Integration Service: ServiceNow CMDB Synchronization PowerPack

Version 3.1.0

Table of Contents

Chapter

1

Introduction to the ServiceNow CMDB Synchronization PowerPack

Overview

This chapter describes the *ServiceNow CMDB* Synchronization PowerPack, which lets you integrate SL1 with the ServiceNow Configuration Management Database (CMDB).

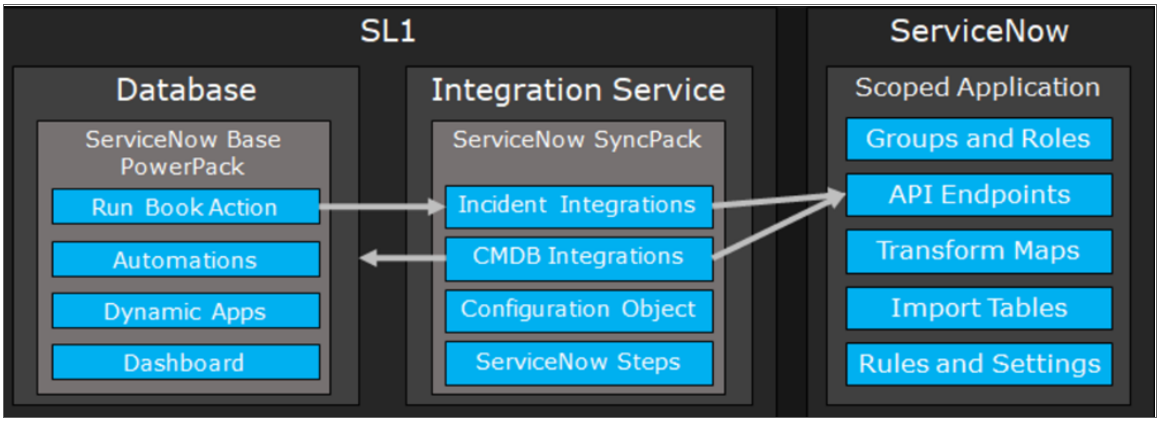
This Synchronization PowerPack maintains and enhances the ServiceNow CMDB by sharing discovered device information, importing and exporting data bi-directionally between SL1 and ServiceNow, and by automatically maintaining ServiceNow Configuration Item (CI) relationships.

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

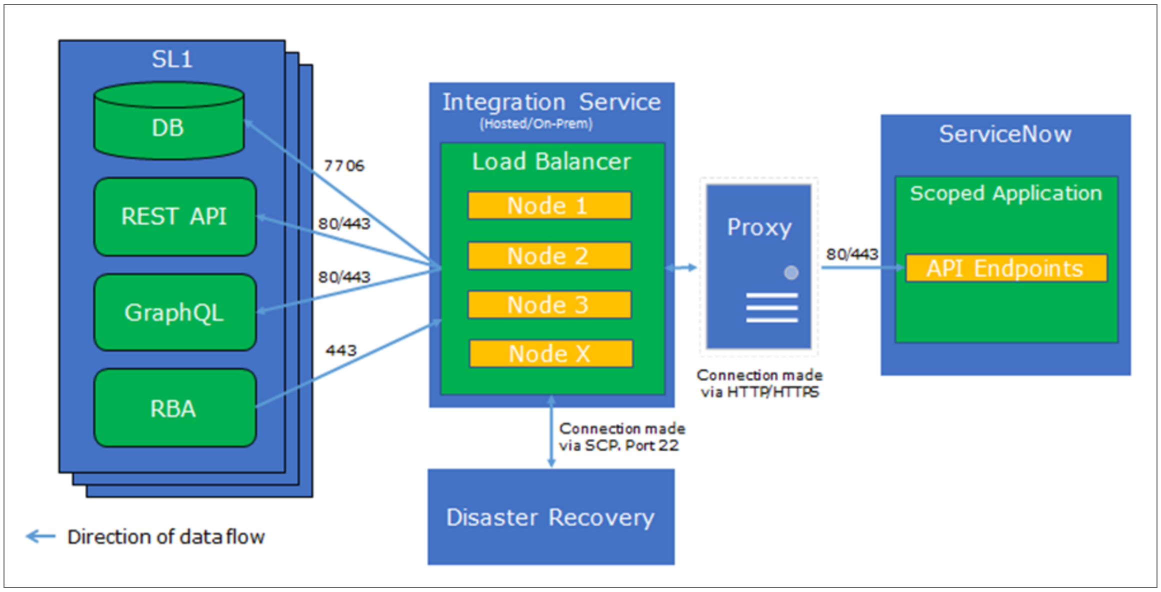
This chapter covers the following topics:

Architecture Overview for ServiceNow Synchronization PowerPacks

The following diagram details the various elements that are contained in SL1 and the Integration Service, and how the Integration Service sits between the core SL1 platform and an external data platform:



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



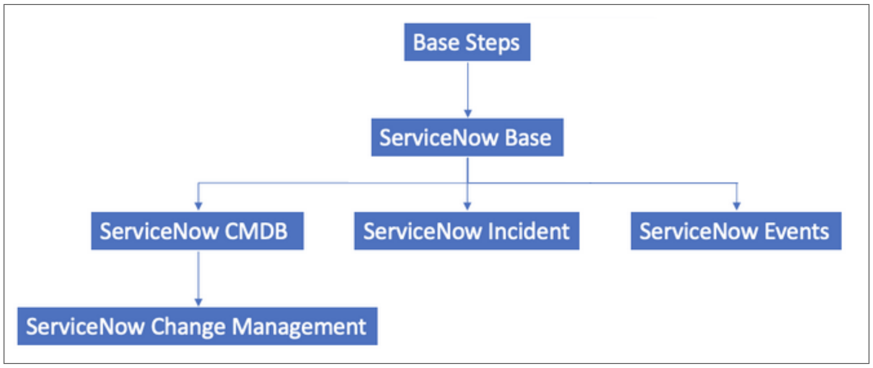
SL1 and ServiceNow Terminology

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

SL1	ServiceNow
Device	CI (Configuration Item)
Discovery Session	Service Catalog Request
Event	Incident
Organization	Company/Domain
Schedule	Maintenance Schedule
Topology	Dependency

Dependency Map for ServiceNow Synchronization PowerPacks

The following graphic describes which Synchronization PowerPacks depend on other Synchronization PowerPacks:



TIP: For more information about the Standard *Base Steps* Synchronization PowerPack, see the ***Integration Service Platform*** manual.

Prerequisites for ServiceNow Synchronization PowerPacks

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

To install any of the ScienceLogic ServiceNow Synchronization PowerPacks, you must have administrator access to both SL1 and ServiceNow. Specifically, you will need:

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

Additional information related to uploading, installing, and upgrading the ServiceNow Synchronization PowerPacks:

- If you want to upload and install multiple ServiceNow Synchronization PowerPacks at the same time, you should upload *all* of the Synchronization PowerPacks first, and then install them to address any dependencies between the Synchronization PowerPacks.
- If you are upgrading from version 1.8.4 of the Integration Service, you can first move to version 2.5.0 of the *ServiceNow* Synchronization PowerPack, then upgrade to version 2.0 of the Integration Service, and finally move to version 3.0.0 or later of the relevant ServiceNow Synchronization PowerPacks.
- If you made customizations to version 2.5.0 or earlier of the *ServiceNow* Synchronization PowerPack, you must make a copy of that Synchronization PowerPack and save it as a new Synchronization PowerPack to keep your customizations. For more information, see the ***Integration Service for Developers*** manual.
- If you are starting out with version 2.0.0 of the Integration Service platform, you should skip version 2.5.0 of the *ServiceNow* Synchronization PowerPack and instead install version 3.0.0 of the relevant ServiceNow Synchronization PowerPacks.
- There is no backwards compatibility after you upgrade a ServiceNow Synchronization PowerPack from version 2.5.0 to 3.0.0.

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

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

Source IP	Integration Service Destination	Integration Service Source Port	Destination Port	Requirement
Integration Service	SL1 API	Any	TCP 443	SL1 API Access
Integration Service	ServiceNow API	Any	TCP 443	ServiceNow API Access
Integration Service	SL1 Database	Any	TCP 7706	SL1 Database Access

Integration Applications Included in the ServiceNow CMDB Synchronization PowerPack

This section lists the integration applications that are in the *ServiceNow CMDB Synchronization PowerPack*.

Integration Applications

The following integration applications are included with the ServiceNow Synchronization PowerPack:

- ***Cache ServiceNow CIs and SL1 Device Classes***. Reads all existing SL1 Device Classes and ServiceNow CIs and writes them to a cache. To perform a Device Sync, run this integration application before you run the "Sync Devices from SL1 to ServiceNow" integration application. For more information, see ***Syncing Devices from SL1 to ServiceNow***.
- ***Delete Devices from SL1***. Lets you delete devices in a specific SL1 Virtual Collector Group (VCUG) if those devices have not been modified in SL1 for a specified amount of time that is set in the application. For more information, see ***Deleting Devices***.
- ***Generate Required CI Relations for ServiceNow***. Pulls device class mappings from the "Sync Devices from SL1 to ServiceNow" and the "Sync CI Attributes from ServiceNow to SL1" integration 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 Integration Service user interface. For more information about the log messages, see ***Log Messages for the "Generate Required CI Relations for ServiceNow" Application Integration***.
- ***Sync Advanced Topology from SL1 to ServiceNow***. Reads Dynamic Component Mapping relationships from SL1 and syncs those relationships with ServiceNow. You must run both the "Sync Devices from SL1 to ServiceNow" application and the "Sync Interfaces from SL1 to ServiceNow" application at least twice on new Integration Service systems to populate the cache for this integration application. For more information, see ***Syncing Advanced Topologies from SL1 to ServiceNow***.
- ***Sync Business Services from SL1 to ServiceNow***. Reads Business Services, IT Services, and Device Services in SL1 and syncs them with business services in ServiceNow. This integration application creates and updates services, but it does not delete services. For more information, see ***Syncing Business Services from SL1 to ServiceNow***.
- ***Sync CI Attributes from ServiceNow to SL1***. Reads CI attributes from ServiceNow and maps those attributes to asset and attribute fields in SL1. This application uses the mappings and additional attributes options from the "Sync Devices from SL1 to ServiceNow" application. This integration application can also sync the location and production state attributes from ServiceNow to SL1. For more information, see ***Syncing CI Attributes from ServiceNow to SL1***.
- ***Sync Device Groups from SL1 to ServiceNow***. Collects all device groups and group IDs from SL1 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. For more information about Device Sync, see ***Syncing Devices from SL1 to ServiceNow***.
- ***Sync Devices from SL1 to ServiceNow***. Syncs devices and their properties and relationships from SL1 to ServiceNow. For more information, see ***Syncing Devices from SL1 to ServiceNow***.

- ***Sync Discovery Requirements***. Processes credentials from SL1, processes collector groups, device templates, virtual device classes, and collectors, and then syncs organizations and device groups. For more information, see *Discovery Sync*.
- ***Sync Discovery Session Status from SL1 to ServiceNow***. Collects and processes Discovery sessions from SL1, and collects Discovery session logs. For more information, see *Discovery Sync*.
- ***Sync Discovery Templates from SL1 to ServiceNow***. Syncs SL1 discovery sessions that contain a configured string to ServiceNow and creates Service Catalog templates in ServiceNow. You can use those templates for discovering or monitoring CIs. For more information, see *Discovery Sync*.
- ***Sync File Systems from SL1 to ServiceNow***. Reads file systems discovered in SL1 and then maps them to a parent CI record in ServiceNow. For more information, see *Syncing File Systems from SL1 to ServiceNow*.
- ***Sync Installed Software from SL1 to ServiceNow***. Reads all available software packages from ServiceNow and the devices aligned to that software by region and syncs them with SL1. For more information, see *Syncing Installed Software between SL1 and ServiceNow*.
- ***Sync Interfaces from SL1 to ServiceNow***. Collects network interface data from ServiceNow and SL1, and then runs multiple CI syncs for each interface to be synced. For more information, see *Syncing Network Interfaces from SL1 to ServiceNow*.
- ***Sync Organizations from SL1 to ServiceNow***. Pulls organizations from SL1 and syncs to ServiceNow. For more information, see *Syncing Organizations from SL1 to ServiceNow*.
- ***Sync Service Requests from ServiceNow to SL1***. Processes Discovery sessions and posts Discovery sessions and new virtual devices to SL1. Also enables device decommissioning for devices you no longer want to monitor. This application was formerly named "Sync Discovery Session Requests from ServiceNow to SL1". For more information, see *Configuring a ServiceNow Service Request for Discovery Sync*.
- ***Sync Software Packages from SL1 to ServiceNow***. Reads all software packages from and creates new CIs in ServiceNow. Run this integration before running the "Sync Installed Software" integration application. For more information, see *Syncing Installed Software between SL1 and ServiceNow*.

Integration Applications (Internal)

To view the internal integration applications, click the Filter icon (≡) on the **Integrations** page and select *Show Hidden Integrations*. Internal integration applications are hidden by default. The following integration 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 SL1.
- ***Create Discovery Session in SL1***. Creates and starts a Discovery session in SL1 and updates the ServiceNow service request.
- ***Create ServiceNow CI***. Creates a new ServiceNow CI with a mappings dictionary, but does not attempt to look up new CIs.
- ***Create Virtual Device in SL1***. Creates a virtual device in SL1 and updates the Requested Item (RITM) value.
- ***Post Attribute DB Calls to SL1***. Posts attribute database calls to SL1.
- ***Post Attribute Rest Calls to SL1***. Posts attribute REST calls to SL1.

- ***Post Company and Organization Updates***. Posts company and organization updates to ServiceNow or SL1.
- ***Post Discovery-dependent Data to ServiceNow***. Posts data used by a Discovery session to ServiceNow.
- ***Post Installed Software to ServiceNow***. Posts installed software data to ServiceNow.
- ***Post New Companies to ServiceNow***. Posts new companies to ServiceNow.
- ***Post New Organization to SL1***. Posts a new organization to SL1.
- ***Process Remove Device Requests from ServiceNow to SL1***. Pulls requested device information form SL1 and validates the requests to remove a device from monitoring. Removed devices are placed in an SL1 Virtual Collector Group.
- ***Pull and Post Discovery Logs***. Pulls Discovery session logs from SL1 and posts updates to ServiceNow.

Chapter

2

Installing and Configuring the CMDB Synchronization PowerPack

Overview

The following workflow covers how to install and configure this Synchronization PowerPack:

1. In the Integration Service, download, import, and install the *ServiceNow CMDB* Synchronization PowerPack.
2. In ServiceNow, enable cross-scoped access and install the "ScienceLogic SL1: CMDB & Incident Automation" application (also called the "Scoped Application").
3. In ServiceNow, install and activate the "ServiceNow Configuration Management for Scoped Apps (CMDB)" Plugin.
4. In ServiceNow, enable the "ServiceNow Identification and Reconciliation Module".
5. For domain-separated ServiceNow instances only, install the "ScienceLogic Domain Separation (Global)" update set in ServiceNow.
6. In ServiceNow, create an update set with containment rules and hosting rules for Device Sync.

These installation and configuration steps are covered in the following topics:

Downloading, Importing, and Installing the ServiceNow CMDB Synchronization PowerPack

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

WARNING: If you are *upgrading* to this version of the Synchronization PowerPack from a previous version, make a note of any settings you made on the **Configuration** pane of the various integration applications in this Synchronization PowerPack, as these settings are *not* retained when you upgrade. However, any mappings you added to the **attribute_mappings** section for the "Sync Devices from SL1 to ServiceNow" integration application *are* retained when you upgrade.

Downloading the Synchronization PowerPack

To locate and download the Synchronization PowerPack:

1. Go to the ScienceLogic Support site at https://support.sciencelogic.com/s/.
2. Click the **Product Downloads** tab, select *PowerPacks*, and then click the "Synchronization" link. The **Synchronization PowerPack Downloads** page appears.
3. Click the name of the Synchronization PowerPack you want to install. The **PowerPack** page appears.
4. In the **Files** list, locate the Synchronization PowerPack **.whl** file, click the down arrow button, and select *Download*.

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

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

Importing the Synchronization PowerPack

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

To import a Synchronization PowerPack in the Integration Service user interface:

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

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

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

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

Installing the Synchronization PowerPack

To install a Synchronization PowerPack in the Integration Service user interface:

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

> **TIP:** By default, the **SyncPacks** page displays only activated and installed PowerPacks. If you do not see the PowerPack that you want to install, click the toggle icon (≡) on the **SyncPacks** page and select *Show All SyncPacks* to see a list of the uninstalled PowerPacks.

2. Click **[Yes]** to confirm the activation and installation. When the Synchronization PowerPack is activated, the **SyncPacks** page displays a green check mark icon (✓) for that Synchronization PowerPack. If the activation or installation failed, then a red exclamation mark icon (❗) appears.

TIP: While the Synchronization PowerPack is installing, you cannot click any of the options that appear when you click the **[Actions]** button (⋮).

3. For more information about the activation and installation process, click the check mark icon (✓) or the exclamation mark icon (!) in the **Activated** column for that Synchronization PowerPack. For a successful installation, the "Activate & Install SyncPack" integration application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

Allowing Cross-Scoped Access in ServiceNow

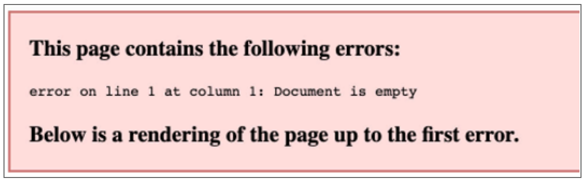
When using custom tables that are extended from the cmdb_ci table, you must configure cross-scope access for any custom tables created outside of the base ServiceNow deployment.

