

**WARNING:** Skylar Automation only syncs file systems that are aligned with devices that are already synced with ServiceNow. Before setting up file system sync, you must first [sync devices between Skylar One and ServiceNow](#).

To sync Skylar One file systems with ServiceNow:

1. In the Skylar Automation user interface, go to the **Applications** page and run the "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application to read all existing Skylar One Device Classes, Organizations, ServiceNow CIs, and Companies and write them to a cache.

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

2. When that application completes, go to the **Applications** page and select the "Sync File Systems from Skylar One to ServiceNow" application.
3. Click **[Configure]** to open the **Configuration** pane.
4. Complete the following fields, as needed:
  - **Configuration.** Select the relevant configuration object to align with this application. You cannot edit fields that are populated by the configuration object. Required.
  - **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\_request\_limit.** Specify the number of objects fetched from ServiceNow in each request. The default is 500 objects per chunk.
  - **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.
  - **disconnect\_field.** Acts as the boolean field on the Configuration Items table to determine "True/False" for the Skylar One monitoring status.
  - **snow\_batch\_size.** Specify the total number of objects being sent to ServiceNow in each triggered application. The default is 150 objects per batch.
  - **discovery\_source.** Specify the ServiceNow Discovery source. The default is "Other Automated".

- ***snow\_chunk\_size***. Specify the number of objects to send in each chunk or in each sub-payload with a batch (specified in the ***snow\_batch\_size*** field). The default is 150 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.
- ***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.
- ***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 these ***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.

- ***sync\_unsupported\_file\_systems***. Toggling this on will allow file systems that have a parent device class of **Network.Switch** or **Network.Router** to sync when the application is run.
- ***Domain\_Separation***. Select this option if your ServiceNow environment is *domain-separated*, where the data, processes, and administrative tasks have been organized into logical groupings called *domains*. If your ServiceNow instance is domain-separated, the user listed in the ***snow\_user*** field must be a member of the top domain and have access to *all* of the domains you intend to integrate. Also, ServiceNow should be the "source of truth" for organizations if your environment is domain-separated.
- ***change\_fs\_organizations***. Toggle on (blue) this option to allow file systems to change companies with an Interface Sync if the companies are within the same domain in ServiceNow. If you change the Organization/Company and another field, but do not enable this parameter, Skylar Automation will not apply any changes to the file system. By default, this option is not selected.

- **drop\_company.** Toggle on (blue) this option to remove the `sys_id` in existing companies from the sync. This will prevent the "Company" field from sending to ServiceNow. This option has no effect if the **domain\_separation** parameter is enabled for this application. By default, this option is not selected.

**NOTE:** If you select this option, the ServiceNow company **sys\_id** for the file system and the Skylar One organization **crm\_id** for the Skylar One device will not be compared when determining if an update has occurred. Do not select this option if you selected the **Domain\_Separation** option, as ServiceNow requires PowerFlow to send the "company" field in domain-separated environments. Also, do not select this option if you selected the **change\_interface\_organizations** option, as changing the company for a file system in ServiceNow requires the "company" field to be sent.

- **verify\_snow\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
- **sl1\_org\_filter\_only.** Toggle on (blue) this option to apply the organization filter to only the Skylar One side of the data pull.

**CAUTION:** Enabling this toggle will only apply organization filtering to the Skylar One side of the data pulls, while all file systems for the provided region in ServiceNow will be returned. Be cautious using this option, as enabling it may cause unwanted disconnects in ServiceNow.

- **generate\_report.** Select this option to create a "Sync File Systems from Skylar One to ServiceNow" report. This report contains counts of the number of file systems gathered from Skylar One and ServiceNow, the number of unchanged file systems, and the number of creations, updates, and disconnects sent to ServiceNow. Detailed file system information for the file systems sent to ServiceNow is also displayed. Finally, the report contains file systems that are in an unsupported format. The reports are available on the **Reports** page of the Skylar Automation user interface.

**NOTE:** Due to the differences between how Skylar One and ServiceNow discover file systems, only certain formats of file systems contained in the Skylar One database will be synced to ServiceNow. Only file systems where the name and the mount point can be determined from the name database field stored in the Skylar One will be synced.

- **Simulation\_Mode.** Select this option if you want to perform a simulated run of this application to show you the potential results of that run.
- **retry\_jitter.** When selected, instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. By default, this option is not selected.

- **retry\_backoff**. When selected, instead of using a defined interval between retries, Skylar Automation will incrementally increase the interval between retries. By default, this option is not selected.
  - **disconnect\_type**. Specify the disconnect method for disconnecting Skylar One entities from ServiceNow. Enter either *flip\_boolean\_field* or *remove\_sources\_entry* to determine the behavior for disconnects.
5. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.
  6. In the **Attribute Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit the mappings for any other custom device attributes you want to sync between Skylar One (the first column) and ServiceNow (the second column).

**NOTE:** When an attribute value is "0" in Skylar One, the corresponding field in ServiceNow might display as empty.

7. Click **[Save]**. The **Configuration** pane closes.
8. Click **[Run]** to run the application.

---

## Syncing Installed Software between Skylar One and ServiceNow

You can use the following application to sync your installed software assets between and ServiceNow:

- "Sync Installed Software from Skylar One to ServiceNow". Reads all software packages from Skylar One and creates new CIs in ServiceNow. Reads all available software packages from Skylar One and the devices aligned to that software by region and syncs them with ServiceNow.

The application does not currently support domain separation.

**NOTE:** The Software Asset Management (SAM) application in ServiceNow is not supported with the current level of installed software data acquired with Skylar One. As a result, syncing installed software data with ServiceNow Discovery and other Software Asset Management software is not currently supported.

To sync installed software between Skylar One and ServiceNow:

1. Make sure that you have recently run the "Sync Devices from Skylar One to ServiceNow" application to populate the device cache.
2. In the Skylar Automation user interface, go to the **Applications** page and select the "Sync Installed Software from Skylar One to ServiceNow" application.

5. Click **[Configure]** (⚙️) to open the **Configuration** pane:

Field Name	Value	Status
sl1_db_host	`\${config.sl1_host}`	Locked
snw_hostname	ven01770.service-now.com	Locked
snw_user	is4user1	Locked
`\${config.sl1_db_host}`	`\${config.snw_host}`	Locked
`\${config.snw_user}`	`\${config.snw_user}`	Locked
snw_password	.....	Locked
sl1_db_user	root	Locked
sl1_db_password	.....	Locked
`\${config.sl1_db_user}`	`\${config.sl1_db_user}`	Locked
region		Locked
chunk_size	500	Locked
read_timeout	20	Locked
del_test		Locked
`\${config.region}`		Locked

9. Complete the following fields, as needed:

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

**NOTE:** The **region** field is populated by the configuration object you aligned with this application. 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 the application should wait for a response before timing out. The default is 20 seconds.
  - **snw\_request\_limit.** Specify the number of objects fetched from ServiceNow in each request. The default is 500 objects per chunk.
  - **snw\_chunk\_size.** Specify the number of objects to send in each chunk. The default is 150 objects per chunk.
  - **verify\_snw\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
  - **region.** The region value is populated by the configuration object you selected. The region value must match the value in the **SL\_Region** field in ServiceNow. If you need to update this value, you will need to define the **region** variable in the configuration that is aligned with this application, or align a different configuration that has the correct region value.
  - **chunk\_size.** Specify the number of services to include in each chunk sent to ServiceNow when you run this application. The default chunk size is 500.
10. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
11. Click **[Run]** (▶️) to run the application.

---

# Syncing Interfaces from Skylar One to ServiceNow

You can map and sync network interfaces in much the same way you sync devices between Skylar One and ServiceNow. The "Sync Interfaces from Skylar One to ServiceNow" application collects interface data from ServiceNow and Skylar One and runs multiple CI syncs for each interface to be synced.

Version 3.5.0 and later of this SyncPack requires Skylar One version 11.2.0 or later.

**IMPORTANT:** The speed of ServiceNow processing is reduced by ServiceNow errors generated when consuming payloads. If you experience slow processing or "maximum execution time exceeded" messages, you must review and address any ServiceNow errors reported to resolve. If there are significant ServiceNow errors in Device Sync, those errors will also impact Interface sync processing. When running dependent syncs at large scale, such as Interface Sync and Device Sync, ScienceLogic recommends that you run them serially, not at the same time. Running both syncs at the same time will greatly hinder ServiceNow processing and scalability.

**WARNING:** Skylar Automation only syncs network interfaces that are aligned with devices that are already synced with ServiceNow. Before setting up network interface sync, you must first *sync devices between Skylar One and ServiceNow*.

To sync Skylar One network interfaces with ServiceNow:

1. In the Skylar Automation user interface, go to the **Applications** page and run the "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application to read all existing Skylar One Device Classes, Organizations, ServiceNow CIs, and Companies and write them to a cache.

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

1. When that application completes, go to the **Applications** page and select the "Sync Interfaces from Skylar One to ServiceNow" application.

- Click **[Configure]**. The **Configuration** pane appears:

- Complete the following fields, as needed:

- Configuration.** Select the relevant configuration object 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, using the **Configuration** field. If you need to update this value, you will need to define the **region** variable in the configuration object 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.

- ***disconnect\_field***. Acts as the boolean field on the Configuration Items table to determine "True/False" for the Skylar One monitoring status.
- ***Include\_Orgs***. If you want to include a specific set of Skylar One Organizations in the interface 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.

**CAUTION:** A misconfiguration in the ***Include Orgs*** field might change the monitoring flag for one or more devices. In other words, a misconfiguration in this field could switch the Skylar One Monitored flag from "true" to "false", removing that interface or interfaces from the device sync.

- ***snow\_request\_limit***. Specify the number of objects fetched from ServiceNow in each request. The default is 500 objects per chunk.
- ***snow\_batch\_size***. Specify the total number of objects being sent to ServiceNow in each triggered application. The default is 150 objects per batch.
- ***discovery\_source***. Specify the ServiceNow Discovery source. The default is "Other Automated".
- ***snow\_chunk\_size***. Specify the number of objects to send in each chunk or in each sub-payload with a batch (specified in the ***snow\_batch\_size*** field). The default is 150 objects per chunk or sub-payload.
- ***max\_count\_for\_relationships***. Specify the maximum number of device IDs to query for sets of relationships in a single chunk from Skylar One.
- ***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 only sync enabled interfaces. When enabled, this filter is applied to both the Skylar One and ServiceNow side pulls. By default, this option is not selected.
- ***override\_snow\_mappings***. Toggle on to send all interfaces to the "cmdb\_ci\_network\_adapter" table in ServiceNow, ignoring the user-entered IANA type mappings and skipping the hardcoded mappings. However, the interface will be ignored if it is missing the required fields to be correctly created and identified in ServiceNow on the cmdb\_ci\_network\_adapter table. The required fields are "ip", "macAddress", and "hardwareDescription".