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
```

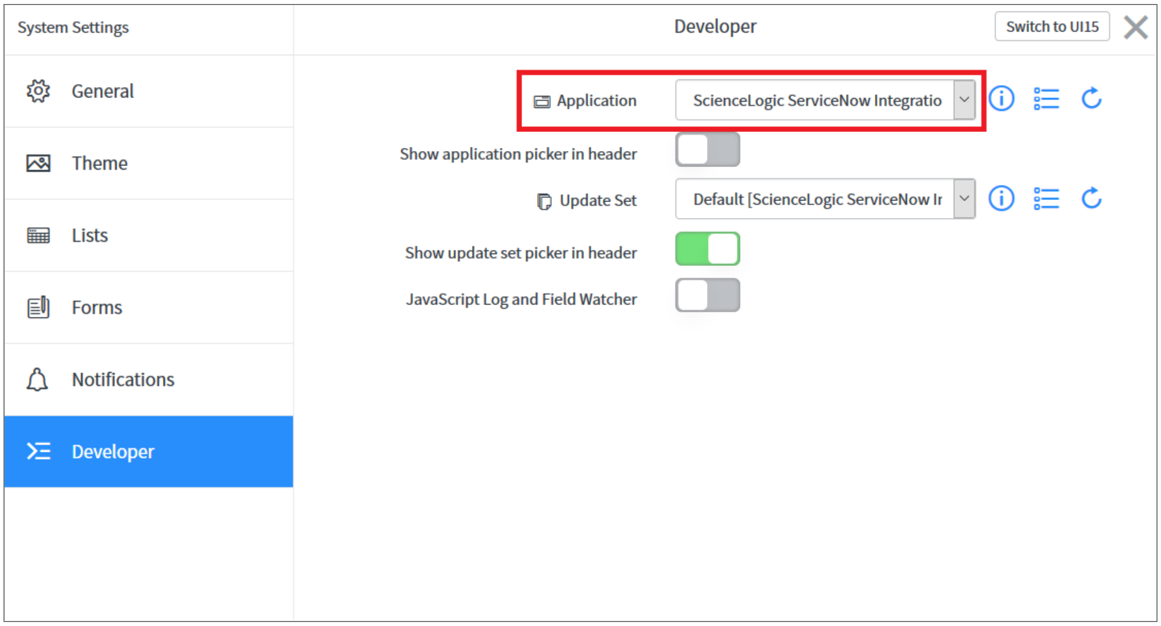
Below is a rendering of the page up to the first error.

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. Click the **Settings** icon ( ) and select the **Developer** tab. The **Developer System Settings** window appears:



Developer

Application: ScienceLogic ServiceNow Integratio

Show application picker in header: ○

Update Set: Default [ScienceLogic ServiceNow Ir

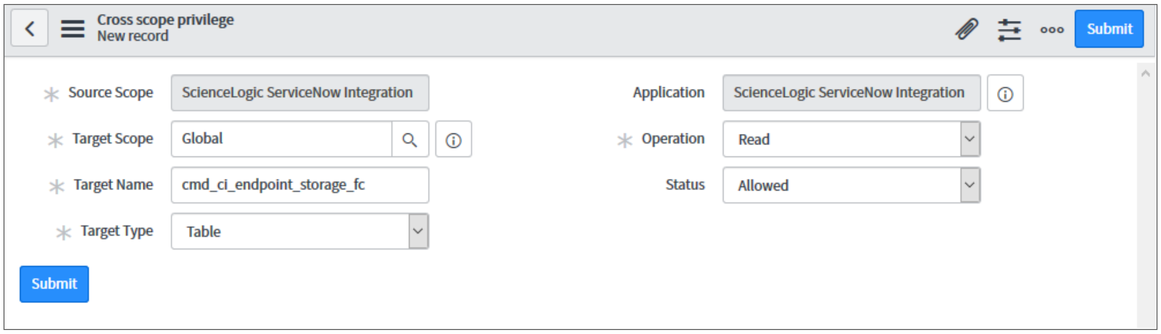
Show update set picker in header: ●

JavaScript Log and Field Watcher: ○

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).

NOTE: 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:



Cross scope privilege
New record

* Source Scope	ScienceLogic ServiceNow Integration	Application	ScienceLogic ServiceNow Integration
* Target Scope	Global	* Operation	Read
* Target Name	cmd_ci_endpoint_storage_fc	Status	Allowed
* Target Type	Table		

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*.
   - ***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 https://docs.servicenow.com/bundle/madrid-application-development/page/build/applications/reference/c_CrossScopePrivilegeRecord.html.

Installing the ScienceLogic SL1: CMDB & Incident Automation Application in ServiceNow

You must install the "ScienceLogic SL1: CMDB & Incident Automation" application on your ServiceNow instance to enable this Synchronization PowerPack. The "ScienceLogic SL1: CMDB & Incident Automation" application is also known as the "Certified Application" or the "Scoped Application".

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

WARNING: Integration Service instances running version 2.0.0 or later of the ServiceNow integration applications are not backwards-compatible with the previous ServiceNow update sets or with SyncServer. After you install the "ScienceLogic SL1: CMDB & Incident Automation" application on your ServiceNow instance, you need to upgrade your ServiceNow integration applications to version 2.0.0 or later on all Integration Service instances. The "ScienceLogic SL1: CMDB & Incident Automation" application is also not backwards-compatible with SyncServer. This change cannot be reverted.

You must first request the "ScienceLogic SL1: CMDB & Incident Automation" application from the ServiceNow Store, and then you can install it.

To request and install the Certified Application:

1. Go to the ServiceNow Store at https://store.servicenow.com and search for "ScienceLogic SL1".
2. Select the "ScienceLogic SL1: CMDB & Incident Automation" application. The detail page for the application appears.
3. Click the **[Get]** button and log in with your HI credentials.

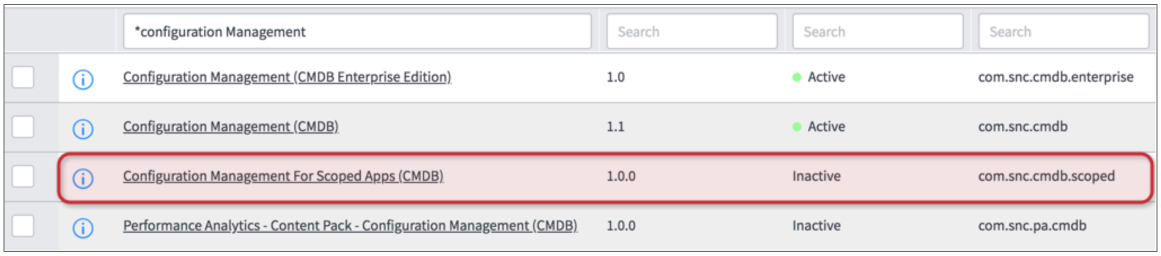
4. After the request is approved, log in to ServiceNow as an administrator and navigate to **Application Manager** (System Applications > Applications or My Company Applications).
5. Click **[Downloads]** in the menu header or search for "ScienceLogic".
6. Click the version drop-down for the "ScienceLogic ServiceNow Integration" application listing to make sure you are using the correct version of the application that is compatible with your version of this Synchronization PowerPack.
7. Click the **[Install]** button for the "ScienceLogic ServiceNow Integration" 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.

Installing and Activating the CMDB Plugin in ServiceNow

Installing the "ServiceNow Configuration Management for Scoped Apps (CMDB) Plugin" is required to manage your Configuration Items. This involves activating the "Configuration Management For Scoped Apps (CMDB) Plugin" on your ServiceNow instance.

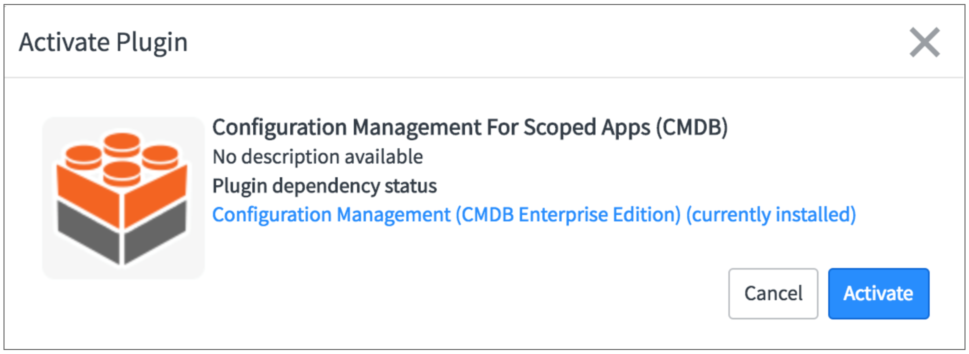
To activate the "Configuration Management for Scoped Apps (CMDB) Plugin":

1. In ServiceNow, log in as an administrator and navigate to **Plugins** (System Definition > Plugins).
2. Search for **Configuration Management For Scoped Apps (CMDB)** and select it.



		*configuration Management	Search	Search	Search
□	ⓘ	Configuration Management (CMDB Enterprise Edition)	1.0	● Active	com.snc.cmdb.enterprise
□	ⓘ	Configuration Management (CMDB)	1.1	● Active	com.snc.cmdb
□	ⓘ	Configuration Management For Scoped Apps (CMDB)	1.0.0	Inactive	com.snc.cmdb.scoped
□	ⓘ	Performance Analytics - Content Pack - Configuration Management (CMDB)	1.0.0	Inactive	com.snc.pa.cmdb

3. Click *Activate/Upgrade* in the ***Related Links*** section.
4. In the Activate Plugin notification, click **[Activate]**.



Activate Plugin

Configuration Management For Scoped Apps (CMDB)

No description available

Plugin dependency status

Configuration Management (CMDB Enterprise Edition) (currently installed)

Enabling the ServiceNow Identification and Reconciliation Module

This Synchronization PowerPack uses the " ServiceNow Identification and Reconciliation" module to create and de-duplicate CI records. The Integration Service builds a JSON-formatted string that is sent to the "ServiceNow Identification and Reconciliation" module. The following link provides additional detail about the formatting of the JSON-formatted string: IdentificationEngineScriptableApi.

The JSON-formatted string is sent directly to a custom-scripted API endpoint and run through the IdentificationEngineScriptable API. Identification (Insert or Update) of Configuration Items (CIs) is handled by the ServiceNow Identification and Reconciliation module.

For more information about the "ServiceNow Identification and Reconciliation" module, see CMDB Identify and Reconcile. See also Reconciliation Rules, CMDB Identification Rules, and Identification engine error messages.

Configuring Service Rules for Device Sync

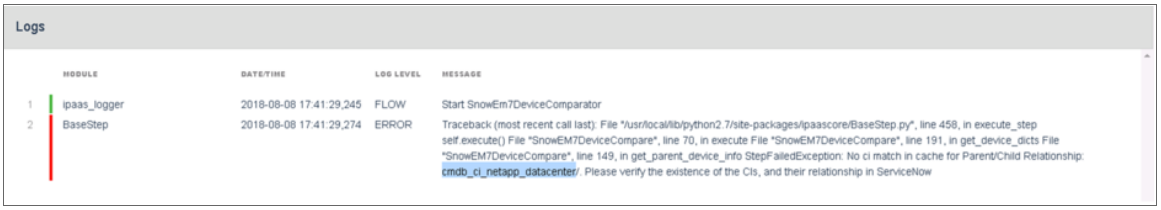
The *ServiceNow CMDB* Synchronization PowerPack utilizes class hierarchies to build relationships in ServiceNow. This requires building ***service rules*** (containment rules and hosting rules) in ServiceNow to correctly identify dependent CIs during the business discovery process and service mapping. ***Containment rules*** describe which CIs are contained by a given CI. ***Hosting rules*** describe the environment on which a CI runs.

ScienceLogic recommends packaging all of the service rules into a ServiceNow update set so that you can be easily package and deploy these changes across environments. For more information, see ***Creating a ServiceNow Update Set***.

These rules or "mappings" are defined in the "Sync Devices from SL1 to ServiceNow" integration application in the Integration Service user interface. These mappings connect an **SL1 device class** to a **ServiceNow CI class**, which determines the CI class that ServiceNow uses when creating the CI in ServiceNow.

For more information about the "ServiceNow Identification and Reconciliation" module, see CMDB dependent relationship rules and CMDB Identification Rules at the ServiceNow website.

For example, if you experience error messages about missing relationships in ServiceNow when you run the "Sync Devices from SL1 to ServiceNow" integration application in the Integration Service user interface, you might be missing certain containment rules or mappings that are needed to complete the export process:



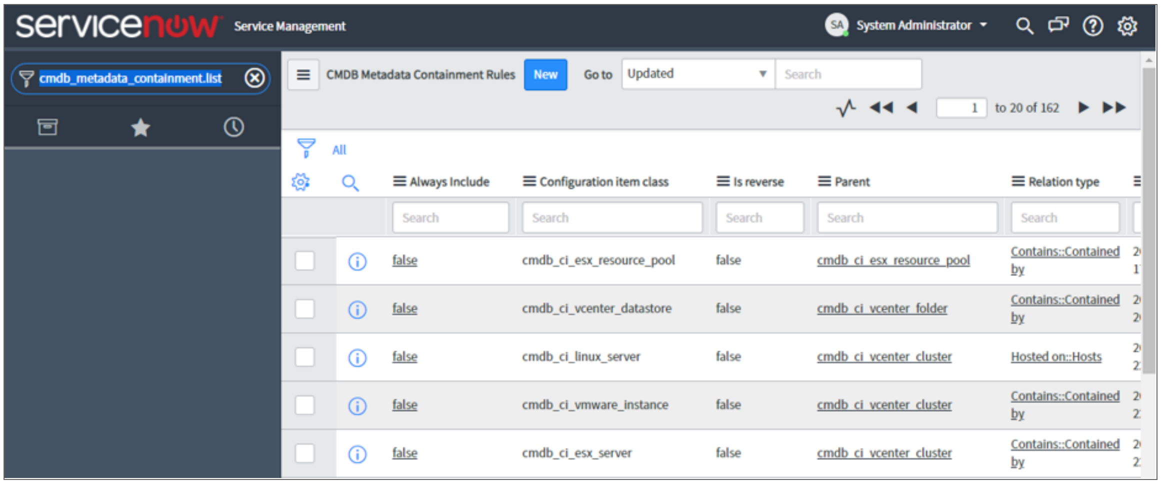
```
   MODULE          DATE/TIME                LOG LEVEL  MESSAGE
1  ipaas_logger    2018-08-08 17:41:29,245  FLOW       Start SnowEm7DeviceComparator
2  BaseStep        2018-08-08 17:41:29,274  ERROR      Traceback (most recent call last): File "/usr/local/lib/python2.7/site-packages/ipaascore/BaseStep.py", line 458, in execute_step
                                                       self.execute() File "SnowEM7DeviceCompare", line 70, in execute File "SnowEM7DeviceCompare", line 191, in get_device_dicts File
                                                       "SnowEM7DeviceCompare", line 149, in get_parent_device_info StepFailedException: No ci match in cache for Parent/Child Relationship:
                                                       cmdb_ci_netapp_datacenter/. Please verify the existence of the CIs, and their relationship in ServiceNow
```

Containment Rules

Containment rules are chained to each other in a group, with a CI type that is the top-level (root) parent of the group.

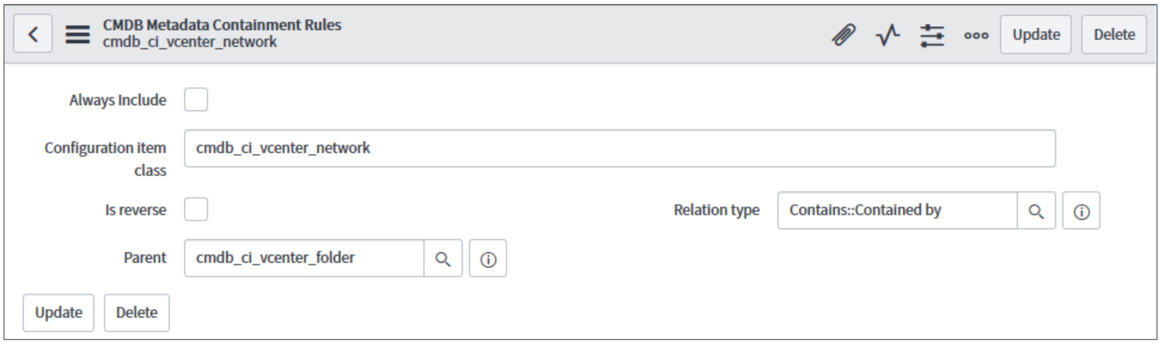
To create containment rules:

1. In ServiceNow, type "cmdb_metadata_containment.list" n the filter navigator to access the **CMDB Metadata Containment Rules** page:



Always Include	Configuration item class	Is reverse	Parent	Relation type
false	cmdb_ci_esx_resource_pool	false	cmdb_ci_esx_resource_pool	Contains::Contained by
false	cmdb_ci_vcenter_datastore	false	cmdb_ci_vcenter_folder	Contains::Contained by
false	cmdb_ci_linux_server	false	cmdb_ci_vcenter_cluster	Hosted on::Hosts
false	cmdb_ci_vmware_instance	false	cmdb_ci_vcenter_cluster	Contains::Contained by
false	cmdb_ci_esx_server	false	cmdb_ci_vcenter_cluster	Contains::Contained by

2. Click **[New]**. A new **CMDB Metadata Containment Rules** record appears:



CMDB Metadata Containment Rules
cmdb_ci_vcenter_network

Always Include □

Configuration item class: cmdb_ci_vcenter_network

Is reverse □

Relation type: Contains::Contained by

Parent: cmdb_ci_vcenter_folder

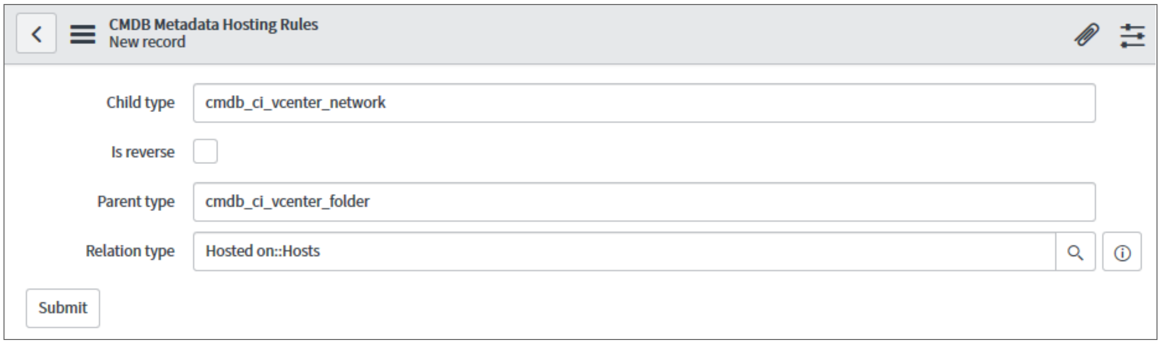
3. Complete the following fields:
   - ***Configuration item class***. Specify the child CI class.
   - ***Parent***. Specify the parent CI class.
   - ***Relation type***. Specify the relationship type. The common relationship types used by the ServiceNow integration are "contained" or "contained by", depending on your CMDB. Click the magnifying glass icon to select the correct value.

4. Click **[Submit]**.
5. In the Integration Service user interface, go to the **Integrations** page and manually run the "Cache ServiceNow CIs and SL1 Device Classes" integration application.
6. Run the "Sync Devices from SL1 to ServiceNow" integration application and make sure that no errors exist due to missing CI relationships.

Hosting Rules

Hosting rules can only be one level, and they always involve resources such as physical or virtual hardware.

1. In the ServiceNow filter navigator, type "cmdb_metadata_hosting.list" to view the **CMDB Metadata Hosting Rules** page.
2. Click **[New]**. A new CMDB Metadata Hosting Rules record appears:



CMDB Metadata Hosting Rules
New record

Child type: cmdb_ci_vcenter_network

Is reverse: □

Parent type: cmdb_ci_vcenter_folder

Relation type: Hosted on::Hosts

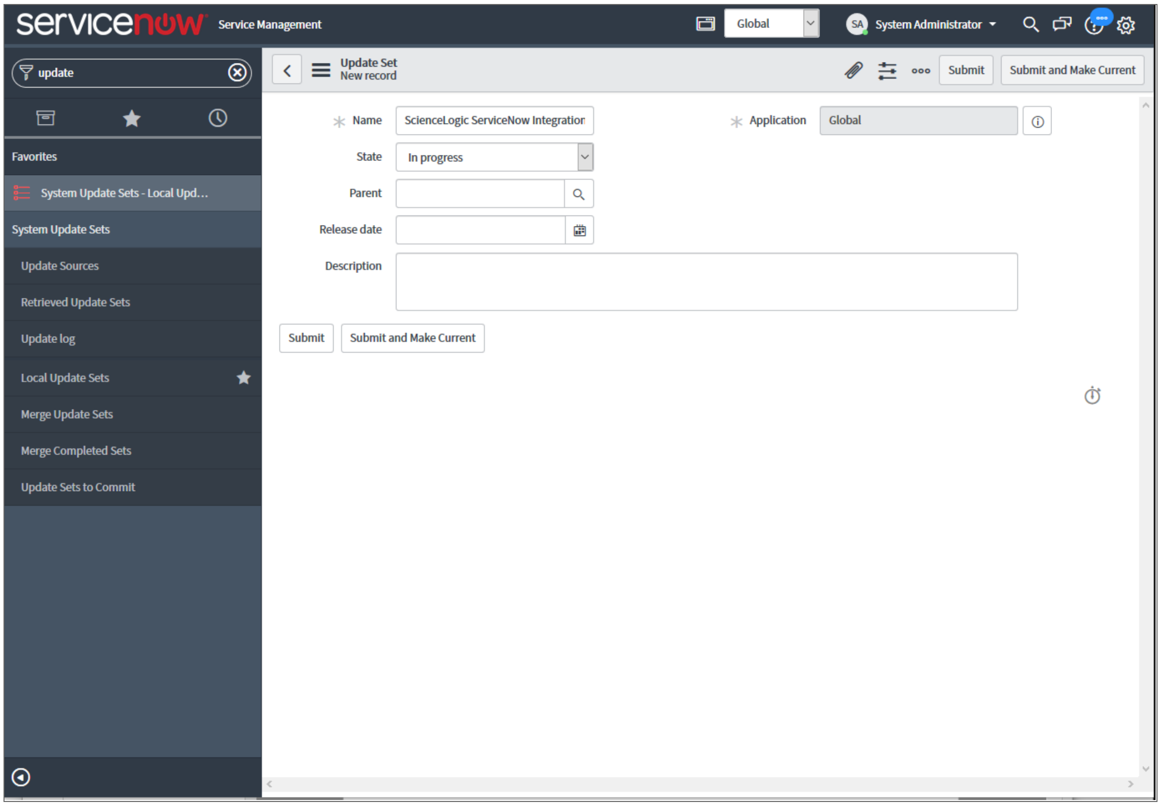
3. In the New Metadata Hosting Rules record, complete the following fields:
   - ***Child type***. Specify the child CI class.
   - ***Parent type***. Specify the parent CI class.
   - ***Relation type***. Specify the relationship type. The common relationship types used by the ServiceNow integration are "Hosts" or "Hosted on", depending on your CMDB. Click the magnifying glass icon to select the correct value.
4. Click **[Submit]**.
5. Add any additional containment and hosting rules that are needed to build the CI relationships in ServiceNow.
6. In the Integration Service user interface, go to the **Integrations** page and manually run the "Cache ServiceNow CIs and SL1 Device Classes" integration application.
7. Run the "Sync Devices from SL1 to ServiceNow" integration application and make sure that no errors exist due to missing CI relationships.

Creating a ServiceNow Update Set

ScienceLogic recommends packaging the service rules into a standalone ServiceNow update set that you can export if needed. An ***update set*** is an XML file containing a group of customizations that can be moved from one ServiceNow instance to another. This update set should include any changes or configurations to the service rules for the ServiceNow Identification and Reconciliation Module.

To create a standalone update set in ServiceNow:

1. In ServiceNow, enable the Developer Update set picker by clicking the **Settings** icon (⚙) and selecting the **Developer** tab.
2. Select the **Show update set picker in header** toggle to enable it, and then close the **System Settings** page.
3. In the filter navigator, search for local update sets.
4. Under **System Update Sets**, select **Local Update Sets** and click **[New]**. A new Update Set record appears:



Name: ScienceLogic ServiceNow Integration

Application: Global

State: In progress

Parent:

Release date:

Description:

5. Complete the following fields:

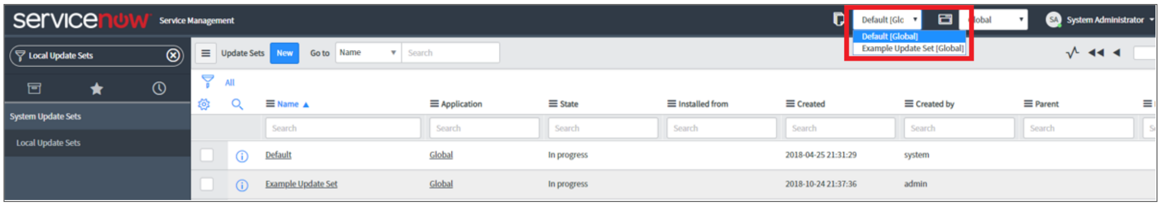
   - ***Name***. Specify a name that describes the rules of this update set.
   - ***Application***. Set the application scope to *Global*.
   - ***State***. Set to *In Progress*.
   - Complete the remaining fields as needed.

6. Click **[Submit]** or **[Submit and Make Current]**. If you selected **[Submit and Make Current]**, go to step 8.
7. If you clicked **[Submit]**, you can select the update set in the picker in the header or navigate to the update set and select *Make This My Current Set* in the **Related links** section. You are now ready to make changes in your ServiceNow Instances.
8. When you are done with all updates in the update set, change the update set ***State*** field to *Complete*.

Adding Service Rules to an Update Set

If you submitted your new update set and made it "Current" in ***Creating a ServiceNow Update Set***, skip this step and go to ***Exporting an Update Set***.

If you did not make your update set current, you will need to identify your current update set and move all of the service rules you need into your update set. You can find this information in a dropdown located in the ServiceNow navigation bar:



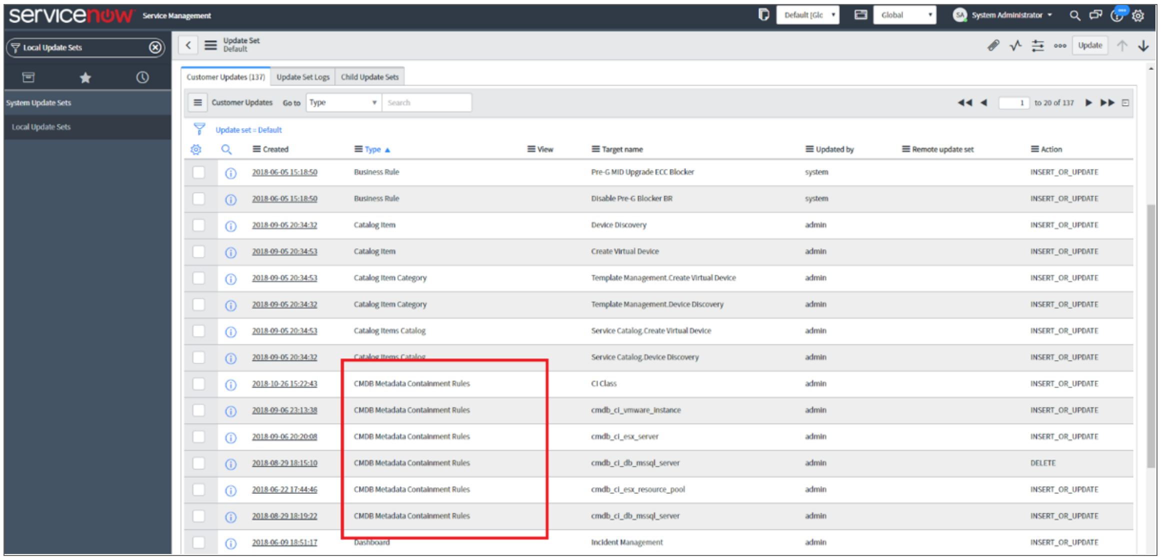
Name ▲	Application	State	Installed from	Created	Created by	Parent
Default	Global	In progress		2018-04-25 21:31:29	system	
Example Update Set	Global	In progress		2018-10-24 21:37:36	admin	

All of the service rules that you defined are tracked in the update set record under the **[Customer Updates]** tab.

To add all created service rules to your update set:

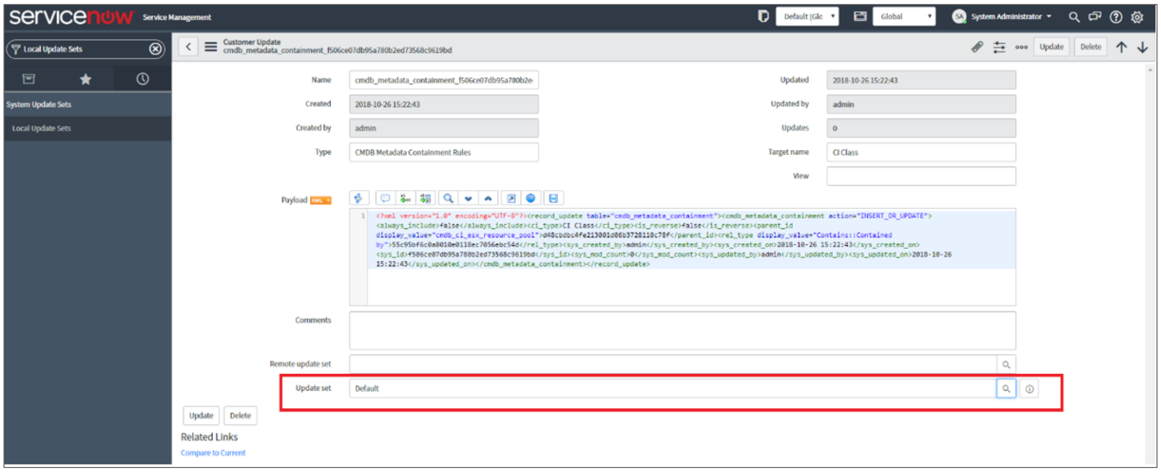
1. In the ServiceNow filter navigator, type "local update sets" to view a list of update sets on the ServiceNow instance.
2. Identify your current update set, which should have all of the created service rules tracked.
3. Identify the self-created update set that you want to contain all the service rules. This is the update set that you want to export.
4. Select the current update set that has all of the already-created service rules.

5. On the **[Customer Updates]** tab, identify all of the records with a ***Type*** of either *CMDB Metadata Containment Rules* or *CMDB Metadata Hosting Rules*:



	Created	Type ▲	View	Target name	Updated by	Remote update set	Action
☐	2018-06-05 15:18:50	Business Rule		Pre-G MID Upgrade ECC Blocker	system		INSERT_OR_UPDATE
☐	2018-06-05 15:18:50	Business Rule		Disable Pre-G Blocker BR	system		INSERT_OR_UPDATE
☐	2018-09-05 20:34:32	Catalog Item		Device Discovery	admin		INSERT_OR_UPDATE
☐	2018-09-05 20:34:53	Catalog Item		Create Virtual Device	admin		INSERT_OR_UPDATE
☐	2018-09-05 20:34:53	Catalog Item Category		Template Management.Create Virtual Device	admin		INSERT_OR_UPDATE
☐	2018-09-05 20:34:32	Catalog Item Category		Template Management.Device Discovery	admin		INSERT_OR_UPDATE
☐	2018-09-05 20:34:53	Catalog Items Catalog		Service Catalog.Create Virtual Device	admin		INSERT_OR_UPDATE
☐	2018-09-05 20:34:32	Catalog Items Catalog		Service Catalog.Device Discovery	admin		INSERT_OR_UPDATE
☐	2018-10-26 15:22:43	CMDB Metadata Containment Rules		CI Class	admin		INSERT_OR_UPDATE
☐	2018-09-06 23:13:38	CMDB Metadata Containment Rules		cmdb_ci_vmware_instance	admin		INSERT_OR_UPDATE
☐	2018-09-06 20:20:08	CMDB Metadata Containment Rules		cmdb_ci_esx_server	admin		INSERT_OR_UPDATE
☐	2018-08-29 18:15:10	CMDB Metadata Containment Rules		cmdb_ci_db_mssql_server	admin		DELETE
☐	2018-06-22 17:44:46	CMDB Metadata Containment Rules		cmdb_ci_esx_resource_pool	admin		INSERT_OR_UPDATE
☐	2018-08-29 18:19:22	CMDB Metadata Containment Rules		cmdb_ci_db_mssql_server	admin		INSERT_OR_UPDATE
☐	2018-06-09 18:51:17	Dashboard		Incident Management	admin		INSERT_OR_UPDATE

6. Select each of the relevant service rule records and set the ***Update set*** field to match the update set you want to export. Click the magnifying glass icon to select the correct value:



Customer Update
cmdb_metadata_containment_f506ce07db95a780b2ed73568c9619bd

Name: cmdb_metadata_containment_f506ce07db95a780b2e

Created: 2018-10-26 15:22:43

Created by: admin

Type: CMDB Metadata Containment Rules

Updated: 2018-10-26 15:22:43

Updated by: admin

Updates: 0

Target name: CI Class

View:

Payload

```
<?xml version="1.0" encoding="UTF-8"?><record_update table="cmdb_metadata_containment"><cmdb_metadata_containment action="INSERT_OR_UPDATE">
<always_include>false</always_include><ci_type>CI Class</ci_type><is_reverse>false</is_reverse><parent_id
display_value="cmdb_ci_esx_resource_pool">d48cbdbc4fe213001d06b3728110c78f</parent_id><rel_type display_value="Contains::Contained
by">55c95bf6c0a8010e0118ec7056ebc54d</rel_type><sys_created_by>admin</sys_created_by><sys_created_on>2018-10-26 15:22:43</sys_created_on>
<sys_id>f506ce07db95a780b2ed73568c9619bd</sys_id><sys_mod_count>0</sys_mod_count><sys_updated_by>admin</sys_updated_by><sys_updated_on>2018-10-26
15:22:43</sys_updated_on></cmdb_metadata_containment></record_update>
```

Comments:

Remote update set:

Update set: Default

7. Click **[Update]**.

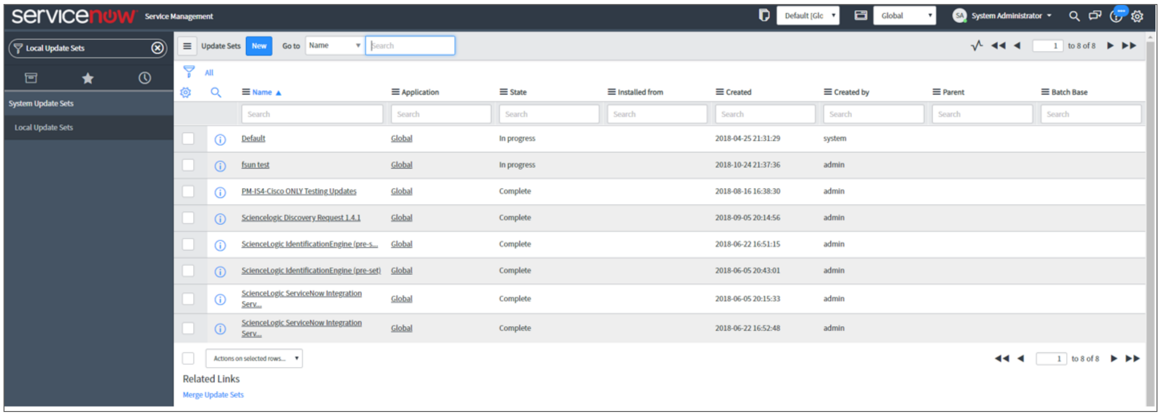
8. Repeat steps 6-7 until all relevant containment and hosting rules are in the new update set, and then go to ***Exporting an Update Set***.

Exporting an Update Set

After you have created your update set and defined the service rules, mark your update set as *Complete* and export it to an XML file.

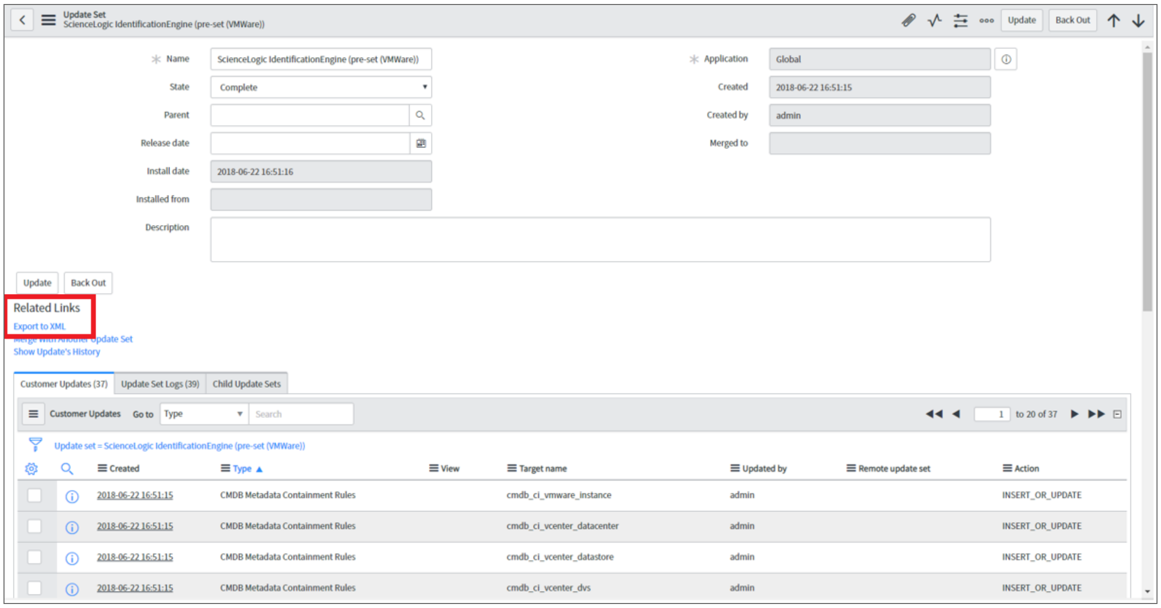
To export an update set:

1. In the ServiceNow filter navigator, type "Local Update Sets" to view a list of update sets in ServiceNow:



Name ▲	Application	State	Installed from	Created	Created by	Parent	Batch Base
Default	Global	In progress		2018-04-25 21:31:29	system		
fsun test	Global	In progress		2018-10-24 21:37:36	admin		
PM-IS4-Cisco ONLY Testing Updates	Global	Complete		2018-08-16 16:38:30	admin		
Sciencelogic Discovery Request 1.4.1	Global	Complete		2018-09-05 20:14:56	admin		
ScienceLogic IdentificationEngine (pre-s...	Global	Complete		2018-06-22 16:51:15	admin		
ScienceLogic IdentificationEngine (pre-set)	Global	Complete		2018-06-05 20:43:01	admin		
ScienceLogic ServiceNow Integration Serv...	Global	Complete		2018-06-05 20:15:33	admin		
ScienceLogic ServiceNow Integration Serv...	Global	Complete		2018-06-22 16:52:48	admin		

2. Select your update set from the list.
3. Set the ***State*** to *Complete* and click **[Update]**.
4. From the **Update Sets** page, select your completed update set from the list.
5. Under the ***Related Links*** section, click **Export to XML**.



Name: ScienceLogic IdentificationEngine (pre-set (VMWare))

State: Complete

Parent:

Release date:

Install date: 2018-06-22 16:51:16

Installed from:

Description:

Application: Global

Created: 2018-06-22 16:51:15

Created by: admin

Merged to:

Update set = ScienceLogic IdentificationEngine (pre-set (VMWare))

Created	Type ▲	View	Target name	Updated by	Remote update set	Action
2018-06-22 16:51:15	CMDB Metadata Containment Rules		cmdb_ci_vmware_instance	admin		INSERT_OR_UPDATE
2018-06-22 16:51:15	CMDB Metadata Containment Rules		cmdb_ci_vcenter_datacenter	admin		INSERT_OR_UPDATE
2018-06-22 16:51:15	CMDB Metadata Containment Rules		cmdb_ci_vcenter_datastore	admin		INSERT_OR_UPDATE
2018-06-22 16:51:15	CMDB Metadata Containment Rules		cmdb_ci_vcenter_dvs	admin		INSERT_OR_UPDATE

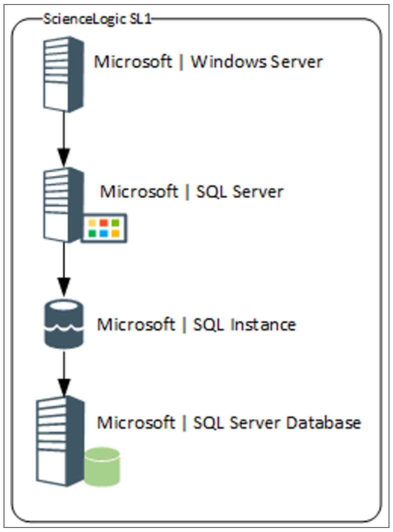
6. Save the downloaded XML file.

Configuring Customer CI Relation Overrides

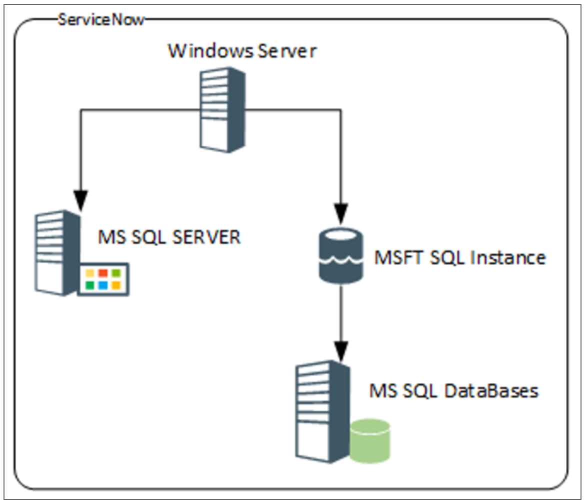
When you are mapping Device Classes and attributes, you might find that the relationship mapping between SL1 and ServiceNow might be complex, or both applications construct device relationships in very different ways. In the "Sync Devices from SL1 to ServiceNow" integration application, you can use the ***customer_ci_relation_overrides*** field on the ***Configuration*** pane to override the relationship linking and directly control the link between Device Classes and attributes.

WARNING: This process is intended for advanced users that are familiar with how SL1 and ServiceNow construct device relationships.

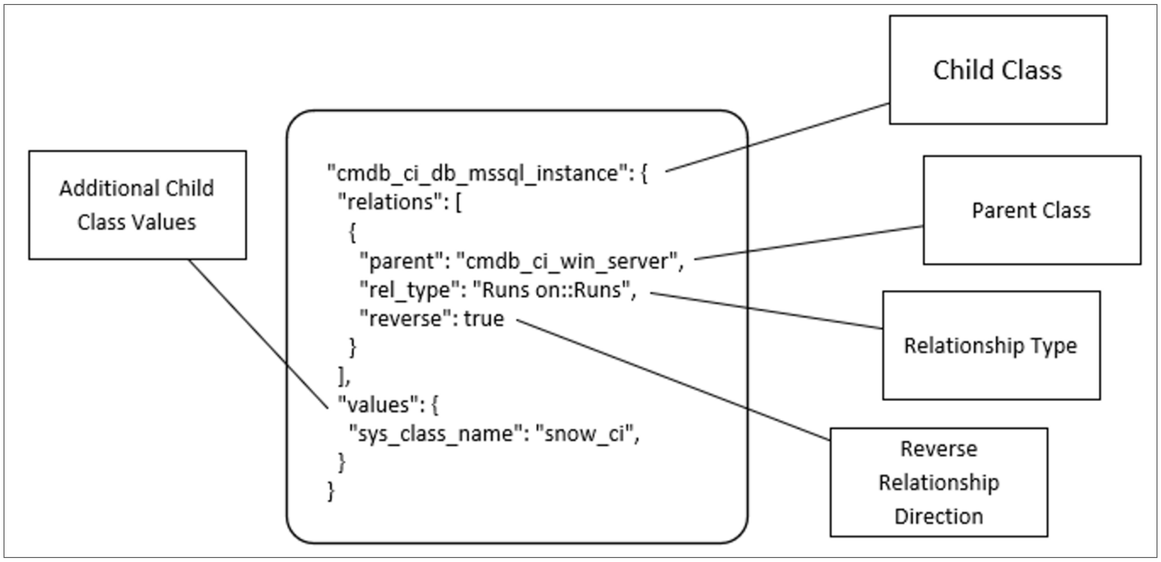
In the following example, the relationship structure in SL1 is linear :



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:



```
"cmdb_ci_db_mssql_instance": {
 "relations": [
  {
   "parent": "cmdb_ci_win_server",
   "rel_type": "Runs on::Runs",
   "reverse": true
  }
 ],
 "values": {
  "sys_class_name": "snow_ci",
 }
}
```

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 SL1 to ServiceNow" integration application.

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", "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", "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", "instance_name": "name"}
   }
}
```

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 SL1: CMDB & Incident Automation" application (also called the Certified application).

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

TIP: For more information about ServiceNow domain separation, see ***Using ServiceNow Domain Separation with the Integration Service.***

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

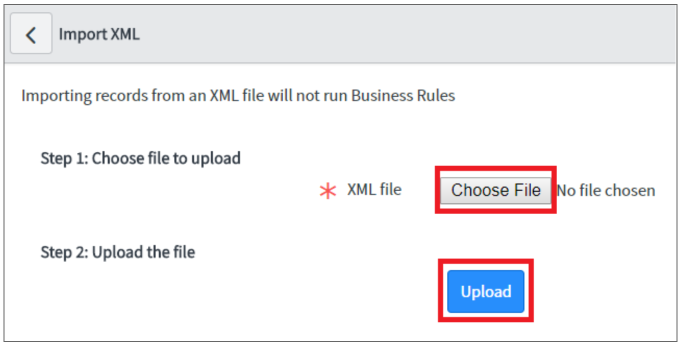
For more information about how the Identification Engine handles incoming payloads in domain-separated systems, see the following ServiceNow Knowledge Base article: KB0695949.

The payload and the user domain must match, or the ServiceNow Identification Engine (IDE) will by default insert the CMDB record. Safeguards within the Integration Service Device Sync integration application were put in place for payloads that have relationships. The integration 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. Retrieve the latest update set from your ScienceLogic representative and download the file.
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]**.



Import XML

Importing records from an XML file will not run Business Rules

Step 1: Choose file to upload

* XML file: No file chosen

Step 2: Upload the file

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.