**NOTE:** If the *override\_snow\_mappings\_toggle* is toggled off, the following IANA types have hard-coded mappings that cannot be overwritten. This is done to match ServiceNow discovery:

- "ethernetCsmacd"
  - If the parent type is "Servers", maps to the **cmdb\_ci\_network\_adapter** table in ServiceNow
  - If the parent type is "Network.Firewall", maps to the **cmdb\_ci\_network\_adapter** table in ServiceNow
  - If the parent type is "Network.Switches", maps to the **cmdb\_ci\_network\_adapter** and **dscy\_switchport** tables in ServiceNow
  - If the parent type is "Network.Router", maps to the **cmdb\_ci\_network\_adapter** and **dscy\_router\_interface** tables in ServiceNow
  - If the parent type is "Network.Balancers", maps to the **cmdb\_ci\_network\_adapter** and **cmdb\_ci\_lb\_vlan** tables in ServiceNow
- "slip"
  - If the parent type is "Network.Switches", maps to the **dscy\_switchport** table in ServiceNow
- "propVirtual"
  - If the parent type is "Network.Switches", maps to the **cmdb\_ci\_network\_adapter** and **dscy\_swch\_partition** tables in ServiceNow
  - If the parent type is "Network.Router", maps to the **cmdb\_ci\_network\_adapter** and **dscy\_swch\_partition** tables in ServiceNow
- "tunnel"
  - If the parent type is "Network.Router", maps to the **cmdb\_ci\_network\_adapter** table in ServiceNow

- **Domain\_Separation.** Select this option if your ServiceNow environment is *domain-separated*, where the data, processes, and administrative tasks have been organized into logical groupings called *domains*. If your ServiceNow instance is domain-separated, the user listed in the *snow\_user* field must be a member of the top domain and have access to *all* of the domains you intend to integrate. Also, ServiceNow should be the "source of truth" for organizations if your environment is domain-separated. By default, this option is not selected.
- **change\_interface\_organizations.** Toggle on (blue) this option to allow interfaces to change companies with an Interface Sync if the companies are within the same domain in ServiceNow. If you change the Organization/Company and another field, but do not enable this parameter, Skylar Automation will not apply any changes to the interface. By default, this option is not selected.

- **drop\_company.** Toggle on (blue) this option to remove the `sys_id` in existing companies from the sync. This will prevent the "Company" field from sending to ServiceNow. This option has no effect if the **domain\_separation** parameter is enabled for this application. By default, this option is not selected. If you select this option, the ServiceNow company **sys\_id** for the interface and the Skylar One organization **crm\_id** for the Skylar One device will not be compared when determining if an update has occurred. Do not select this option if you selected the **Domain\_Separation** option, as ServiceNow requires PowerFlow to send the "company" field in domain-separated environments. Also, do not select this option if you selected the **change\_interface\_organizations** option, as changing the company for an interface in ServiceNow requires the "company" field to be sent.
- **verify\_snow\_ssl.** Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
- **generate\_report.** Select this option to create a "Sync Interfaces from Skylar One to ServiceNow" report. This report contains counts of the number of interfaces gathered from Skylar One and ServiceNow, the number of unchanged interfaces, as well as the number of creations, updates, and disconnects sent to ServiceNow. Detailed interface information for the interfaces sent to ServiceNow is also displayed. The reports are available on the **Reports** page of the Skylar Automation user interface.
- **Simulation\_Mode.** Select this option if you want to perform a simulated run of this application to show you the potential results of that run. This option is not selected by default.
- **enable\_advanced\_topology.** Select this option to allow the pull and process advanced topology step to be executed when the sync is run. If disabled, this step will be skipped. This toggle defaults to a value of *False*.
- **disconnect\_type.** Specify the disconnect method for disconnecting Skylar One entities from ServiceNow. Enter either *flip\_boolean\_field* or *remove\_sources\_entry* to determine the behavior for disconnects.
- **retry\_jitter.** When selected, instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. By default, this option is not selected.
- **retry\_backoff.** When selected, instead of using a defined interval between retries, Skylar Automation will incrementally increase the interval between retries. By default, this option is not selected.
- **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.
- **interface\_table\_relations.** Using JSON, define the relationship for any ServiceNow tables used in the **mappings** parameter, below. If a table is included in the mappings, but the relationship is not defined in the **interface\_table\_relations**, the Interface Sync application will fail.
- **interface\_advanced\_mappings.** Using JSON, define the mappings for the combination of interface IANA type and parent device type to a specific ServiceNow table. This configuration option is blank by default, and must be configured to define the combination of specific mappings for splitting relationships.

**NOTE:** An example value for the *interface\_advanced\_mappings* field is as follows: {  
 "ethernetCsmacd": { "Servers": [ "dscy\_router\_interface" ],  
 "System.Skylar One": [ "cmdb\_ci\_network\_adapter", "dscy\_router\_interface" ] }, "12vlan": { "Network.Router": [ "dscy\_router\_interface" ] } }. In this example, an additional split is added to the ethernetCsmacd / Servers mapping. This will cause the two server splits (even if one is thrown out) to be tagged as "duplicates", which will affect the relationships they defined. Additionally, a new parent device type mapping of System.Skylar One to ethernetCsmacd is added. Also, a new IANA/parent device type mapping is added for 12vlan.

**TIP:** The hardcoded mappings will take priority and cannot be overwritten. If accidentally added to the user-inputted mappings, it will be ignored. The tables defined in the advanced mappings must be one of the following:

```
[
  "cmdb_ci_network_adapter",
  "dscy_switchport",
  "dscy_router_interface",
  "dscy_swch_partition",
  "cmdb_ci_lb_vlan",
]
```

The tables defined in the mappings have required fields (field names are Skylar One interface fields) which cannot be ignored. Any interface missing these fields will be thrown out:

```
{
  "cmdb_ci_network_adapter": ["ip", "macAddress",
  "hardwareDescription"],
  "dscy_switchport": ["index"],
  "dscy_router_interface": ["macAddress", "hardwareDescription"],
  "dscy_swch_partition": ["index"],
  "cmdb_ci_lb_vlan": ["hardwareDescription"],
}
```

- **interface\_relation\_mappings.** Using JSON, define which specific interface relationships to send to ServiceNow. This configuration option is blank by default, and must be configured to define the relationships you want to send directly.

**NOTE:** An example value for the **interface\_relation\_mappings** field is as follows: `{ "cmdb_ci_network_adapter": { "relations": [ { "parent": "cmdb_ci_network_adapter", "rel_type": "Connects to::Connected by", "create_to_duplicate": False, "create_from_duplicate": False } ] } }`. In this example, the sync creates relationships between interfaces where the child interface is on the "cmdb\_ci\_network\_adapter" table and the parent interface is on the "cmdb\_ci\_network\_adapter" table, and neither is a "duplicate". The "create\_to\_duplicate" parameter being false means that relationships where the child interface is a duplicate will not be created. The "create\_from\_duplicate" parameter being false means that relationships where the parent interface is a duplicate will not be created. Determining if an interface is a "duplicate" depends on the number of destination tables in the hard-coded or user provided mappings. For example, in the hard-coded mappings, if the IANA type of the interface is "ethernetcsmacd" and the parent device type is "network.switches", the interface is sent to two tables: "dscy\_switchport" and "cmdb\_ci\_network\_adapter". Even if one of the interfaces is discarded due to missing required fields for creation, both interfaces are tagged as "duplicates".


5. In the **Mappings** section, click the **[Edit]** button (Skylar Automation 2.7.0 or later) or the expand button (▼) to view and edit the mappings between specific Skylar One interface types to different tables in ServiceNow. These mappings can be used to send an IANA type to a specific ServiceNow table if its parent is not included in the hardcoded list above. The hard-coded mappings will supersede any user-input mappings.

For more information about how to view, edit, and create mappings, see [Editing Mappings in a Skylar Automation Application](#). This topic also covers how to use the **Company Mapping Override** and the **Attribute Mappings** sections, below.

6. In the **Company Mapping Override** section, you can create a new mapping for the ServiceNow **company** field. You can override the **company** field with a different ServiceNow field where you can read and write the **company\_sys\_id** for CIs.
7. In the **Attribute Mappings** section, click the Edit button to view and edit the mappings for any other custom device attributes you want to sync between Skylar One (the first column) and ServiceNow (the second column).

**NOTE:** When an attribute value is "0" in Skylar One, the corresponding field in ServiceNow might display as empty.

8. Click **[Save]**. The **Configuration** pane closes.
9. Click **[Run]** to run the application.
10. When the application completes, go to ServiceNow and type "cmdb\_ci\_network\_adapter.list". The **Network Adapters** page appears with a list of synced interfaces.

11. You can add additional columns to this page using the Update Personalized List button (  ) to add columns like *Operational status*, which is synced from Skylar One. The *Operational status* value is different from the [Skylar One Monitored] value, but Skylar Automation tracks both values.
12. Select a network interface from the list tab to see more information about the interface.

---

## Scheduling Skylar Automation Applications

Using the Skylar Automation user interface, you can configure Skylar Automation applications to run on a schedule instead of manually running the applications. As a best practice, if you use any of these applications, ScienceLogic recommends that you schedule those applications, in the following order:

- "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes"

**IMPORTANT:** The "Cache ServiceNow Companies, CIs and Skylar One Orgs, Device Classes" application was removed from version 4.0.0 of the "ServiceNow CMDB" SyncPack and moved to the "ServiceNow Base" SyncPack version 4.0.0. In order to run the application, you must have version 4.0.0 of the "ServiceNow Base" SyncPack installed.