WARNING: 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 SL1 to ServiceNow" integration application. This option ensures that the Integration Service 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 the Integration Service

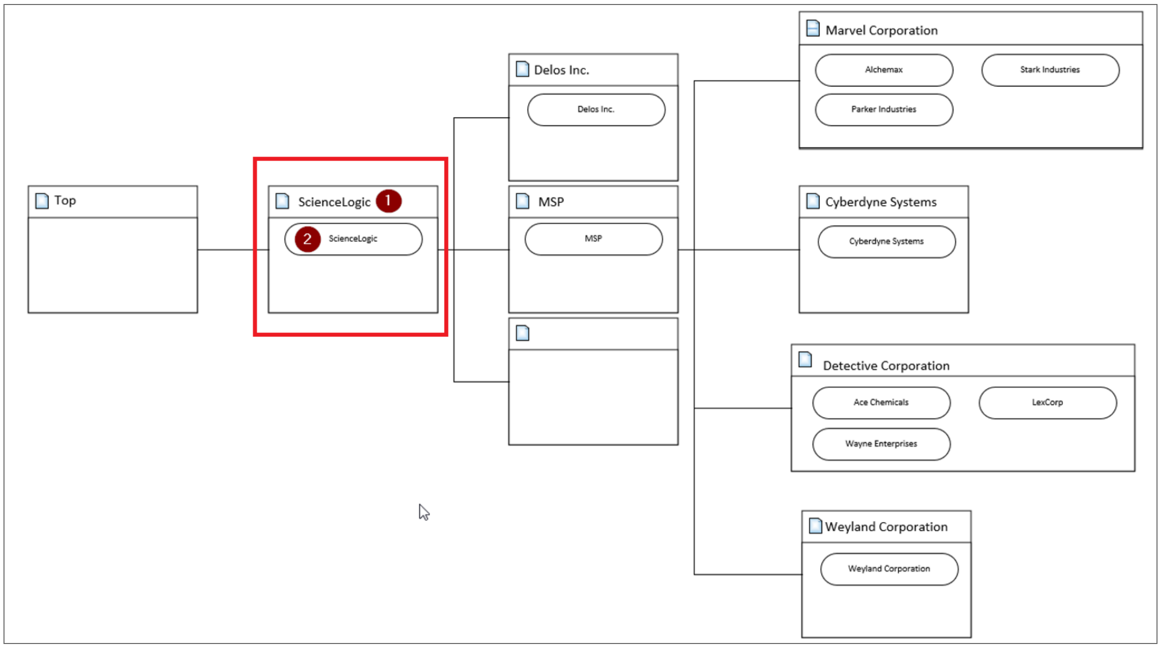
The following topics provide more information about ServiceNow domain separation and how it relates to the Integration Service. For more information, see Domain separation in the ServiceNow Documentation.

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.

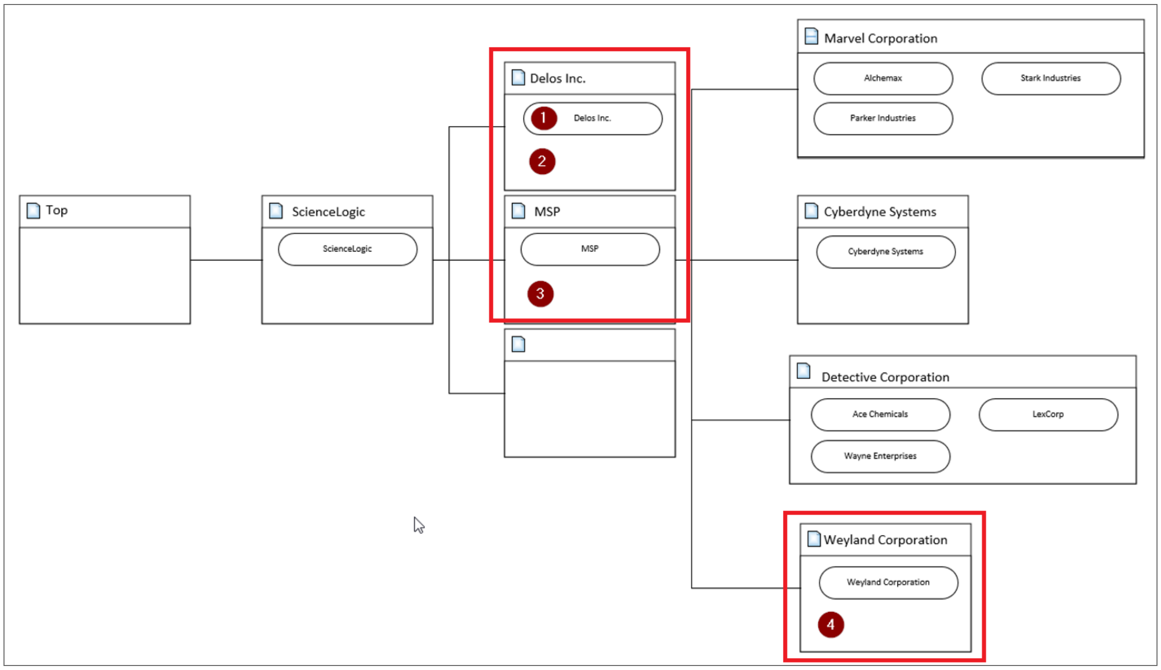
Example 1

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

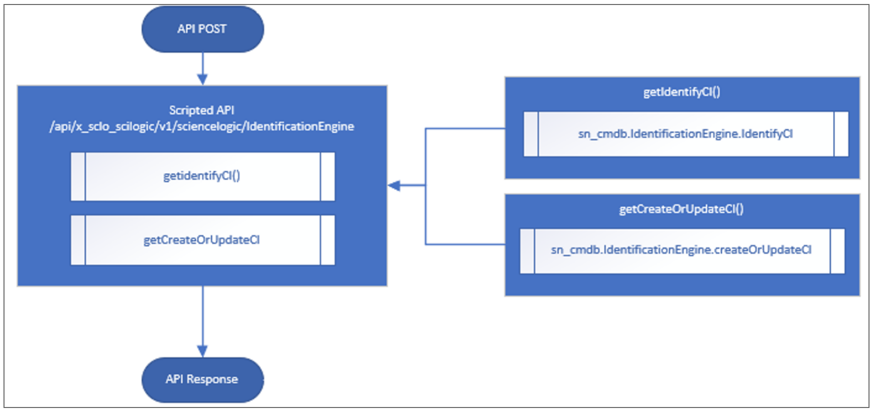
In the following example, **Delos Inc. (1)** is the company within the **Delos Inc.** domain. The Integration Service service account is associated with the **Delos Inc. (1)** company. The **Delos Inc.** domain has no children domains, and if domain visibility is not assigned, the Integration Service 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.

Workflow

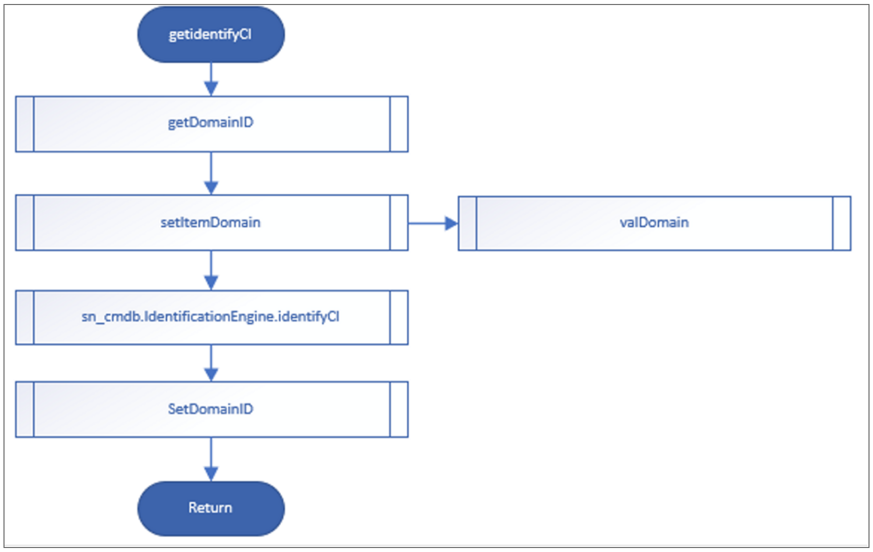
The API endpoint is based on the API query parameter “test” being true or not, which determines which Identification Engine API resource should be used. There are two primary avenues supplied with this REST resource: "createOrUpdateCI" or "identifyCI", and the only difference is that "identifyCI" does not commit the results:


API POST
Scripted API
/api/x_sclo_scilogic/v1/sciencelogic/IdentificationEngine
getidentifyCI()
getCreateOrUpdateCI
getIdentifyCI()
sn_cmdb.IdentificationEngine.IdentifyCI
getCreateOrUpdateCI()
sn_cmdb.IdentificationEngine.createOrUpdateCI
API Response

The "getCreateOrUpdateCI" function uses the following workflow:

1. Retrieves the current session Domain ID (`sys_id`).
2. Sets the user Domain ID by creating an array of domain `sys_id` values and returning only the unique domains, or setting the domain if the array has only one unique domain.
3. Submits the JSON formatted string to "createOrUpdateCI()" or "identifyCI()" API.
4. Sets the user's Domain ID back to the original session ID.

The following image shows this workflow:



Chapter 3

Configuring Integrations for the CMDB Synchronization PowerPack

Overview

This chapter describes the how to configure and run the various Integration Service integration applications contained in the *ServiceNow CMDB* Synchronization PowerPack.

An ***integration application*** is a JSON object that includes all the information required for executing an integration on the Integration Service platform. An integration application combines a set of steps that execute a workflow. You can configure the parameters in the integration application to customize the sync process.

TIP: While an integration application is running on the **Integrations** page, you will see a dark green, horizontal line at the top of the page until the process completes.

This chapter covers the following topics:

Creating and Aligning a Configuration Object

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

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

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

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

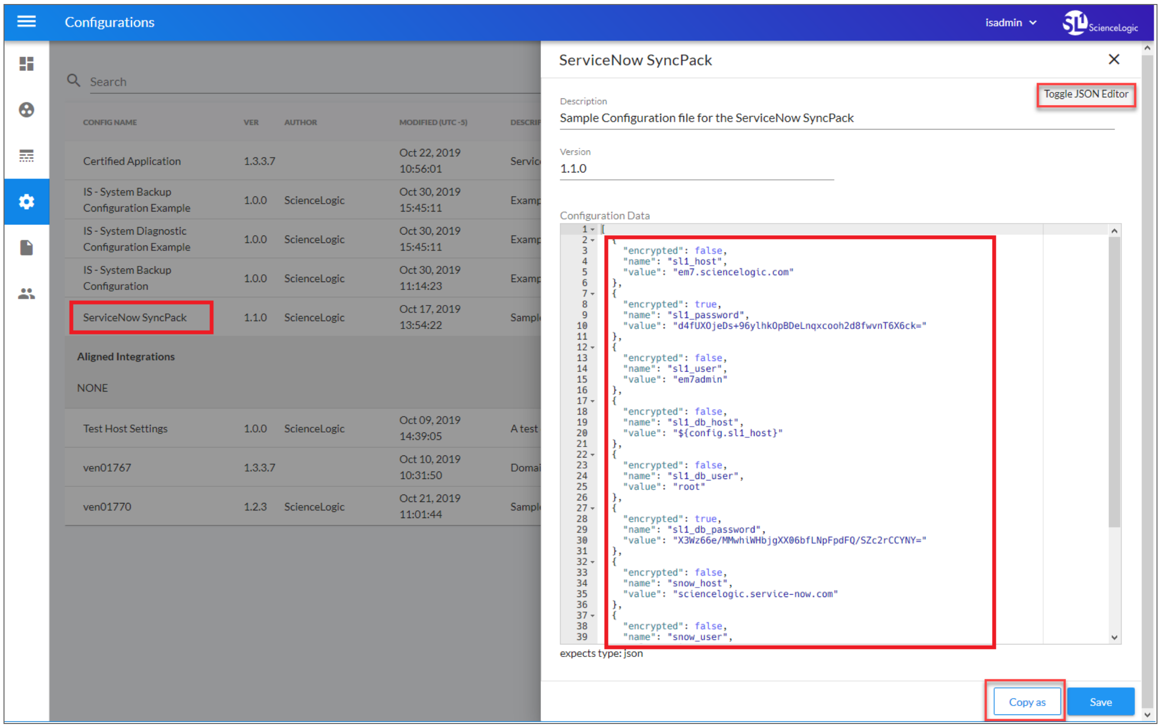
Creating a Configuration Object

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

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

1. In the Integration Service user interface, go to the **Configurations** page (⚙).

2. Click the **[Actions]** button (⋮) for the "ServiceNow SyncPack" configuration object and select *Edit*. The **Configuration** pane appears:



CONFIG NAME	VER	AUTHOR	MODIFIED (UTC -5)	DESCRIP
Certified Application	1.3.3.7		Oct 22, 2019 10:56:01	Servic
IS - System Backup Configuration Example	1.0.0	ScienceLogic	Oct 30, 2019 15:45:11	Examp
IS - System Diagnostic Configuration Example	1.0.0	ScienceLogic	Oct 30, 2019 15:45:11	Examp
IS - System Backup Configuration	1.0.0	ScienceLogic	Oct 30, 2019 11:14:23	Examp
ServiceNow SyncPack	1.1.0	ScienceLogic	Oct 17, 2019 13:54:22	Sampl
Aligned Integrations NONE				
Test Host Settings	1.0.0	ScienceLogic	Oct 09, 2019 14:39:05	A test
ven01767	1.3.3.7		Oct 10, 2019 10:31:50	Domai
ven01770	1.2.3	ScienceLogic	Oct 21, 2019 11:01:44	Sampl

ServiceNow SyncPack

Description
Sample Configuration file for the ServiceNow SyncPack

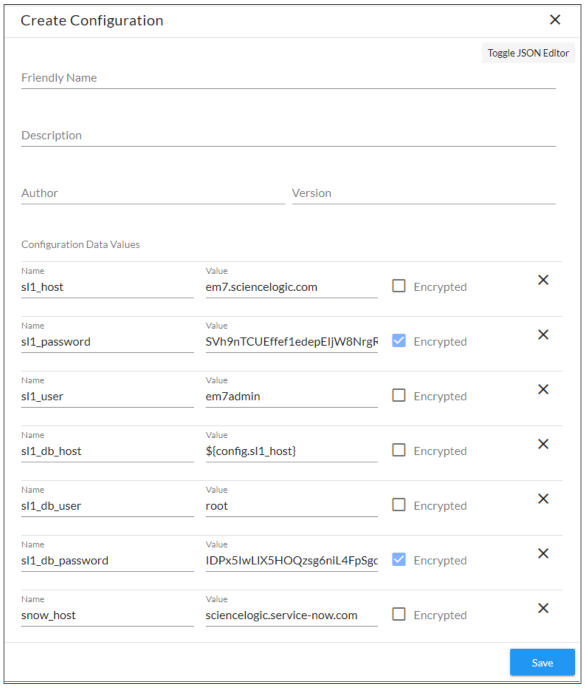
Version
1.1.0

Configuration Data

```
[
  {
    "encrypted": false,
    "name": "sl1_host",
    "value": "em7.sciencelogic.com"
  },
  {
    "encrypted": true,
    "name": "sl1_password",
    "value": "d4fUXOjeDs+96ylhkOpBDeLnqxcooh2d8fwvnT6X6ck="
  },
  {
    "encrypted": false,
    "name": "sl1_user",
    "value": "em7admin"
  },
  {
    "encrypted": false,
    "name": "sl1_db_host",
    "value": "${config.sl1_host}"
  },
  {
    "encrypted": false,
    "name": "sl1_db_user",
    "value": "root"
  },
  {
    "encrypted": true,
    "name": "sl1_db_password",
    "value": "X3Wz66e/MMwhiWHbjgXX06bfLNpFpdFQ/SZc2rCCYNY="
  },
  {
    "encrypted": false,
    "name": "snow_host",
    "value": "sciencelogic.service-now.com"
  },
  {
    "encrypted": false,
    "name": "snow_user",
```

expects type: json

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



Create Configuration

Friendly Name

Description

Author

Version

Configuration Data Values

Name	Value		
sl1_host	em7.sciencelogic.com	☐ Encrypted	✕
sl1_password	SVh9nTCUEffef1edepEIjW8NrgR	☑ Encrypted	✕
sl1_user	em7admin	☐ Encrypted	✕
sl1_db_host	${config.sl1_host}	☐ Encrypted	✕
sl1_db_user	root	☐ Encrypted	✕
sl1_db_password	IDPx5IwLIX5HOQzsg6niL4FpSgc	☑ Encrypted	✕
snow_host	sciencelogic.service-now.com	☐ Encrypted	✕

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. Click **[Toggle JSON Editor]** to show the JSON code.
6. In the ***Configuration Data*** field, be sure to include the required block of code to ensure that the integration applications aligned to this configuration object do not fail:

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

   For example:

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

NOTE: If you are using SL1 with an External Database (SL1 Extended architecture or a cloud-based architecture), update the `"value"` of that block of code to be the host of your database. This field accepts IP addresses. For example: `"value": "db.sciencelogic.com"`. If you are *not* using the SL1 Extended architecture or a cloud-based architecture, you do not need to make any changes to the block of code other than pasting the code into the configuration object.

7. In the ***Configuration Data*** field, update the default variable definitions to match your Integration Service configuration.

TIP: The "ServiceNow SyncPack" configuration object contains all of the required variables. Simply update the variables from that object to match your SL1 and ServiceNow settings.

NOTE: The ***region*** value is a user-defined variable that will identifies your SL1 instance within ServiceNow.

8. To create a configuration variable, 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, the Integration Service 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 Integration Service 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.

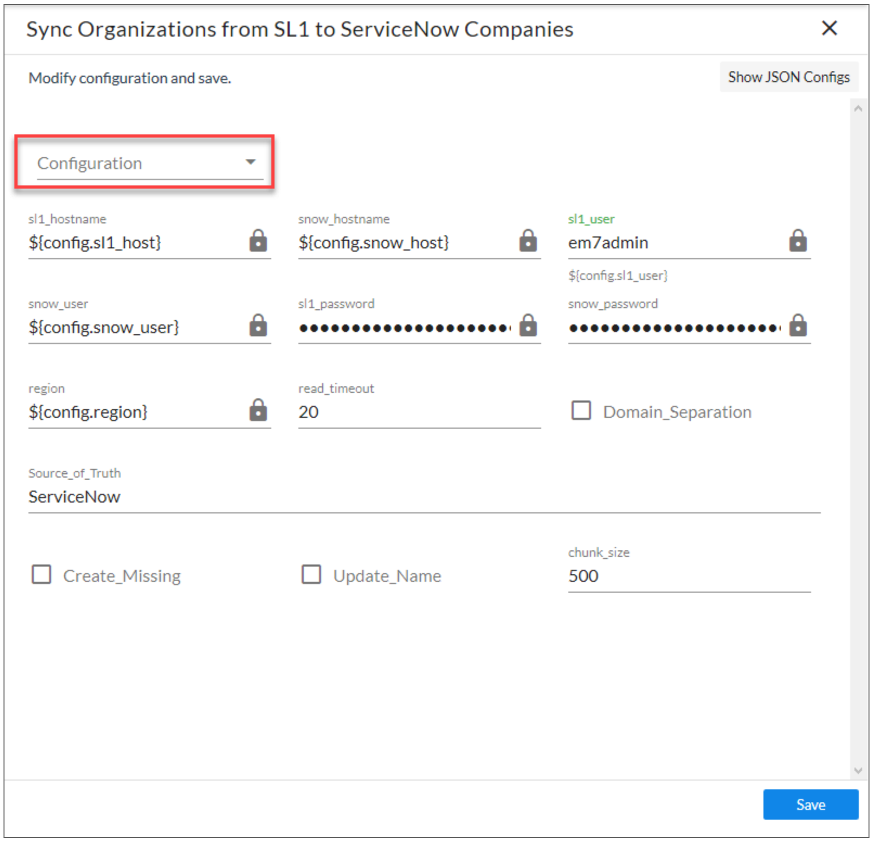
9. Click **[Save]**. You can now align this configuration object with one or more integration applications.

Aligning a Configuration Object

Before you can run the integration applications in this Synchronization PowerPack, you must first "align" a configuration object with the integration application you want to use.

To align a configuration object with an integration application:

1. From the **Integrations** page of the Integration Service user interface, open the relevant integration application and click **[Configure]** (⚙). The **Configurations** pane for that application appears:



Sync Organizations from SL1 to ServiceNow Companies

Modify configuration and save.

sl1_hostname: ${config.sl1_host}

snow_hostname: ${config.snow_host}

sl1_user: em7admin
${config.sl1_user}

snow_user: ${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

region: ${config.region}

read_timeout: 20

□ Domain_Separation

Source_of_Truth: ServiceNow

□ Create_Missing

□ Update_Name

chunk_size: 500

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

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

Syncing Organizations

If your ServiceNow configuration uses domain separation, the first sync you should run on a new Integration Service system is an ***Organization Sync***. This sync uses the "Sync Organizations from SL1 to ServiceNow" integration application to sync *organizations* from SL1 with ServiceNow *companies*. Be sure to select the ***Domain Separation*** option on the **Configuration** pane in the "Sync Organizations from SL1 to ServiceNow" integration application. You must also select *ServiceNow* from the ***Source of Truth*** field.

If your ServiceNow configuration does *not* use domain separation, ScienceLogic recommends that your first sync on a new Integration Service system is an ***Organization Sync*** as well, but you should not select the ***Domain Separation*** option on the **Configuration** pane in the "Sync Organizations from SL1 to ServiceNow" integration application.

Organization Sync uses the "Sync Organizations from SL1 to ServiceNow" integration application to pull *organizations* from SL1 and sync them with ServiceNow *companies*.

For Domain-separated ServiceNow Environments Only

If your ServiceNow environment is *domain-separated*, where the data, processes, and administrative tasks have been organized into logical groupings called *domains*, then the first sync you should run on a new Integration Service system is an ***Organization Sync***. .

For a domain-separated ServiceNow environment, you must update three fields in ServiceNow for the companies you want to sync. Because these fields do not display by default on the **Companies** page in ServiceNow, navigate to the **Companies** page, click the Update Personalized List icon (⚙), and add the ***SL1 Monitored*** and ***SL1 Region*** columns to that page.

In ServiceNow, update the following fields:

- ***SL1 Monitored***. Set to *true*.
- ***SL1 Region***. Set to match the ***region*** value in the configuration object aligned with the "Sync Organizations from SL1 to ServiceNow" integration application in the Integration Service user interface. See step 3, below.
- ***SL1 ID***. Set to match the ***Organization ID*** in SL1.