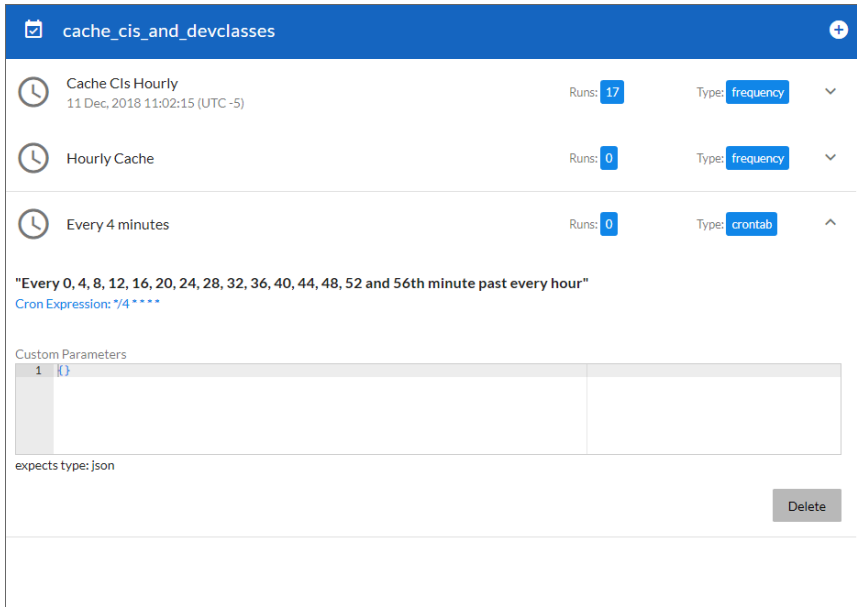
- "Sync Devices from Skylar One to ServiceNow"
- "Sync Interfaces from Skylar One to ServiceNow"

**TIP:** ScienceLogic recommends that you schedule these applications to run at least every 23 hours. You can also schedule additional applications as needed.

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 **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"*.

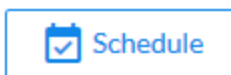
2. Select a schedule from the list to view the details for that schedule.
3. Click the + icon to create a schedule. A blank **Schedule** window appears:

The screenshot shows a window titled "cache\_cis\_and\_devclasses" with a blue header. Below the header, there is a "Schedule Name" field with a placeholder "Schedule Name". To the right of this field is a toggle switch labeled "Switch to Cron Expression". Below that is a "Frequency" field with a "secs" unit. Underneath is a "Custom Parameters" field containing a JSON editor with the following content:

```
1 {
2 {}
3 }
```

Below the JSON editor, it says "expects type: json". At the bottom right of the window is a blue button labeled "Save Schedule".

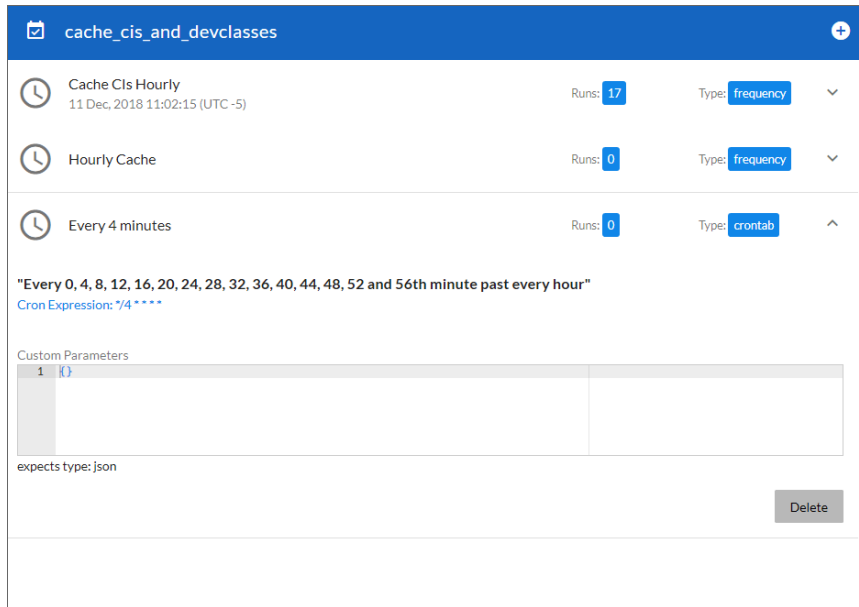
4. In the **Schedule** window, complete the following fields:
  - **Schedule Name.** Type a name for the schedule.
  - **Switch to.** Use this toggle to switch between a cron expression and setting the frequency in seconds.
    - **Cron expression.** Select this option to schedule the application using 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. The values in this field will override any application variables that exist in the PowerFlow application you are scheduling.
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]**.

**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.

---

## Log Messages for the "Generate Required CI Relations for ServiceNow" Application

This section describes the different types of log messages you might see in the Step Log when you run the "Generate Required CI Relations for ServiceNow" application. Please note that this application is a report used by PowerFlow, and it does not send any data to ServiceNow.

The following message displays if there are devices in a device tree that do not currently have a CI class mapping assigned.

```
Warning: 2751 Relations with missing mappings detected. Please re-run app
with log level 10 to troubleshoot.
```

In this situation, the device tree cannot be built in ServiceNow. To address this issue, make sure that you have your entire technology tree mapped out in the *mappings* section of the "Sync Devices from Skylar One to ServiceNow" application or in the *mappings* section of the "Generate Required CI Relations for ServiceNow" application.

If you do a Custom Run of the "Generate Required CI Relations for ServiceNow" application in Debug mode (log level 10), the application will create a log that displays the parent and child class, CI, and device ID. For example:

```
Debug: Missing Mapping for Device. Parent: {"class": "VMware | Cluster",
      "ci": None, "id": 76}, Child: {"class": "VMware | Host Server", ci:
      "cmdb_ci_esx_server", id: 363 }
```

The following message appears if the GraphQL payloads had bad data for parent and or child devices:

```
Warning: 10 bad payloads received from Skylar One. Re-run app in debug to
troubleshoot.
```

If you do a Custom Run the application in Debug mode, the application will create a log that displays these payloads.