You must also configure and successfully run the "Sync Organizations from SL1 to ServiceNow" integration application *before* you can sync any additional CI items or devices.

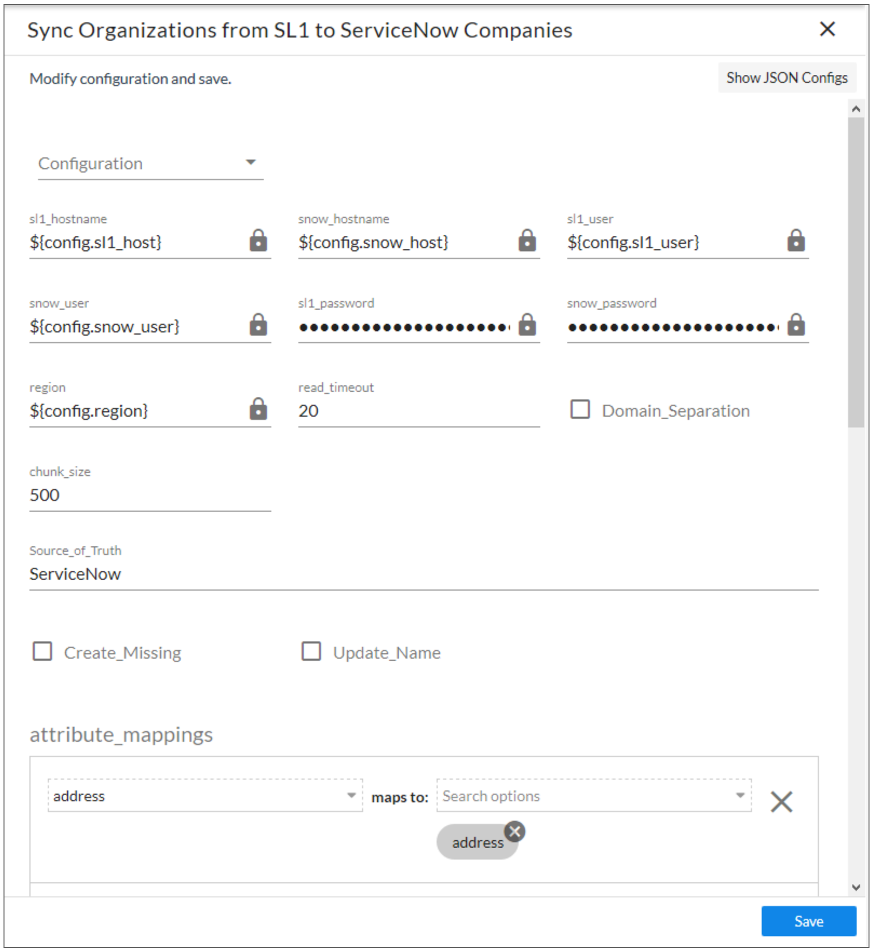
Configuring Organization Sync

If your ServiceNow configuration does *not* use domain separation, ScienceLogic recommends that your first sync on a new Integration Service system is an Organization Sync as well, but you should *not* select the ***Domain Separation*** option on the **Configuration** pane in the "Sync Organizations from SL1 to ServiceNow" integration application.

To sync SL1 organizations with ServiceNow companies:

1. In the Integration Service user interface, go to the **Integrations** page and select the "Sync Organizations from SL1 to ServiceNow" integration application. The **Integration Application** page for that application appears.

2. Click **[Configure]** (⚙). The **Configuration** pane appears:



Sync Organizations from SL1 to ServiceNow Companies

Modify configuration and save.

sl1_hostname: ${config.sl1_host}

snow_hostname: ${config.snow_host}

sl1_user: ${config.sl1_user}

snow_user: ${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

region: ${config.region}

read_timeout: 20

□ Domain_Separation

chunk_size: 500

Source_of_Truth: ServiceNow

□ Create_Missing

□ Update_Name

attribute_mappings

address maps to: Search options

address

3. Complete the following fields, as needed:

   - ***Configuration***. Select the configuration object with the relevant SL1 and ServiceNow credentials to align with this integration 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 integration application. The region value must match the value in the ***SL1 Region*** field in ServiceNow. If you need to update this value, you will need to define the ***region*** variable in the configuration object that is aligned with this integration 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 integration application should wait for a response before timing out. The default is 20 seconds.
   - ***Domain_Separation***. Select this option *only* 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.
   - ***chunk_size***. Specify the number of organizations to include in each chunk sent to ServiceNow when you run this integration application. The default is 500.
   - ***Source_of_Truth***. Select whether you want to use data from ServiceNow or ScienceLogic as the "source of truth" when this integration application encounters duplicate data or data collisions.
     - If you select *ServiceNow*, you must specify the values in the ***SL1 Monitored*** and ***SL1 Region*** fields in ServiceNow. Because these fields do not display by default on the **Companies** page in ServiceNow, navigate to the **Companies** page, click the Update Personalized List icon (⚙), and add the ***SL1 Monitored*** and ***SL1 Region*** columns to that page. If your ServiceNow configuration uses domain separation, you must select *ServiceNow* as the source of truth.
     - If you select *ScienceLogic*, you do not need to do anything else related to this field.
   - ***Create_Missing***. Select this option if you want the Integration Service to create a new organization or company if that record is missing, based on your selection in the *Source_of_Truth* field. By default, this option is not selected.
   - ***Update_Name***. This option addresses the situation where the Integration Service finds a match with an organization and a company, but the names do not match. This option updates a company or organization name based on your selection in the *Source_of_Truth* field, below. For example, if you selected *ScienceLogic* as the source of truth, the Integration Service uses the company name from ScienceLogic as the updated name. By default, this option is not selected.

4. In the **attribute_mappings** section, you can edit or create a mapping for any other company attributes, such as address and contact information, that you want to sync between SL1 (the first column) and ServiceNow (the second column). A set of company attributes are already mapped by default.

NOTE: You can use Jinja2 Templates in fields that are aligned with the "Source of Truth" you selected (the left column is for SL1, and the right column is for ServiceNow). For more information, see *Using a Jinja2 Template*.

5. To *edit* an existing company attribute in the **attribute_mappings** section, click the attribute name and either select an attribute from the list or type a new name for the attribute. Press **[Enter]** after editing the attribute to make sure your changes are saved.

TIP: Use the **[Tab]** button to move down through the list of options in a dropdown list, press **[Shift]**+ **[Tab]** to move up, and press **[Enter]** to select a highlighted option.

6. To *create* a company attribute in the **attribute_mappings** section, click the **[Add Mapping]** button at the bottom of the section, type a name for the attribute in the first field, and select one or more ServiceNow attributes to which the SL1 attribute should sync in the ***maps to*** field. Press **[Enter]** after editing the attribute to make sure your changes are saved.

NOTE: When an attribute value is "0" in SL1, the corresponding field in ServiceNow might display as empty.

7. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
8. Click **[Run]** (▶) to run the integration application.
9. When the integration application completes, open the **Step Log** and review the log messages for the "Process Organizations" step to see if any Company or Organization records were created. As needed, select the other steps to review the logs on the **Step Log** for those steps.

TIP: SL1 Organizations that are synced to a ServiceNow Company will have the ***crm_id*** field on the **[Properties]** tab for that organization populated with the ServiceNow Company ***sys_id*** variable.

Syncing Devices from SL1 to ServiceNow

The "Sync Devices from SL1 to ServiceNow" integration application syncs devices and virtual device relationships from SL1 to ServiceNow. You can also sync devices based on organization and collector group.

The Device Sync process use rules or "mappings" that you can define in the "Sync Devices from SL1 to ServiceNow" integration application. These mappings connect an **SL1 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 SL1 to ServiceNow" integration application can also collect manufacturer and model attributes from asset records aligned with devices in SL1 and sync that information with ServiceNow. The Integration Service only populates the manufacturer and model attributes if the values exist in ServiceNow CIs; the Integration Service does not create new manufacturer values in ServiceNow. The "Sync Devices from SL1 to ServiceNow" integration application uses the ***sys_id*** field as a reference when syncing manufacturer and model information between SL1 and ServiceNow. For more information, see *Default Device Attribute Mappings*.

NOTE: If you have merged devices in SL1, the "Sync Devices from SL1 to ServiceNow" integration application syncs the physical device record from SL1 to ServiceNow. However, if you have other data sources syncing into the ServiceNow CMDB, ScienceLogic recommends not merging devices in SL1 as this could cause issues when the other data sources insert/update into the CMDB.

Common Fields Used by Device Sync

The "Sync Devices from SL1 to ServiceNow" integration application uses the following ServiceNow fields to determine which devices to sync from SL1 to ServiceNow:

- ***SL1 Monitored***. This field displays a Boolean (true or false) value that is impacted by whether the device is in SL1 or not. The device being found in ServiceNow depends on the ***SL1 Monitored*** field. The device being found in SL1 depends on the class mappings defined in the "Sync Devices from SL1 to ServiceNow" integration application.
  - If the CI is in ServiceNow *and* the device is in SL1 , the ***SL1 Monitored*** flag is set to *true*.
  - If the CI is in ServiceNow but the device is *not* in SL1, the ***SL1 Monitored*** field is set to *false*.
- ***SL1 Region***. This field represents an ID for the SL1 instance or instances being synced to the ServiceNow instance. The ***SL1 Region*** field is determined by the user when configuring the IS applications. In a multi-SL1 environment, ScienceLogic recommends that you make the ***SL1 Region*** field descriptive so the ServiceNow user knows from which SL1 stack the CI originated.
  - If the ***SL1 Region*** field is defined as an identifier by the CI class, ServiceNow will create new CI records with the new ***SL1 Region*** value, and the user must manually delete the duplicate CIs in the old ***SL1 Region*** field.
  - If the ***SL1 Region*** field is *not* defined as an identifier by the CI class, ServiceNow will not treat these devices as new CIs, and the ***SL1 Region*** field will be automatically updated.

NOTE: Changing the ***SL1 Region*** value after an initial run of the "Sync Devices from SL1 to ServiceNow" application will have differing results depending on the service rules defined in ServiceNow that dictate reconciliation of the CI. If you change the ***SL1 Region*** value, you will ll need to run "Sync Devices from SL1 to ServiceNow" twice: once to align the CIs with the new region, and a second time to enable the Integration Service to re-cache the newly updated CIs in the region.

Running a Device Sync

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

- ***Cache ServiceNow CIs and SL1 Device Classes***. Reads all existing SL1 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 SL1 to ServiceNow" integration application.
- ***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 SL1 to ServiceNow***. Syncs devices and virtual device relationships from SL1 to ServiceNow. In previous versions, this integration application was named "ScienceLogic To ServiceNow Device Sync using GraphQL".

To sync SL1 devices with ServiceNow:

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

   NOTE: If you change any of the containment rules or hosting rules in ServiceNow, you will need to run "Cache ServiceNow CIs and SL1 Device Classes" again. For more information, see *Configuring Service Rules for Device Sync*.

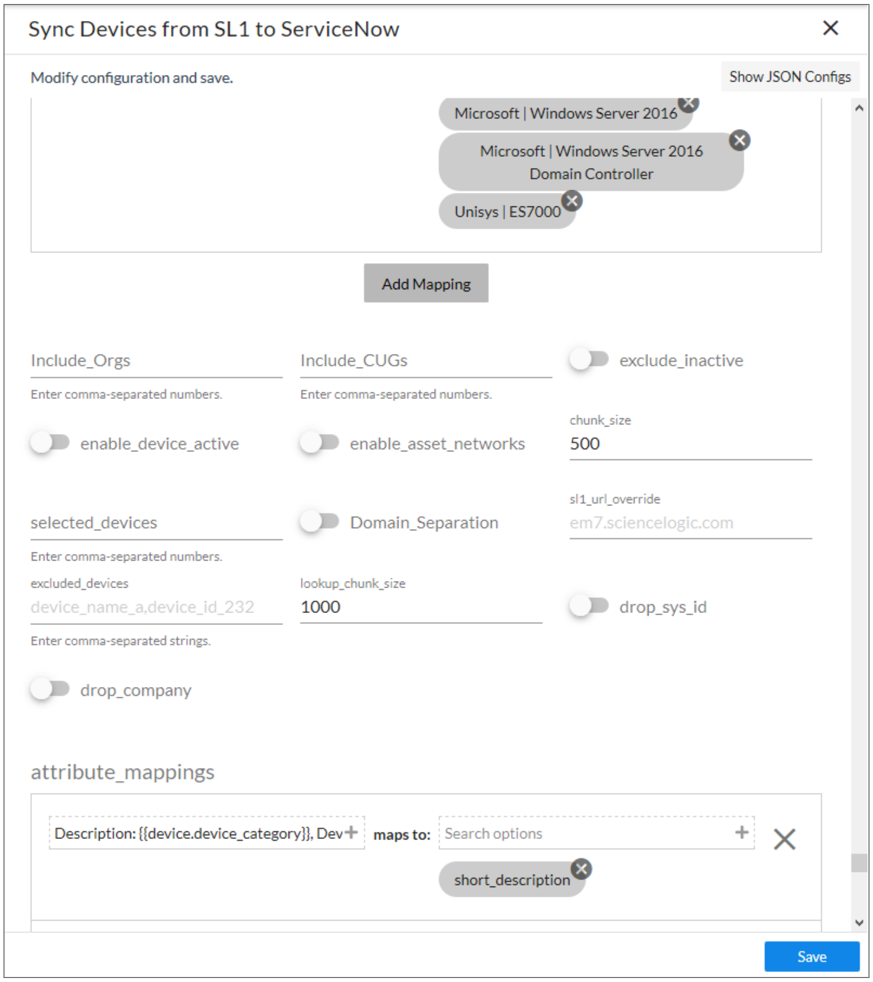
2. Select the "Generate Required CI Relations for ServiceNow" integration application from the **Integrations** page, click **[Configure]** (⚙), align a configuration object, and then click **[Run]** (▶).

   NOTE: The Integration Service 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" integration application. Any mappings you add to this integration application will *overwrite* mappings in the "Sync Devices from SL1 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 Integration*.

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

4. Select the "Sync Devices from SL1 to ServiceNow" integration application from the **Integrations** page and click **[Configure]** (⚙). The **Configuration** pane appears:



Sync Devices from SL1 to ServiceNow

Modify configuration and save.

Show JSON Configs
Microsoft | Windows Server 2016
Microsoft | Windows Server 2016 Domain Controller
Unisys | ES7000
Add Mapping

Include_Orgs
Enter comma-separated numbers.

Include_CUGs
Enter comma-separated numbers.

exclude_inactive

enable_device_active

enable_asset_networks

chunk_size: 500

selected_devices
Enter comma-separated numbers.

Domain_Separation

sl1_url_override: em7.sciencelogic.com

excluded_devices: device_name_a,device_id_232
Enter comma-separated strings.

lookup_chunk_size: 1000

drop_sys_id

drop_company

attribute_mappings

Description: {{device.device_category}}, Dev maps to: Search options

short_description

5. Complete the following fields, as needed:

- ***Configuration***. Select the configuration object with the relevant SL1 and ServiceNow credentials to align with this integration 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 integration application. The region value must match the value in the ***SL1 Region*** field in ServiceNow. If you need to update this value, you will need to define the ***region*** variable in the configuration object that is aligned with this integration 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 integration application should wait for a response before timing out. The default is 20 seconds. If the Device Sync continually times out, you might want to increase the timeout to 120 seconds, as this sync can be data-intensive.

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

- ***Include_Orgs***. If you want to include SL1 Organizations in the device sync, add the Organization IDs from SL1 in this field, separated by commas. Leave this field empty to sync all SL1 Organizations.
- ***Include_CUGs***. If you want to include SL1 Collector Groups (CUGs) in the device sync, add the Collector Group IDs from SL1 in this field, separated by commas. Leave this field empty to sync all SL1 Collector Groups.
- ***exclude_inactive***. Select this option to prevent syncing devices to ServiceNow that are disabled, unavailable, or in maintenance. By default, this field is not selected.
- ***enable_device_active***. Select this option to enable the **Device Active** block in the device GraphQL query, which contains information about the active state of the SL1 device. By default, this field is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template. For more information, see ***Using a Jinja2 Template***.
- ***enable_asset_networks***. Select this option to enables the **assetNetworks** block in the device GraphQL query, which returns a list of asset networks. By default, this field is not selected. Accessing this data in the attribute mappings requires a Jinja2 Template. For more information, see ***Using a Jinja2 Template***.

> WARNING: Please note that enabling this option might cause performance issues on the SL1 side.

- ***chunk_size***. Specify the number of devices to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500 devices.
- ***selected_devices***. If you want to sync a sub-set of all discovered devices, type a comma-separated list of the Device IDs from SL1 for only the devices that you want to sync. Leave this field empty to sync all SL1 devices.
- ***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, the Integration Service syncs the ServiceNow Company ***sys_id*** to with the corresponding SL1 Organization.
- ***sl1_url_override***. Update this field if you want to use an URL that is different from the standard SL1 URL that gets sent to the ServiceNow CI record. Optional.
- ***excluded_devices***. Type a list of comma-separated device names or device IDs for any devices that you want to exclude from the device sync. Optional.

- ***lookup_chunk_size***. Specify the number of devices in each chunk when pulling the device cache from Couchbase. The default is 1000 devices.
- ***drop_sys_id***. Select this option if you want to remove the ***sys_id*** in existing CIs from the sync. If you set ***drop_sys_id*** to *true*, make sure that ServiceNow can correctly identify and correlate your existing CIs with the properties that are available.
- ***drop_company***. Select this option if you want to remove the ***sys_id*** in existing Companies from the sync. Selecting this option has no effect if you selected the ***Domain_Separation*** option for this integration application.

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

- ***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 *Configuring Customer CI Relation Overrides*.

NOTE: You can also install the "ScienceLogic_Identification_Engine_VMware.xml" update set for ServiceNow to get access to additional relationships and overrides that you might need to be able to sync VMware trees. Depending on the plug-ins and custom configurations in your environment, you might need additional relationships. These mapping do not handle making classes independent, as that is based on the specific user environment. Identifier rules are not defined in this update set because they depend on the data in the SL1 environment. For more information about VMware relationships in ServiceNow, see https://docs.servicenow.com/bundle/orlando-it-operations-management/page/product/discovery/reference/r_VCenterDataCollected.html.

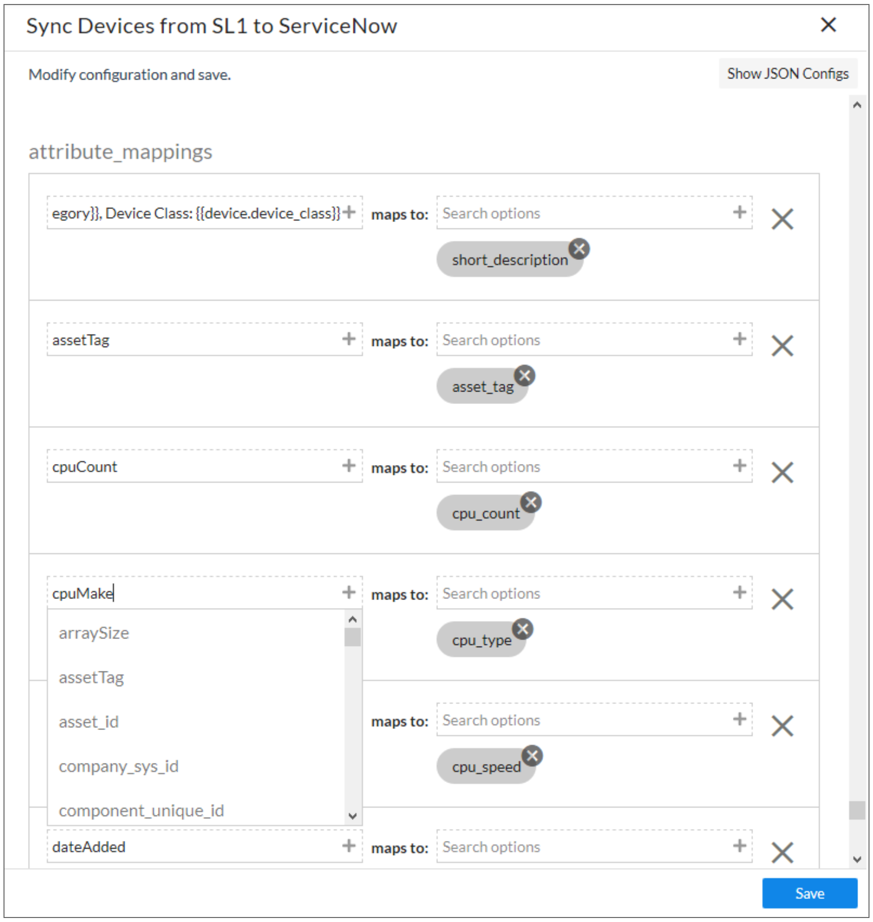
- ***Simulation_Mode***. Select this option if you want to perform a simulated run of this integration application to show you the potential results of that run.

6. Scroll to the **mappings** section to view the mappings between SL1 Device Classes and ServiceNow CI classes. This section is pre-loaded with a large number of default device mappings. For a complete list of the default mappings and a list of available mappings, see *Device Attribute Mappings*.

NOTE: The "Sync Devices from SL1 to ServiceNow" integration application will *only* sync a device from SL1 if the Device Class for that device is mapped to a ServiceNow CI class in **mappings** section. The default mappings in this section do not cover all technologies, however, and syncing additional technologies from SL1 to ServiceNow might require additional research to understand the class structure.

7. To create a custom mapping for the device sync, click **[Add Mapping]** at the bottom of the section. ServiceNow CI classes display on the left, and SL1 device classes display on the right. You can map a single ServiceNow CI class with multiple SL1 devices classes.

8. Scroll to the **attribute_mappings** section, which in previous versions was named the **additional_attributes** section. In this section, you can create a mapping for any other custom device attributes you want to sync between SL1 (the first column) and ServiceNow (the second column):



Sync Devices from SL1 to ServiceNow

Modify configuration and save.

attribute_mappings

egory}}, Device Class: {{device.device_class}} maps to: short_description

assetTag maps to: asset_tag

cpuCount maps to: cpu_count

cpuMake maps to: cpu_type

maps to: cpu_speed

dateAdded maps to:

NOTE: All custom attributes for each SL1 device are automatically synced.

TIP: You can use a Jinja2 Template for device attribute fields on the SL1 side (the left column). For more information, see *Using a Jinja2 Template*.

9. To *edit* an existing device attribute in the **attribute_mappings** section, click the attribute name and either select an attribute from the list or type a new name for the attribute. Press **[Enter]** after editing the attribute to make sure your changes are saved.

10. To *create* a custom device attribute in the **attribute_mappings** section, click the **[Add Mapping]** button at the bottom of the section, type a name for the attribute in the first field, and select one or more ServiceNow attributes to which the SL1 attribute should sync in the ***maps to*** field. Press **[Enter]** after editing the attribute to make sure your changes are saved.

    NOTE: When an attribute value is "0" in SL1, the corresponding field in ServiceNow might display as empty.

11. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
12. Run the "Sync Devices from SL1 to ServiceNow" integration application. If this is the first time you are running this integration application, run it a second time to build the internal cache.
13. When the integration application completes, open the **Step Log** and review the log messages for the "Compare SL1 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.

NOTE: Depending on the number of devices you are syncing to ServiceNow, it might take a few minutes for all devices to get fully synced to the CMDB. You might notice after running device sync that the number of SL1 Monitored CIs continues to increase after each refresh. This is expected behavior due to payload chunking in ServiceNow. ServiceNow processes each payload as an individual chunk.

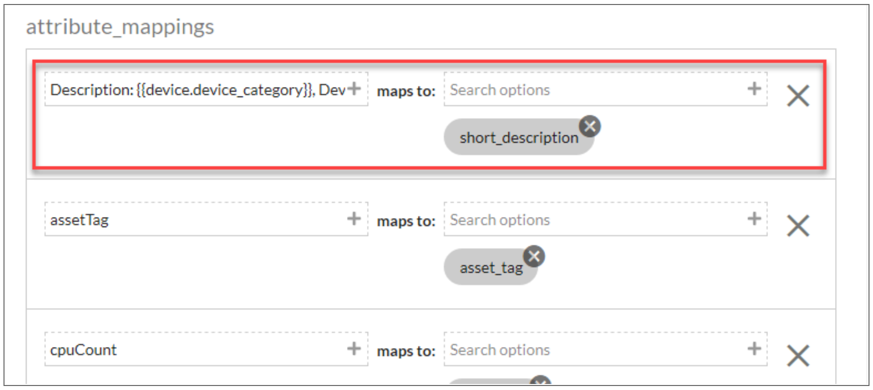
Using a Jinja2 Template

The attribute mappings in both the "Sync Devices from SL1 to ServiceNow" and the "Sync CI Attributes from ServiceNow to SL1" integration applications now support Jinja2 Templates, which let you sync complex, concatenated (linked) fields from SL1 to ServiceNow. For example, you can add these complex values in the SL1 side of the **attribute_mappings** section of the **Configuration** pane for the "Sync Devices from SL1 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 SL1 to ServiceNow" integration application, the SL1 side can be a Template. In the "Sync CI Attributes from ServiceNow to SL1" integration application, the ServiceNow side can be a Template.

Example Template

This example is included in the "Sync Devices from SL1 to ServiceNow" integration application as the first default value in the **attribute_mappings** section of the **Configuration** pane:


attribute_mappings

Description: {{device.device_category}}, Dev maps to: short_description

assetTag maps to: asset_tag

cpuCount maps to:

This Template, when used on the SL1 side of the **attribute_mappings** section, populates the **short_description** field in ServiceNow:

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

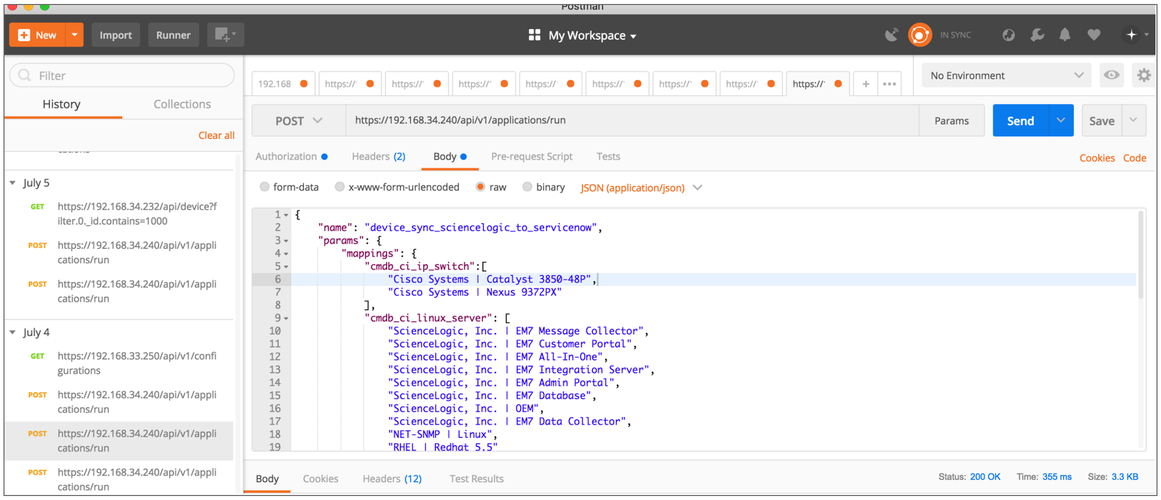
In the above example, for a device with a **category: Testing** and a Device Class of **Testing | Testing**, the end result would be **Description: Testing, Device Class: Testing | Testing**, which will be posted to the **short_ description** field in ServiceNow.

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

Adding Device Mappings with Postman

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*.

The following image displays an example of using Postman to send the mapping data to Integration Service:



July 5

GET https://192.168.34.232/api/device?filter.0._id.contains=1000

POST https://192.168.34.240/api/v1/applications/run

POST https://192.168.34.240/api/v1/applications/run

July 4

GET https://192.168.33.250/api/v1/configurations

POST https://192.168.34.240/api/v1/applications/run

POST https://192.168.34.240/api/v1/applications/run

POST https://192.168.34.240/api/v1/applications/run

POST https://192.168.34.240/api/v1/applications/run

```
{
    "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"
```

Status: 200 OK Time: 355 ms Size: 3.3 KB

To add device mappings using Postman:

1. In Postman, POST the following JSON file to trigger the required integration applications in the Integration Service user interface to model SL1 devices to ServiceNow:

NOTE: This example only maps device classes to ServiceNow for VMware, SL1 devices, and a few Cisco devices. If the your environment contains other device classes, you must manually create the mappings.

```
{
  "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 Mappings with the API

You can persistently save device mappings using the API.

1. Use Postman or cURL to do a GET to load the "Sync Devices from SL1 to ServiceNow" integration application:

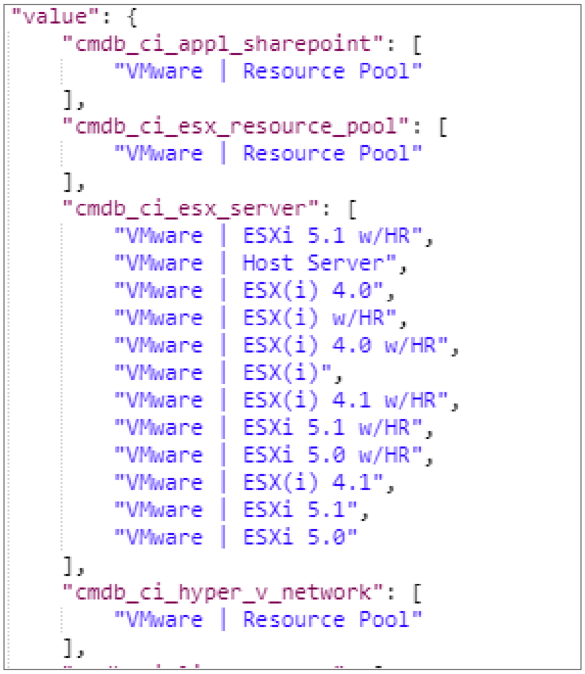
   ```
   GET Integration_Service_hostname/api/v1/applications/device_sync_sciencelogic_to_
   servicenow
   ```

   where *Integration_Service_hostname* is the IP address or URL for your Integration Service system.

NOTE: The response should contain the entire JSON output for the integration 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 integrations:



```
"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 integration application with your new settings. Type the following command at the command line:

```
iscli -uaf device_sync_sciencelogic_to_servicenow -H Integration_Service_hostname
-p password
```

where:

- *Integration_Service_hostename* is the hostname or IP address of the Integration Service system.
- *password* is password you use to log in to the Integration Service system.

Device Attribute Mappings

The "Sync Devices from SL1 to ServiceNow" integration application can also collect manufacturer and model attributes from asset records aligned with devices in SL1 and sync that information with ServiceNow.

The Integration Service only populates the manufacturer and model attributes if the values exist in ServiceNow CIs; the Integration Service does not create new manufacturer values in ServiceNow. "Sync Devices from SL1 to ServiceNow" integration application uses the ***sys_id*** field as a reference when syncing manufacturer and model information between SL1 and ServiceNow.

Default Device Attribute Mappings

The "Sync Devices from SL1 to ServiceNow" integration application contains a set of default device attribute mappings between SL1 and ServiceNow. You can find these mappings in the **attribute_mappings** section of the **Configuration** pane for that integration application.

The following table describes the default device attribute mappings:

SL1 Device Attribute	ServiceNow CI Attribute
Description	`"Description: {{device.device_category}}, Device Class: {{device.device_class}}": [ "short_description"` **NOTE**: This field requires a Jinja2 Template. For more information, see ***Using a Jinja2 Template***.
assetTag	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
ip	ip_address
manufacturer_sys_id	manufacturer
memory	ram
model_sys_id	model_id
name	name
operatingSystem	os
purchaseDate	order_date
serial	serial_number

SL1 Device Attribute	ServiceNow CI Attribute
status	hardware_substatus
warrantyCost	cost
warrantyExpirationDate	warranty_expiration

SL1 Device Attributes Available for Syncing

In the **attribute_mappings** section of the **Configuration** pane for the "Sync Devices from SL1 to ServiceNow" integration application, you can use the following SL1 device attributes from SL1 when syncing attributes with ServiceNow:

- arraySize
- assetTag
- asset_id
- 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
- 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

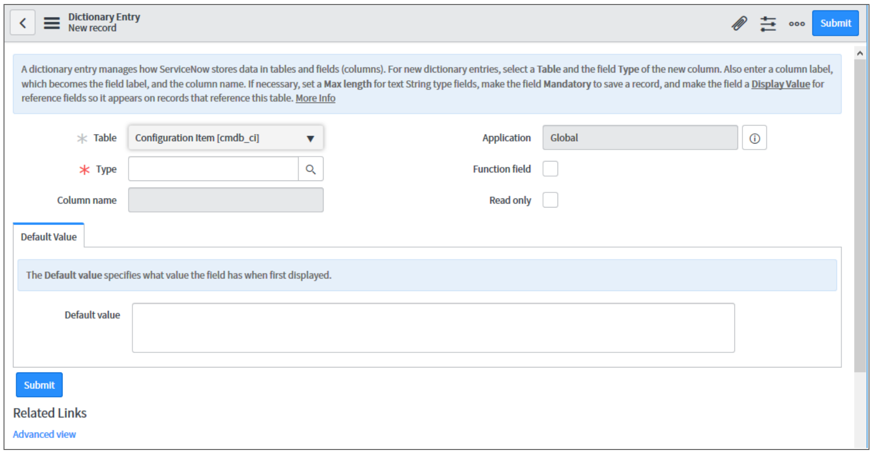
- snow_sys_id
- status
- vitalAssetInformation
- vitalServiceInformation
- warrantyCheck
- warrantyCost
- warrantyDate
- warrantyDescription
- warrantyExpirationDate
- warrantyOrderNumber
- warrantyPolicyNumber
- zone

Adding New Device Attributes to ServiceNow

You can also add one or more new attributes to ServiceNow that you can then sync with SL1.

To add an attribute in ServiceNow:

1. In ServiceNow, search for "Tables" in the filter navigator and select **System Definition > Tables**.
2. From the **Tables** page, search for and select the table to which you want to add a field for a new attribute.
3. From the Table page, click the **[New]** button to add a new field on the table. A new record appears:



Dictionary Entry
New record

A dictionary entry manages how ServiceNow stores data in tables and fields (columns). For new dictionary entries, select a **Table** and the field **Type** of the new column. Also enter a column label, which becomes the field label, and the column name. If necessary, set a **Max length** for text String type fields, make the field **Mandatory** to save a record, and make the field a Display Value for reference fields so it appears on records that reference this table. More Info

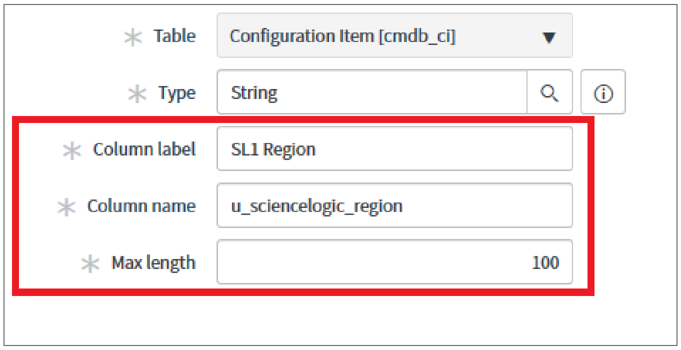
* Table	Configuration Item [cmdb_ci]	Application	Global
* Type		Function field	□
Column name		Read only	□

Default Value

The **Default value** specifies what value the field has when first displayed.

Default value

4. 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:



* Table	Configuration Item [cmdb_ci] ▼
* Type	String
* Column label	SL1 Region
* Column name	u_sciencelogic_region
* Max length	100

> **NOTE**: In the *String* example, above, **Column label** contains the text you want to display in ServiceNow, and **Column name** is the exact column name used by the Integration Service or the API.

5. Complete the required fields and any other fields as needed, and then click the **Submit** button. The field is added to ServiceNow.

Syncing CI Attributes from ServiceNow to SL1

The "Sync CI Attributes from ServiceNow to SL1" integration application imports CI attributes from ServiceNow to the relevant asset and attribute fields in SL1. The CI Sync supports assets, asset configuration, asset maintenance, location, production statuses, and custom attributes.

The "Sync CI Attributes from ServiceNow to SL1" integration 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.