The following message appears if all relations are mapped:

```
Flow: No missing relations found!
```

The following message appears if there is a parent/child relation between ServiceNow CI classes that does not currently exist in ServiceNow and is required to sync those devices:

```
Flow: Missing Relations: [{"parent": "cmdb_ci_vcenter_folder", "child":
      "cmdb_ci_esx_server"}, {"parent": "cmdb_ci_vcenter", "child": "cmdb_ci_
      vcenter_datacenter"}]
```

Refer to the labels in the log (above) to determine which CI class is the parent type and which is the child type. To address this issue, navigate to your ServiceNow instance and create the required service rules based on the recommendations in the **Step Log**.

The following message appears if the application encounters a list of relations that are required, but were successfully found in ServiceNow:

```
Info: Found Relations: [{"parent": "cmdb_ci_vcenter_folder", "child":  
  "cmdb_ci_esx_server"}, {"parent": "cmdb_ci_vcenter", "child": "cmdb_ci_  
  vcenter_datacenter"}]
```

This message lets you verify that your mappings and relations are configured correctly.

---

# Chapter

# 5

## Troubleshooting the CMDB SyncPack

---

### Overview

This chapter includes troubleshooting resources and procedures to use with the "ServiceNow CMDB" SyncPack.

This chapter covers the following topics:

<i>Initial Troubleshooting Steps</i> .....	96
<i>Resources for Troubleshooting</i> .....	96
<i>Troubleshooting CMDB Sync</i> .....	100

---

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

### Skylar Automation

1. Run `docker service ls` on the Skylar Automation server:
  - Note the Docker container version.
  - Verify that the Docker services are running.
2. If a certain service is failing, make a note of the service name and version.
3. 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`.
4. 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 if you are running a ServiceNow certified application or a Service Graph SyncPack.
3. Make a note of the SyncPack application that is failing in 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.

---

## 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 administrating 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@fsunisl4lab ~]# docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
zmmihuj5v301	iservices_gui	replicated	1/1	repository.auto.sciencelogic.local:5000/is-gui:1.7.0	*:80->80/tcp,*:443->443/tcp
l0w92l1vmb3	iservices_redis	replicated	1/1	redis:4.0.2	
lms8h1jtumif	iservices_flower	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	*:5555->5555/tcp
hh3pt2l0isrf	iservices_scheduler	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	
ht1mltvq6kxh	iservices_contentapi	replicated	1/1	repository.auto.sciencelogic.local:5000/is-api:1.7.0	*:5000->5000/tcp
cyin8qgsudmi	iservices_rabbitmq	replicated	1/1	rabbitmq:3	
xl1l8h3j8f86	iservices_visual	replicated	2/1	dockersamples/visualizer:latest	*:8081->8080/tcp
vy938w8uauw	iservices_couchbase	replicated	1/1	repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0	*:8091->8091/tcp,*:8092->8092/tcp
0->8093/tcp,*:8094->8094/tcp,*:11210->11210/tcp					
z1bxatxz7uf	iservices_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/iseservices` 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.
4. Specify the configuration object for the custom run in the **Configuration** field, and add any JSON parameters in the **Custom Parameters** field, if needed.
5. Click **[Run]**.

To run an application in Debug Mode using the API:

1. POST the following to the API endpoint:

```
https://<Skylar Automation_IP>/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 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:

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... Traceback (most recent call last): File "/usr/local/lib/python3.8/site-packages/ipaascore/BaseStep.py", 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('loascommon.loas.exceptions.MissingRequiredStepParameter: target vcue is required but is not populated in either delete_devices or in Sync Service...

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}}'
```

## Troubleshooting CMDb Sync

This section contains specific troubleshooting steps for the CMDb SyncPack.

### Issues Creating CIs in ServiceNow

If you can successfully send data to your ServiceNow system, but you encounter issues with creating CIs in the ServiceNow CMDb, this section provides troubleshooting steps to help you test the payload and identify possible issues. These steps might be helpful if you have set up datasource precedence rules.

1. In ServiceNow, search for "import" in the filter navigator.
2. Select **ScienceLogic > Device > Imports**. The **Device Import** window appears.
3. From the list, select the Device Import log entry you want to view.
4. Copy the data from the **Payload** field in the log entry and decode the data from its Base64 encoding.
5. In the decoded string of data, remove the square brackets from the first and last line: ("[" , "]"")
6. Copy this modified JSON payload, and then use the filter navigator to search for "Identification Simulation" or select **Configuration > Identification Simulation**.
7. On the **Identification Simulation** page, click the **[Start]** button in the **Start with Existing Payload** section. The **Insert JSON Payload** page appears.
8. In the **Source** field, select *ScienceLogic* as the data source.
9. In the **Please insert payload below** field, paste the JSON payload you edited in step 5.
10. Click the **[Execute]** button and review the payload to identify any potential issues.

## Check the Device Remediation Table

1. In ServiceNow, go to the Device Remediation page (ScienceLogic: CMDB Automation > Remediation > Device Remediation)
2. Check for remediation records.
3. The remediation record provides a comprehensive view of the payload that was submitted and the error returned by the IRE.

## Enabling Debugging of the CI Payload

You must have administrator-level permissions in ServiceNow to access the system properties and enable debugging of the Configuration Item (CI) payload in the [ServiceNow Identification and Reconciliation module](#).