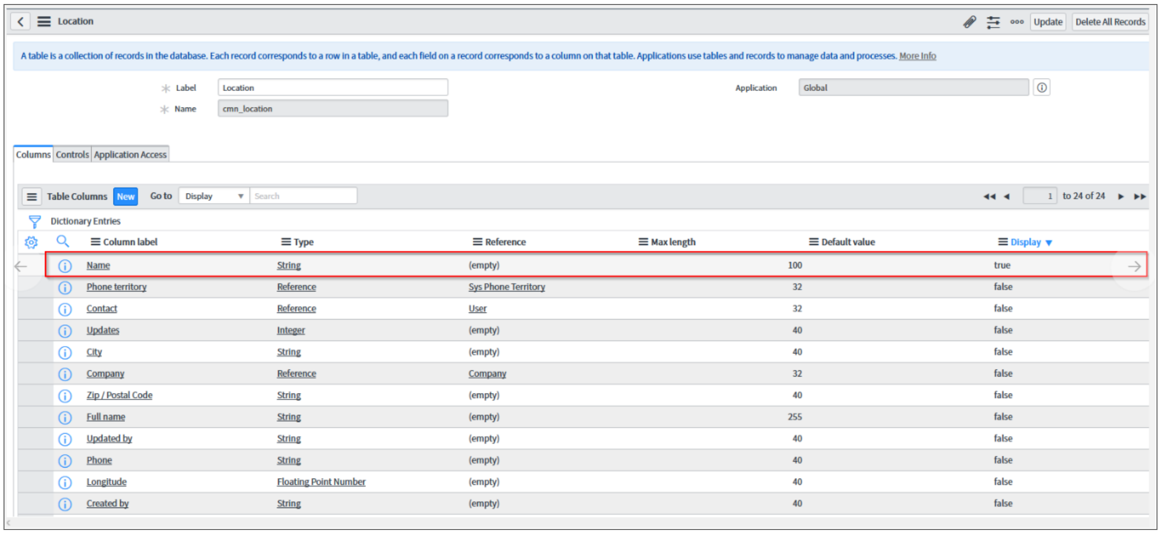
Reference Example:

```
"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 the Integration Service :



A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. More Info

Label: Location

Name: cmn_location

Application: Global

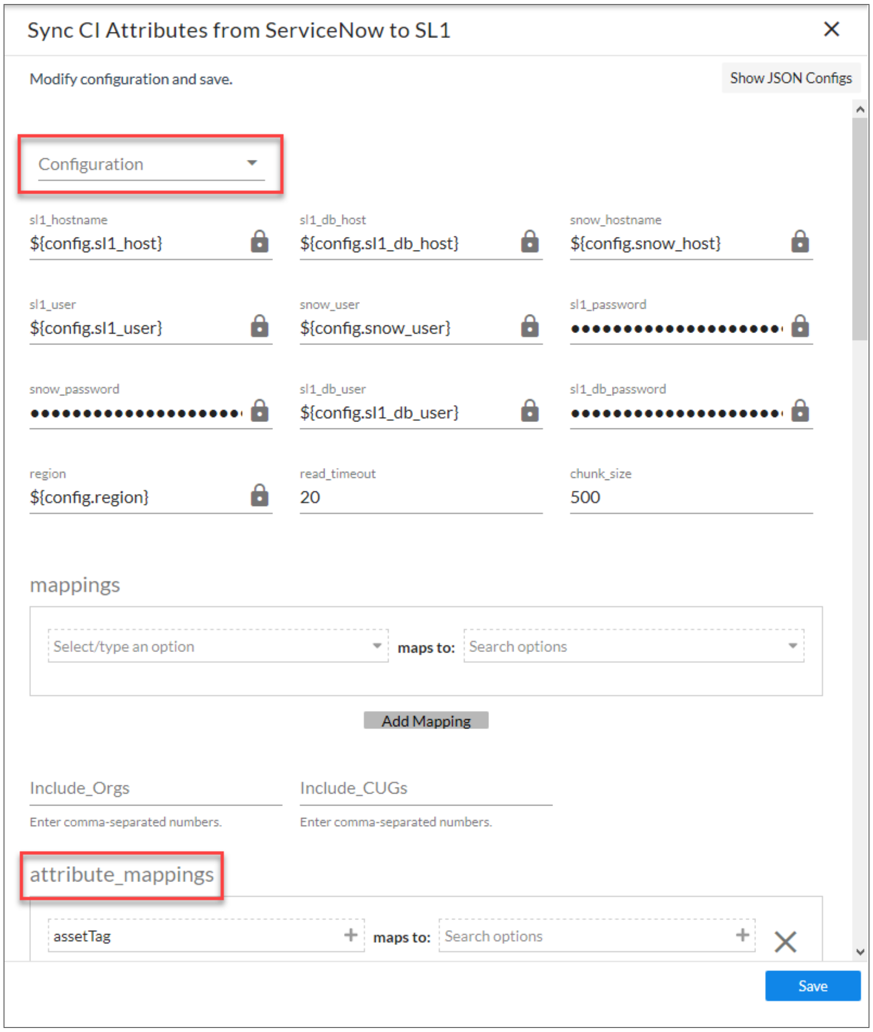
Column label	Type	Reference	Max length	Default value	Display
Name	String	(empty)	100		true
Phone territory	Reference	Sys Phone Territory	32		false
Contact	Reference	User	32		false
Updates	Integer	(empty)	40		false
City	String	(empty)	40		false
Company	Reference	Company	32		false
Zip / Postal Code	String	(empty)	40		false
Full name	String	(empty)	255		false
Updated by	String	(empty)	40		false
Phone	String	(empty)	40		false
Longitude	Floating Point Number	(empty)	40		false
Created by	String	(empty)	40		false

NOTE: When this integration application runs, if no mappings are provided, the Integration Service queries the "Sync Devices from SL1 to ServiceNow" integration application and uses the mappings from that application.

To sync CI attributes from ServiceNow to SL1:

1. Because this integration application uses the mappings and additional attribute options from Device Sync, go to the **Integrations** page of the Integration Service user interface and run the "Sync Devices from SL1 to ServiceNow" integration application.
2. When that application completes, select the "Sync CI Attributes from ServiceNow to SL1" integration application from the **Integrations** page.

3. Click **[Configure]** (⚙). The **Configuration** pane appears:



Sync CI Attributes from ServiceNow to SL1 ✕

Modify configuration and save.

Show JSON Configs

Configuration ▾

sl1_hostname ${config.sl1_host}	sl1_db_host ${config.sl1_db_host}	snow_hostname ${config.snow_host}
sl1_user ${config.sl1_user}	snow_user ${config.snow_user}	sl1_password ••••••••••••••••••••
snow_password ••••••••••••••••••••	sl1_db_user ${config.sl1_db_user}	sl1_db_password ••••••••••••••••••••
region ${config.region}	read_timeout 20	chunk_size 500

mappings

Select/type an option ▾ **maps to:** Search options ▾

Add Mapping

Include_Orgs
Enter comma-separated numbers.

Include_CUGs
Enter comma-separated numbers.

attribute_mappings

assetTag + **maps to:** Search options + ✕

Save

4. From the ***Configuration*** drop-down list, select the configuration object to align with this integration 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 integration application. The region value must match the value in the ***SL1 Region*** field in ServiceNow. If you need to update this value, you will need to define the ***region*** variable in the configuration object that is aligned with this integration application, or align a different configuration object that has the correct ***region*** value.

5. Scroll down to the **attribute_mappings** section, which in previous versions was named the **additional_attributes** section. In this section you can edit an existing attribute, or you can click the **[Add Mapping]** button at the bottom of the section to create a new attribute. Press **[Enter]** after editing the attribute to make sure your changes are saved.

TIP: You can use a Jinja2 Template for attribute fields on the ServiceNow side (the right column). For more information, see *Using a Jinja2 Template*.

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

7. Click **[Run]** (▶) to run the "Sync CI Attributes from ServiceNow to SL1" integration application.

Syncing Advanced Topology Data from SL1 to ServiceNow

The "Sync Advanced Topology from SL1 to ServiceNow" integration application reads Dynamic Component Mapping relationships from SL1 and syncs those relationships with ServiceNow.

If this is a new Integration Service system, you must run both the "Sync Devices from SL1 to ServiceNow" application and the "Sync Interfaces from SL1 to ServiceNow" application at least twice on new Integration Service systems to populate the cache for this integration application.

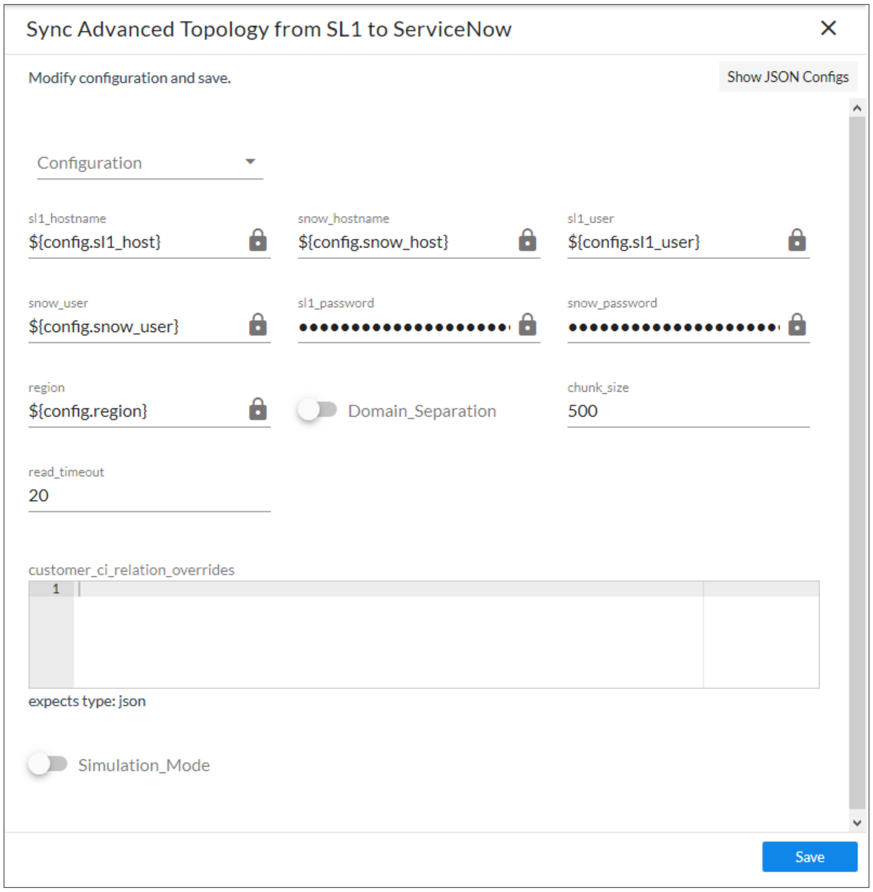
WARNING: The Integration Service only syncs topology data for devices and network interfaces that have already been synced with ServiceNow. Before setting up advanced topology sync, you must first *sync devices* or *sync network interfaces*, depending on your environment.

To sync advanced topology data and relationships from SL1 to ServiceNow:

1. On the **Integrations** page of the Integration Service user interface, click **[Run]** (▶) for the "Sync Devices from SL1 to ServiceNow" integration application. Run the application a second time if this is a new Integration Service system.

2. Click **[Run]** (▶) for the "Sync Interfaces from SL1 to ServiceNow" integration application. Run the application a second time if this is a new Integration Service system.

3. Select the "Sync Advanced Topology from SL1 to ServiceNow" integration application and click **[Configure]** (⚙). The **Configuration** page appears:



Sync Advanced Topology from SL1 to ServiceNow

Modify configuration and save.

Configuration

sl1_hostname: ${config.sl1_host}

snow_hostname: ${config.snow_host}

sl1_user: ${config.sl1_user}

snow_user: ${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

region: ${config.region}

Domain_Separation

chunk_size: 500

read_timeout: 20

customer_ci_relation_overrides

expects type: json

Simulation_Mode

4. Complete the following fields, as needed:

   - ***Configuration***. Select the configuration object with the relevant SL1 and ServiceNow credentials to align with this integration 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 integration application. The region value must match the value in the ***SL1 Region*** field in ServiceNow. If you need to update this value, you will need to define the ***region*** variable in the configuration object that is aligned with this integration application, or align a different configuration object that has the correct ***region*** value.

   - ***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. This application does not support relationships for devices across domains; all devices in a relation payload must be in the same domain.

- ***chunk_size***. Specify the number of topologies to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500.
- ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
- ***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 *Configuring Customer CI Relation Overrides*.
- ***Simulation_Mode***. Select this option if you want to perform a simulated run of this integration application to show you the potential results of that run.

5. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
6. Click **[Run]** (▶) to run the integration application.

Syncing Network Interfaces from SL1 to ServiceNow

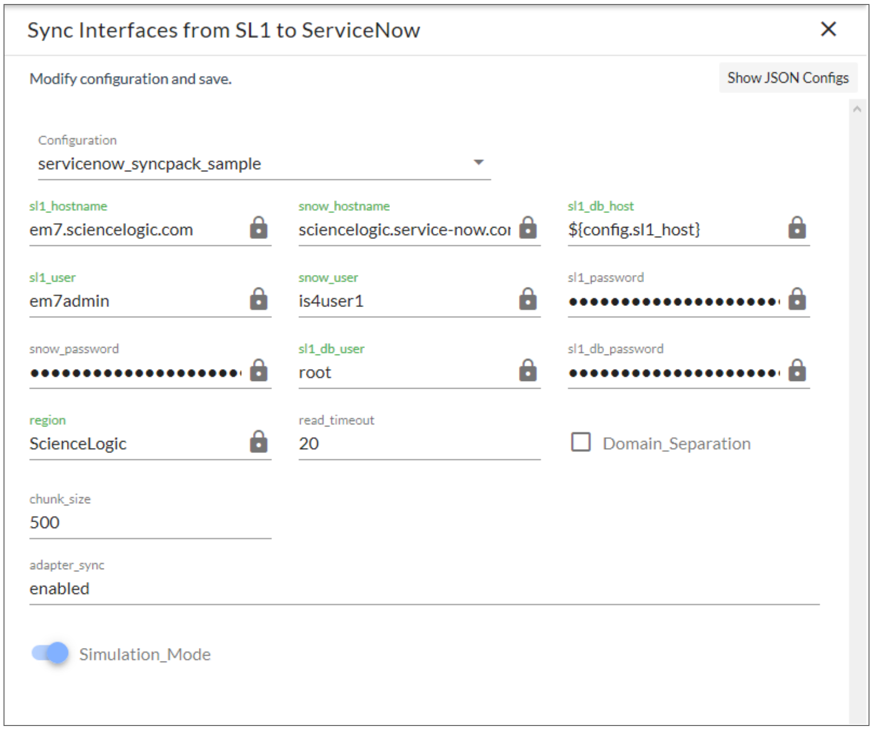
You can map and sync network interfaces in much the same way you sync devices between SL1 and ServiceNow. You run the "Sync Interfaces from SL1 to ServiceNow" integration application, which collects interface data from ServiceNow and SL1 and runs multiple CI syncs for each interface to be synced.

WARNING: The Integration Service 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 SL1 and ServiceNow*.

To sync SL1 network interfaces with ServiceNow:

1. In the Integration Service user interface, go to the **Integrations** page and select the "Sync Interfaces from SL1 to ServiceNow" integration application.

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



Sync Interfaces from SL1 to ServiceNow

Modify configuration and save.

Configuration: servicenow_syncpack_sample

sl1_hostname: em7.sciencelogic.com

snow_hostname: sciencelogic.service-now.coı

sl1_db_host: ${config.sl1_host}

sl1_user: em7admin

snow_user: is4user1

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

sl1_db_user: root

sl1_db_password: ••••••••••••••••••••

region: ScienceLogic

read_timeout: 20

□ Domain_Separation

chunk_size: 500

adapter_sync: enabled

Simulation_Mode

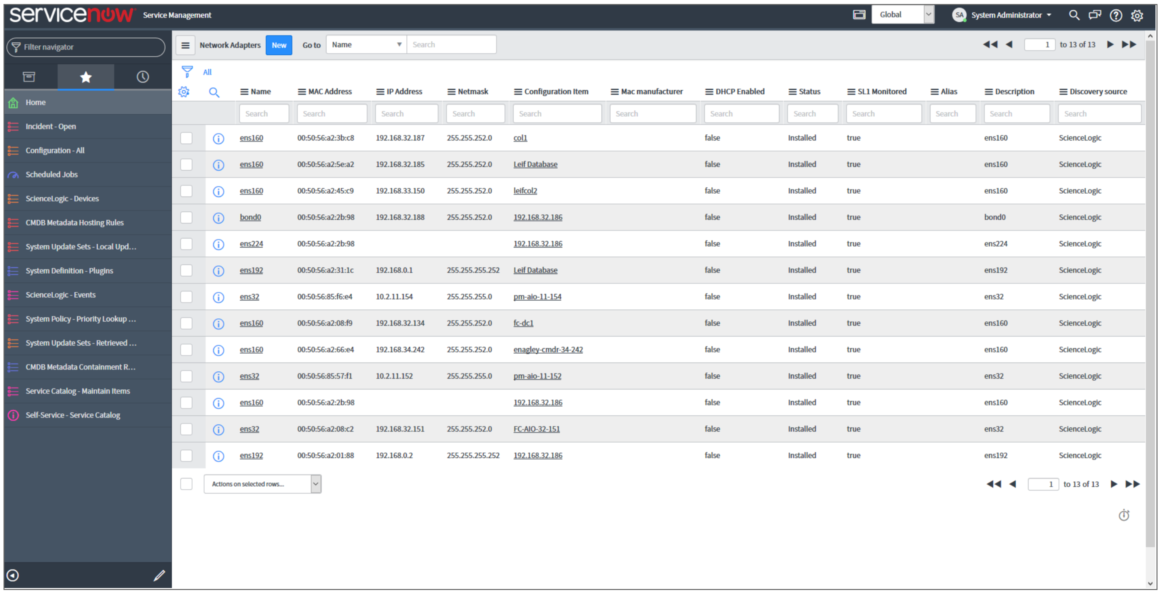
3. Complete the following fields, as needed:

- ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
- ***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 object that is aligned with this integration application, or align a different configuration object that has the correct region value.
- ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
- ***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.
- ***chunk_size***. Specify the number of devices to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500 devices.
- ***adapter_sync***. Select one of the following settings:

- ○ ***off***. Disables interface sync.
- ○ ***all***. Syncs every interface, regardless of its state.
- ○ ***enabled***. Syncs only the interfaces that have a state of "admin up". This is the default setting.

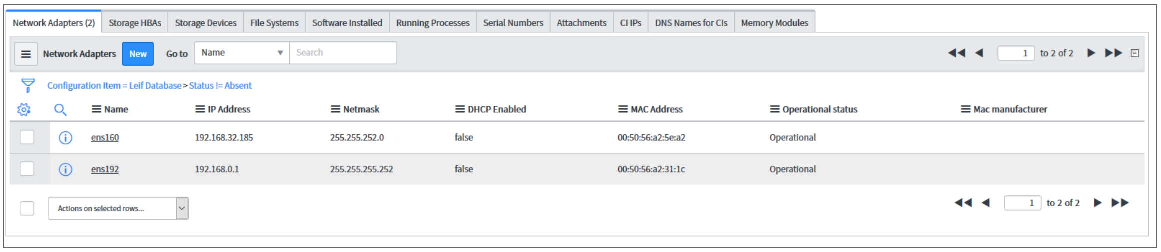
- ***Simulation_Mode***. Select this option if you want to perform a simulated run of this integration application to show you the potential results of that run.

4. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
5. Click **[Run]** (▶) to run the integration application.
6. 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:


servicenow
Service Management
Global
System Administrator
Filter navigator
Home
Incident - Open
Configuration - All
Scheduled Jobs
ScienceLogic - Devices
CMDB Metadata Hosting Rules
System Update Sets - Local Upd...
System Definition - Plugins
ScienceLogic - Events
System Policy - Priority Lookup ...
System Update Sets - Retrieved ...
CMDB Metadata Containment R...
Service Catalog - Maintain Items
Self-Service - Service Catalog
Network Adapters
New
Go to
Name
Search
All
1
to 13 of 13
Actions on selected rows...

Name	MAC Address	IP Address	Netmask	Configuration Item	Mac manufacturer	DHCP Enabled	Status	SL1 Monitored	Alias	Description	Discovery source
ens160	00:50:56:a2:3b:c8	192.168.32.187	255.255.252.0	col1		false	Installed	true		ens160	ScienceLogic
ens160	00:50:56:a2:5e:a2	192.168.32.185	255.255.252.0	Leif Database		false	Installed	true		ens160	ScienceLogic
ens160	00:50:56:a2:45:c9	192.168.33.150	255.255.252.0	leifcol2		false	Installed	true		ens160	ScienceLogic
bond0	00:50:56:a2:2b:98	192.168.32.188	255.255.252.0	192.168.32.186		false	Installed	true		bond0	ScienceLogic
ens224	00:50:56:a2:2b:98			192.168.32.186		false	Installed	true		ens224	ScienceLogic
ens192	00:50:56:a2:31:1c	192.168.0.1	255.255.255.252	Leif Database		false	Installed	true		ens192	ScienceLogic
ens32	00:50:56:85:f6:e4	10.2.11.154	255.255.255.0	pm-aio-11-154		false	Installed	true		ens32	ScienceLogic
ens160	00:50:56:a2:08:f9	192.168.32.134	255.255.252.0	fc-dc1		false	Installed	true		ens160	ScienceLogic
ens160	00:50:56:a2:66:e4	192.168.34.242	255.255.252.0	enagley-cmdr-34-242		false	Installed	true		ens160	ScienceLogic
ens32	00:50:56:85:57:f1	10.2.11.152	255.255.255.0	pm-aio-11-152		false	Installed	true		ens32	ScienceLogic
ens160	00:50:56:a2:2b:98			192.168.32.186		false	Installed	true		ens160	ScienceLogic
ens32	00:50:56:a2:08:c2	192.168.32.151	255.255.252.0	FC-AIO-32-151		false	Installed	true		ens32	ScienceLogic
ens192	00:50:56:a2:01:88	192.168.0.2	255.255.255.252	192.168.32.186		false	Installed	true		ens192	ScienceLogic

6. Select a network interface from the list and scroll down to the **Network Adapters** tab to see more information about the interface, such as the **Operational status** value, which is synced from SL1.



Configuration Item = Leif Database > Status != Absent

Name	IP Address	Netmask	DHCP Enabled	MAC Address	Operational status	Mac manufacturer
ens160	192.168.32.185	255.255.252.0	false	00:50:56:a2:5e:a2	Operational	
ens192	192.168.0.1	255.255.255.252	false	00:50:56:a2:31:1c	Operational	

> NOTE: The **Operational status** value is different from the **SL1 Monitored** value, but the Integration Service tracks both values.

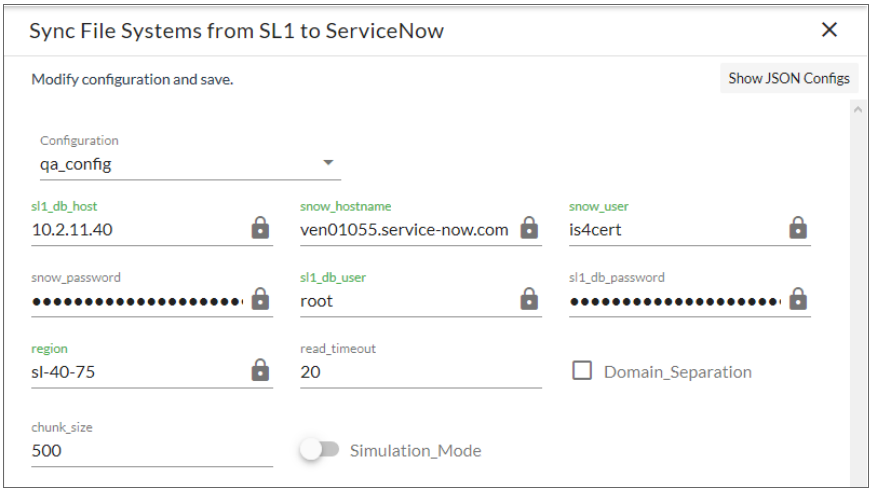
Syncing File Systems from SL1 to ServiceNow

You can map and sync file systems in much the same way you sync devices between SL1 and ServiceNow. The "Sync File Systems from SL1 to ServiceNow" integration application reads file systems discovered in SL1 and then maps them to a parent CI record in ServiceNow.

WARNING: The Integration Service 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 SL1 and ServiceNow***.

To sync SL1 file systems with ServiceNow:

1. In the Integration Service user interface, go to the **Integrations** page and select the "Sync File Systems from SL1 to ServiceNow" integration application.
2. Click the **[Configure]** button to open the **Configuration** pane:



Sync File Systems from SL1 to ServiceNow

Modify configuration and save.

Configuration: qa_config

sl1_db_host: 10.2.11.40

snow_hostname: ven01055.service-now.com

snow_user: is4cert

snow_password: ••••••••••••••••••••

sl1_db_user: root

sl1_db_password: ••••••••••••••••••••

region: sl-40-75

read_timeout: 20

☐ Domain_Separation

chunk_size: 500

○ Simulation_Mode

3. Complete the following fields, as needed:

   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***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 integration application, or align a different configuration that has the correct region value.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.

- ***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.
- ***chunk_size***. Specify the number of file systems to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500.
- ***Simulation_Mode***. Select this option if you want to perform a simulated run of this integration application to show you the potential results of that run.

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

5. Click **[Run]** (▶) to run the integration application.

Syncing Business Services from SL1 to ServiceNow

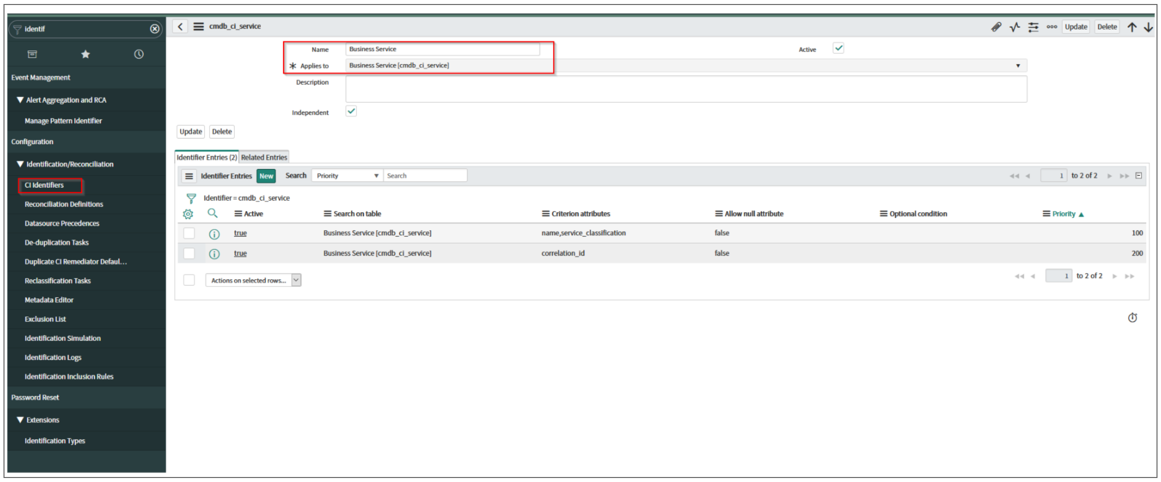
The ***Sync Business Services from SL1 to ServiceNow*** integration application reads Business Services, IT Services, and Device Services from SL1 and syncs them with business services in ServiceNow. This integration application creates and updates services, but it does not delete services.

> **NOTE:** Applications and Application Components from SL1 are *not* synced by the Integration Service from SL1 to ServiceNow.

> **WARNING:** The Integration Service 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 SL1 and ServiceNow***.

To sync SL1 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:



Name: Business Service

Applies to: Business Service [cmdb_ci_service]

Active: ✓

Description:

Independent: ✓

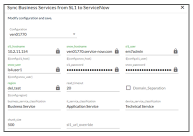
Identifier = cmdb_ci_service

Active	Search on table	Criterion attributes	Allow null attribute	Optional condition	Priority ▲
true	Business Service [cmdb_ci_service]	name,service_classification	false		100
true	Business Service [cmdb_ci_service]	correlation_id	false		200

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. In the Integration Service user interface, go to the **Integrations** page and select the "Sync Business Services from SL1 to ServiceNow" integration application.
8. Click **[Configure]** (⚙). The **Configuration** pane appears:



Modify configuration and save.

Configuration: ven01770

sl1_hostname: 10.2.11.154 — ${config.sl1_host}

snow_hostname: ven01770.service-now.com — ${config.snow_host}

sl1_user: em7admin — ${config.sl1_user}

snow_user: is4user1 — ${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

region: del_test — ${config.region}

read_timeout: 20

□ Domain_Separation

business_service_classification: Business Service

it_service_classification: Application Service

device_service_classificaiton: Technical Service

chunk_size: 500

sl1_url_override