To enable debugging of the CI payload in ServiceNow:

1. On the ServiceNow system, check to see if the `glide.cmdb.logger.source.identification_engine` record exists in `sys_properties.list`.
  - If the record exists, set this value to `(* or debugVerbose)`
  - If the record does not exist, you will need to create the record.
2. To create the record, complete the following fields:
  - **Name.** `glide.cmdb.logger.source.identification_engine`
  - **Description.** Enable and configure the type of details the system logs when using the Identification and Reconciliation module outside the scope of identification simulation, such as when using an API, an ECC queue, or scheduled jobs (info, warn, error, debug, or debugVerbose).
  - **Type.** String.

- **Value:** \* or debugVerbose

**NOTE:** Set the system property of **Value** back to `error` when troubleshooting is complete.

3. Run the "Sync Devices from Skylar One to ServiceNow" application. The system logs will have "identification\_engine" as the source, and the log messages will contain `identification_engine`: Input.

After the identification run is complete, the ServiceNow logs contain additional data about the run.

Created	Level	Message
2026-05-05 15:22:19	Warning	identification_engine - logId:[9fedeaaf47e8] CIProcessor: Payload dependent record 12 has sysId: 9ed48a5a2b4e2210dc44f8600991bf85

## Mappings between Skylar One, ServiceNow, and Other Applications

---

### Overview

This appendix contains information about how Skylar One and ServiceNow discover and model out different technologies, using VMware as an example. The VMware example explains how the differences between the systems requires rules be configured to bridge the gap between Skylar One and ServiceNow.

This chapter covers the following topics:

<i>Overview</i> .....	105
<i>VMware vCenter</i> .....	105
<i>Default Customer Relationship Overrides</i> .....	106
<i>VMware vCenter Folder</i> .....	108
<i>VMware Datastore</i> .....	108
<i>VMware Virtual Machine Instance</i> .....	109
<i>VMware Cluster</i> .....	109
<i>ESX Server</i> .....	110
<i>ESX Resource Pool</i> .....	110
<i>vCenter Network</i> .....	110
<i>Distributed Virtual Switch</i> .....	111

*Distributed Virtual Portgroup* ..... 111

## Overview

Different platforms do not model topology the same way. The default Customer Specific CI relation overrides delivered with the "ServiceNow CMDB" SyncPack provides several topology mappings by default. Existing customers upgrading to new versions of these mappings will not override any mappings that are already set.

The following topics explain how Skylar One and ServiceNow model topology differently and the mappings that are provided in this SyncPack.

**IMPORTANT:** The Discovery and Service Mapping Patterns application (sn\_itom\_pattern) plugin is required to align with your VMware mapping requirements.

## VMware vCenter

The following table lists the Class mappings that are delivered by default. These are the same Class mappings that are used in Service Graph solution.

**IMPORTANT:** The *VMware | vCenter* device class is mapped to the *Cloud Service Account* class and not the *vCenter* class directly. The default mapping adheres to ServiceNow Cloud Management Relationships.

Virtual Type	Device Category	Device Class   Sub Class	Class Name	Table Name
ScienceLogic		ServiceNow		
Component	Virtual.Infrastructure	VMware   Datacenter	VMware vCenter Datacenter	cmdb_ci_vcenter_datacenter
Component	Virtual.Infrastructure	VMware   Folder	VMware vCenter Folder	cmdb_ci_vcenter_folder
PowerFlow	Servers.VMware	VMware   vCenter	Cloud Service Account	cmdb_ci_cloud_service_account
Component	Virtual.Infrastructure	VMware   Datastore	VMware vCenter Datastore	cmdb_ci_vcenter_datastore
Component	Virtual.Guest	VMware   Virtual Machine	VMware Virtual Machine Instance	cmdb_ci_vmware_instance
Component	Servers.VMware	VMware   Cluster	VMware vCenter Cluster	cmdb_ci_cluster
Component	Virtual.Network	VMware   Network	VMware vCenter Network	cmdb_ci_network

Virtual Type	Device Category	Device Class   Sub Class	Class Name	Table Name
Component	Virtual.Network	VMware   Distributed Virtual Portgroup	VMware Distributed Virtual Port Group	cmdb_ci_vcenter_dv_port_group
Component	Virtual.Network	VMware   Distributed Virtual Switch	VMware Distributed Virtual Switch	cmdb_ci_dvs
Component	Virtual.Host	VMware   Host Server	ESX Server	cmdb_ci_esx_server
Component	Virtual.Infrastructure	VMware   Resource Pool	ESX Resource Pool	cmdb_ci_esx_resource_pool

## Default Customer Relationship Overrides

The following table shows the relationship breakdown for the default Customer Relationship Overrides:

Relationships Mapped	Relationship	Child Class	Identifier
VMware vCenter Datacenter	Hosted on::Hosts	Cloud Service Account	Yes
VMware vCenter Datacenter	Contains::Contained by	VMware Distributed Virtual Port Group	
VMware vCenter Datacenter	Contains::Contained by	VMware vCenter Network	
VMware vCenter Datacenter	Contains::Contained by	VMware Distributed Virtual Switch	
VMware vCenter Datacenter	Contains::Contained by	VMware vCenter Folder	
VMware vCenter Datacenter	Contains::Contained by	VMware vCenter Datastore	
VMware vCenter Folder	Hosted on::Hosts	VMware vCenter Datacenter	
VMware vCenter Datastore	Hosted on::Hosts	VMware vCenter Datacenter	
VMware vCenter Datastore	Provides storage for::Stored on	VMware Virtual Machine Instance	
VMware vCenter Datastore	Used by::Uses	ESX Server	
VMware Virtual Machine Instance	Hosted on::Hosts	VMware vCenter Datacenter	
VMware Virtual Machine Instance	Connected by::Connects	VMware Distributed Virtual Port Group	
VMware vCenter	Contains::Contained by	VMware vCenter Cluster	

Relationships Mapped	Relationship	Child Class	Identifier
Datacenter			
VMware vCenter Cluster	Hosted on::Hosts	VMware vCenter Cluster	
VMware vCenter Cluster	Hosted on::Hosts	VMware vCenter Datacenter	
VMware vCenter Cluster	Members::Member of	ESX Resource Pool	
ESX Server	Hosted on::Hosts	VMware vCenter Datacenter	
ESX Resource Pool	Hosted on::Hosts	VMware vCenter Datacenter	
ESX Resource Pool	Defines resources for::Gets resources from	ESX Server	
ESX Resource Pool	Members::Member of	VMware Machine Instance	
ESX Resource Pool	Defines resources for::Gets resources form	VMware vCenter Cluster	
VMware vCenter Network	Hosted on::Hosts	VMware vCenter Datacenter	
VMware vCenter Network	Provided by::Provides	ESX Server	
VMware Distributed Virtual Port Group	Hosted on::Hosts	VMware vCenter Datacenter	
VMware Distributed Virtual Switch	Hosted on::Hosts	VMware vCenter Datacenter	
VMware Distributed Virtual Switch	Connected by::Connects	VMware Distributed Virtual Port Group	
VMware Distributed Virtual Switch	Provided by::Provides	ESX Server	

## Specific Class Breakdown

ScienceLogic Fields	ServiceNow Fields	Identifier	Notes
Instance UUID	Account ID	Yes	Verify that Instance UUID is a custom attribute on the vCenter device class, and is being populated with a GUID. Download the latest version of the VMware PowerPack.
Instance UUID	Object ID	Yes	
Name	name		
Datacenter type	Datacenter Type		
Instance URL	Instance URL		

ScienceLogic Fields	ServiceNow Fields	Identifier	Notes
Component Unique ID	Object ID	Yes	
Manufacture	Manufacturer		
Model	Model ID		
Instance UUID	vCenter Instance UUID		Service Graph Only
company	company		
	Location		
Component Unique ID	Managed object reference ID (MORID)		
Name	Name		

---

## VMware vCenter Folder

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Component Unique ID	Managed object reference ID (MORID)	
Model	Model ID	
	Location	
Instance UUID	vCenter Instance UUID	Service Graph Only
Name	Name	
Company	Company	
Manufacturer	Manufacture	

---

## VMware Datastore

ScienceLogic Fields	ServiceNow Fields	Notes
Instance UUID (Service Graph only), Component Unique ID	Morid, object_id, url	
Name	Name	
Component Unique ID	Object ID	
Instance UUID	vCenter Instance UUID	Service Graph Only

ScienceLogic Fields	ServiceNow Fields	Notes
Host ID	URL	
Model	Model ID	
Company	Company	
Component Unique ID	Managed object reference ID (MORID)	
Manufacturer	Manufacturer	
	Location	

---

## VMware Virtual Machine Instance

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Instance UUID	vCenter Instance UUID	Service Graph Only
Manufacture	Manufacturer	
Model	Model ID	
	Location	Service Graph Only
Company	Company	
Host ID	Bios UUID	

---

## VMware Cluster

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Name	Name	
Component Unique ID	Managed object reference ID (MORID)	
Manufacturer	Manufacturer	
	Location	
Model	Model ID	
Instance UUID	vCenter Instance UUID	Service Graph Only
Company	Company	

---

## ESX Server

ScienceLogic Fields	ServiceNow Fields	Notes
Host ID	Correlation ID	
Component Unique ID	Managed object reference ID (MORID)	
Model	Model ID	
	Location	
Manufacturer	Manufacture	
Serial	Serial Number	
Company	Company	
Name	Name	
Component Unique ID	Object ID	

---

## ESX Resource Pool

ScienceLogic Fields	ServiceNow Fields	Notes
Name	Name	
Component Unique ID	Object ID	
	Location	
Instance UUID	vCenter Instance UUID	Service Graph Only
Manufacturer	Manufacturer	
Company	Company	
Component Unique ID	Managed object reference ID (MORID)	
Model	Model ID	

---

## vCenter Network

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Instance UUID	vCenter Instance UUID	Service Graph Only

ScienceLogic Fields	ServiceNow Fields	Notes
Manufacturer	Manufacturer	
	Location	Service Graph Only
Name	Name	
Company	Company	
Component Unique ID	Managed object reference ID (MORID)	

---

## Distributed Virtual Switch

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Company	Company	
Instance UUID	vCenter Instance UUID	Service Graph Only
Name	Name	
Component Unique ID	Managed object reference ID (MORID)	
Manufacturer	Manufacturer	
Model	Model	

---

## Distributed Virtual Portgroup

ScienceLogic Fields	ServiceNow Fields	Notes
Component Unique ID	Object ID	
Instance UUID	vCenter Instance UUID	Service Graph Only
Model	Model ID	
Name	Name	
	Location	Service Graph Only
Manufacturer	Manufacturer	

© 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.

ScienceLogic

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

International: +1-703-354-1010