9. Complete the following fields, as needed:

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

- ***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 integration application, or align a different configuration that has the correct region value.
- ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
- ***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.
- ***business_service_classification***, ***it_service_classification***, and ***device_service_classification***. Use these fields to update the default service classifications. Optional.
- ***chunk_size***. Specify the number of services to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500.
- ***sl1_url_override***. Specify a URL that is different from the standard SL1 URL that gets sent to the ServiceNow CI record. Optional.

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 integration application.

Syncing Installed Software between SL1 and ServiceNow

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

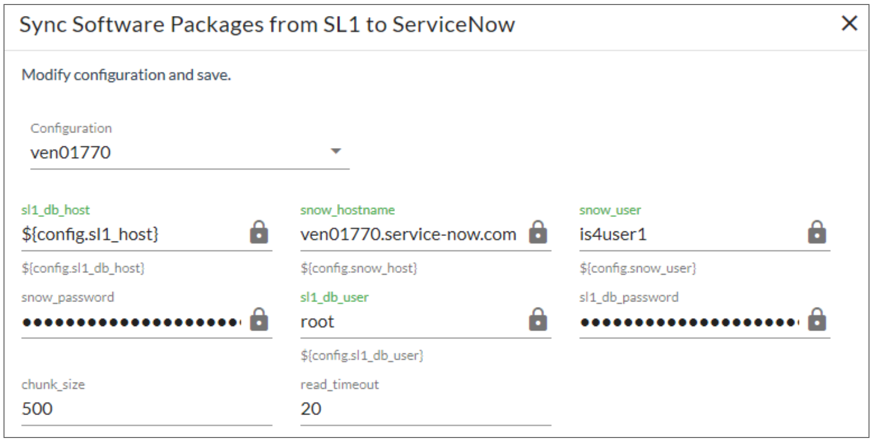
- "Sync Software Packages from SL1 to ServiceNow". Reads all software packages from SL1 and creates new CIs in ServiceNow. Run this integration before running the "Sync Installed Software" integration application.
- "Sync Installed Software from SL1 to ServiceNow". Reads all available software packages from ServiceNow and the devices aligned to that software by region and syncs them with SL1.

The integration applications do 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 SL1. 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 SL1 and ServiceNow:

1. Make sure that you have recently run the "Sync Devices from SL1 to ServiceNow" integration application to populate the device cache.
2. In the Integration Service user interface, go to the **Integrations** page and select the "Sync Software Packages from SL1 to ServiceNow" integration application.
3. Click **[Configure]** (⚙). The **Configuration** pane appears:



Sync Software Packages from SL1 to ServiceNow

Modify configuration and save.

Configuration
ven01770

sl1_db_host
${config.sl1_host}
${config.sl1_db_host}

snow_hostname
ven01770.service-now.com
${config.snow_host}

snow_user
is4user1
${config.snow_user}

snow_password
••••••••••••••••••••

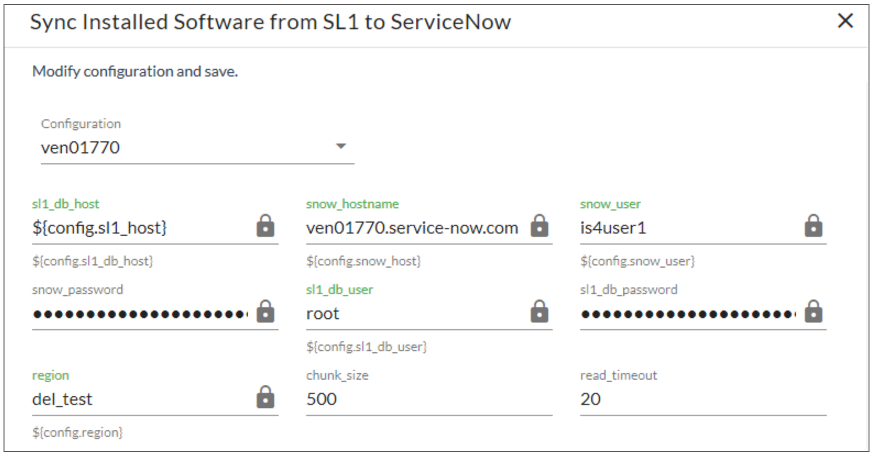
sl1_db_user
root
${config.sl1_db_user}

sl1_db_password
••••••••••••••••••••

chunk_size
500

read_timeout
20

4. Complete the following fields, as needed:
   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***chunk_size***. Specify the number of services to include in each chunk sent to ServiceNow when you run this integration application. The default chunk size is 500.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
5. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
6. Click **[Run]** (▶) to run the integration application.
7. After the "Sync Software Packages from SL1 to ServiceNow" integration application finishes running, go to the **Integrations** page and select the "Sync Installed Software from SL1 to ServiceNow" integration application.
8. Click **[Configure]** (⚙). The **Configuration** pane appears:



Sync Installed Software from SL1 to ServiceNow

Modify configuration and save.

Configuration
ven01770

sl1_db_host
${config.sl1_host}
${config.sl1_db_host}

snow_hostname
ven01770.service-now.com
${config.snow_host}

snow_user
is4user1
${config.snow_user}

snow_password
••••••••••••••••••••

sl1_db_user
root
${config.sl1_db_user}

sl1_db_password
••••••••••••••••••••

region
del_test
${config.region}

chunk_size
500

read_timeout
20

9. Complete the following fields, as needed:

    - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
    - ***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 integration 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 integration application. The default chunk size is 500.
    - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.

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 integration application.

Discovery Sync

The Discovery Sync integration lets you use SL1 for discovering and syncing ServiceNow devices. With Discovery Sync, you start an SL1 discovery session from ServiceNow and then sync the newly discovered SL1 devices or virtual devices and their data with ServiceNow.

Before running a Discovery Sync session, you must complete the following steps first:

1. For domain-separated ServiceNow instances, perform a company sync by running the "Sync Organizations from SL1 to ServiceNow" integration application in the Integration Service user interface. For more information, see *Syncing Organizations from SL1 to ServiceNow*.
2. In ServiceNow, configure a service request for Discovery Sync. For more information, see *Configuring a ServiceNow Service Request for Discovery Sync*.
3. In the Integration Service user interface, run the integration applications listed in the *Discovery Sync Workflow*.

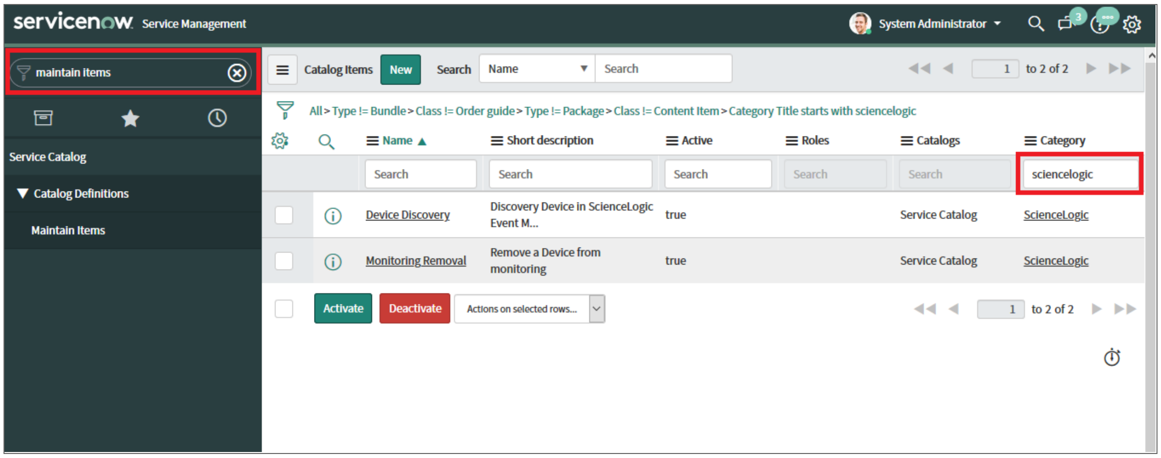
Configuring a ServiceNow Service Request for Discovery Sync

Before you can run a Discovery Sync, you need to configure the catalog and category values in the ServiceNow service request forms. You also need to activate the "Device Discovery" service request in ServiceNow.

NOTE: Because some of the fields in the service request form will only populate if you have completed the previous fields in the form, you need to complete the fields in the service request form in sequential order.

To configure the ServiceNow service requests for Discovery Sync:

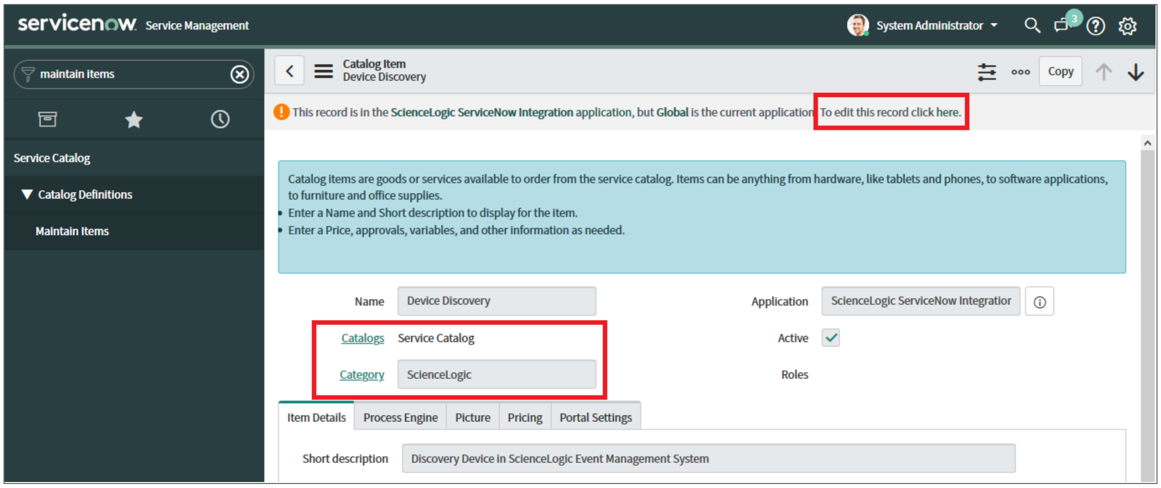
1. In ServiceNow, search for "Maintain Items" in the filter navigator.
2. Go to **Service Catalog > Catalog Definitions > Maintain Items** and type "ScienceLogic" in the Category field. The **Device Discovery** and **Monitoring Removal** service requests appear:



All > Type != Bundle > Class != Order guide > Type != Package > Class != Content Item > Category Title starts with sciencelogic

Name ▲	Short description	Active	Roles	Catalogs	Category
Search	Search	Search	Search	Search	sciencelogic
Device Discovery	Discovery Device in ScienceLogic Event M...	true		Service Catalog	ScienceLogic
Monitoring Removal	Remove a Device from monitoring	true		Service Catalog	ScienceLogic

3. Open the **Device Discovery** service request and ensure that the **Catalogs** and **Category** fields are accurate. For example:



This record is in the ScienceLogic ServiceNow Integration application, but Global is the current application To edit this record click here.

Catalog items are goods or services available to order from the service catalog. Items can be anything from hardware, like tablets and phones, to software applications, to furniture and office supplies.

- Enter a Name and Short description to display for the item.
- Enter a Price, approvals, variables, and other information as needed.

Name	Device Discovery	Application	ScienceLogic ServiceNow Integratior
Catalogs	Service Catalog	Active	✓
Category	ScienceLogic	Roles	

Short description: Discovery Device in ScienceLogic Event Management System

> NOTE: Do not set the ***Category*** to a *Change Request*.

4. If you need to update these fields, click the "To edit this record click ***here***" link at the top of the detail page.
5. Update the fields and click the **[Update]** button to save your changes.
6. From the **Catalog Items** page, click the check box for the **Device Discovery** service request and click **[Activate]**.

> NOTE: This service request is instance-specific, which means that the service request will appear in the same location as the catalogs you specified for that request. In the example, above, the **Catalog** was set to *Service Catalog*.

7. Navigate to the relevant catalog for the service request. For example, if you selected *Service Catalog* for one or both requests, then type "Service Catalog" in the filter navigator, or select **Self-Service > Service Catalog** to view the new service requests. Type "device discovery" in the ***Search catalog*** field to quickly locate the request.
8. Run the integration applications listed in the *Discovery Sync Workflow* before creating the Device Discovery service request in ServiceNow.

Discovery Sync Workflow

To prepare SL1 and ServiceNow for a Discovery Sync, run the following integration applications in the Integration Service user interface, in the following order:

1. ***Sync Discovery Requirements***. This application exports information from SL1 to populate the information in the ServiceNow request form. You must run this application before you can create the discovery sync session

in ServiceNow. This application uses one or more of the following options from the **Configuration** pane:

- ***Configuration***. Select the relevant configuration object to align with this integration 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 integration application. The region value must match the value in the ***SL1 Region*** field in ServiceNow. If you need to update this value, you will need to define the ***region*** variable in the configuration object that is aligned with this integration application, or align a different configuration object that has the correct ***region*** value.

- ***read_timeout***. Specify the maximum amount of time in seconds the integration 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 integration application. The default is 500.
- ***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.
- ***Update_Name***. This option addresses the situation where the Integration Service finds a match with a device or CI, but the names do not match. This option updates a device or CI name based on your selection in the *Source_of_Truth* field, below. For example, if you selected *ScienceLogic* as the source of truth, the Integration Service uses the device name from ScienceLogic as the updated name.
- ***Source_of_Truth***. Select whether you want to use data from ServiceNow or ScienceLogic as the "source of truth" when this integration application encounters duplicate data or data collisions.
- ***Create_Missing***. Select this option if you want the Integration Service to create a new device or CI if that record is missing, based on your selection in the *Source_of_Truth* field.
- ***Sync_Empty_Groups***. Select this option if you want to sync device groups that have no devices, or device groups that have devices but no matching CIs.

2. ***Sync Service Requests from ServiceNow to SL1***. This application sends the request forms to SL1. This application was called "Sync Discovery Session Requests from ServiceNow to SL1" in previous versions of the Synchronization PowerPack. This application uses one or more of the following options from the **Configuration** pane:

   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***Open_State***. The State ID from ServiceNow that specifies which Requested Items (RITMs) to pull and process. The default is 1.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.

- ***Closed_Success_State***. The State ID for a successfully created virtual device. The State ID for a successful run changes from 1 to 2 and then ends with 4. The default is 3.
- ***Closed_Failed_State***. The State ID for failed discoveries or failed virtual device creation, usually caused by invalid payloads. The State ID for a failed run changes from 1 to 2 and then ends with 4. The default is 4.
- ***In_Progress_State***. The State ID for RITMs for a running discovery. The default is 2.
- ***recursively_disable_children***. Leave this field blank.
- ***target_vcug***. Leave this field blank.

3. ***Sync Discovery Session Status from SL1 to ServiceNow***. This application populates the discovery session logs back to ServiceNow. This application uses the following options from the **Configuration** pane:

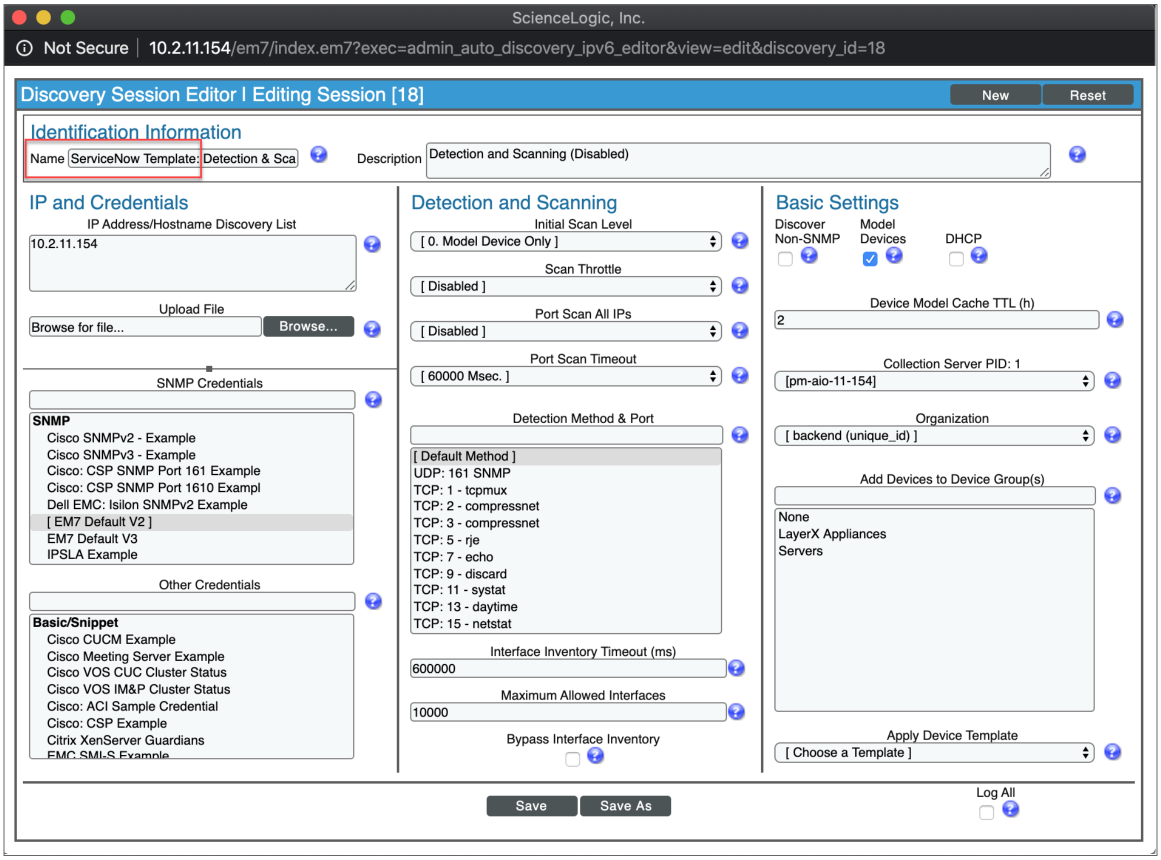
   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
   - ***Closed_Success_State***. The State ID for a successfully created discovery. The State ID for a successful run changes from 1 to 2 and then ends with 4. The default is 3.
   - ***sys_id_target***. Takes the ***sys_id*** value from the CI in the ServiceNow Service Request and populates it in the relevant field in SL1, such as ***c-sys_id***.
   - ***ci_class_target***. Takes the ***ci_class*** value from the CI in the ServiceNow Service Request and populates it in the relevant field in SL1, such as ***c-ci_class***.

   > **NOTE:** If the ***sys_id_target*** field and the ***ci_class_target*** field are not populated, the Integration Service will skip the process of consuming cached data and populating custom attribute fields in SL1 with the ***sys_id*** and ***ci_class*** values of newly discovered devices.

4. ***Sync Discovery Templates from SL1 to ServiceNow***. This application creates Service Catalog templates in ServiceNow based on Discovery Sessions that were created in SL1. This option lets you use any existing SL1 Discovery Sessions as a template for discovering or monitoring a CI with SL1. This application uses the following options from the **Configuration** pane:

   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***chunk_size***. Specify the number of devices to include in each chunk sent to ServiceNow when you run this integration application. The default is 500.
   - ***template_prefix***. Specify the prefix string that the Integration Service will search for in SL1. Any Discovery Sessions that contain that string will be used in ServiceNow to create a service catalog template. The default string is ***ServiceNow Template:***, but you can configure this as needed.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.

In SL1, go to the **Discovery Control Panel** page (Manage > Classic Discovery) and search for the Discovery Session or Sessions that you want to use as a template. The start of the name in the ***Name*** field should match the value in the ***template_prefix*** field, above:



10.2.11.154/em7/index.em7?exec=admin_auto_discovery_ipv6_editor&view=edit&discovery_id=18

Discovery Session Editor | Editing Session [18]

Identification Information

Name: ServiceNow Template: Detection & Sca

Description: Detection and Scanning (Disabled)

IP and Credentials

IP Address/Hostname Discovery List: 10.2.11.154

Upload File: Browse for file...

SNMP Credentials:
SNMP
Cisco SNMPv2 - Example
Cisco SNMPv3 - Example
Cisco: CSP SNMP Port 161 Example
Cisco: CSP SNMP Port 1610 Exampl
Dell EMC: Isilon SNMPv2 Example
[EM7 Default V2]
EM7 Default V3
IPSLA Example

Other Credentials:
Basic/Snippet
Cisco CUCM Example
Cisco Meeting Server Example
Cisco VOS CUC Cluster Status
Cisco VOS IM&P Cluster Status
Cisco: ACI Sample Credential
Cisco: CSP Example
Citrix XenServer Guardians
EMC SMI-S Example

Detection and Scanning

Initial Scan Level: [0. Model Device Only]

Scan Throttle: [Disabled]

Port Scan All IPs: [Disabled]

Port Scan Timeout: [60000 Msec.]

Detection Method & Port:
[Default Method]
UDP: 161 SNMP
TCP: 1 - tcpmux
TCP: 2 - compressnet
TCP: 3 - compressnet
TCP: 5 - rje
TCP: 7 - echo
TCP: 9 - discard
TCP: 11 - systat
TCP: 13 - daytime
TCP: 15 - netstat

Interface Inventory Timeout (ms): 600000

Maximum Allowed Interfaces: 10000

Bypass Interface Inventory: □

Basic Settings

Discover Non-SNMP: □
Model Devices: ☑
DHCP: □

Device Model Cache TTL (h): 2

Collection Server PID: 1: [pm-aio-11-154]

Organization: [backend (unique_id)]

Add Devices to Device Group(s):
None
LayerX Appliances
Servers

Apply Device Template: [Choose a Template]

Log All: □

5. ***Sync Devices from SL1 to ServiceNow***. Running this application ensures that the devices discovered by SL1 get synced to ServiceNow.

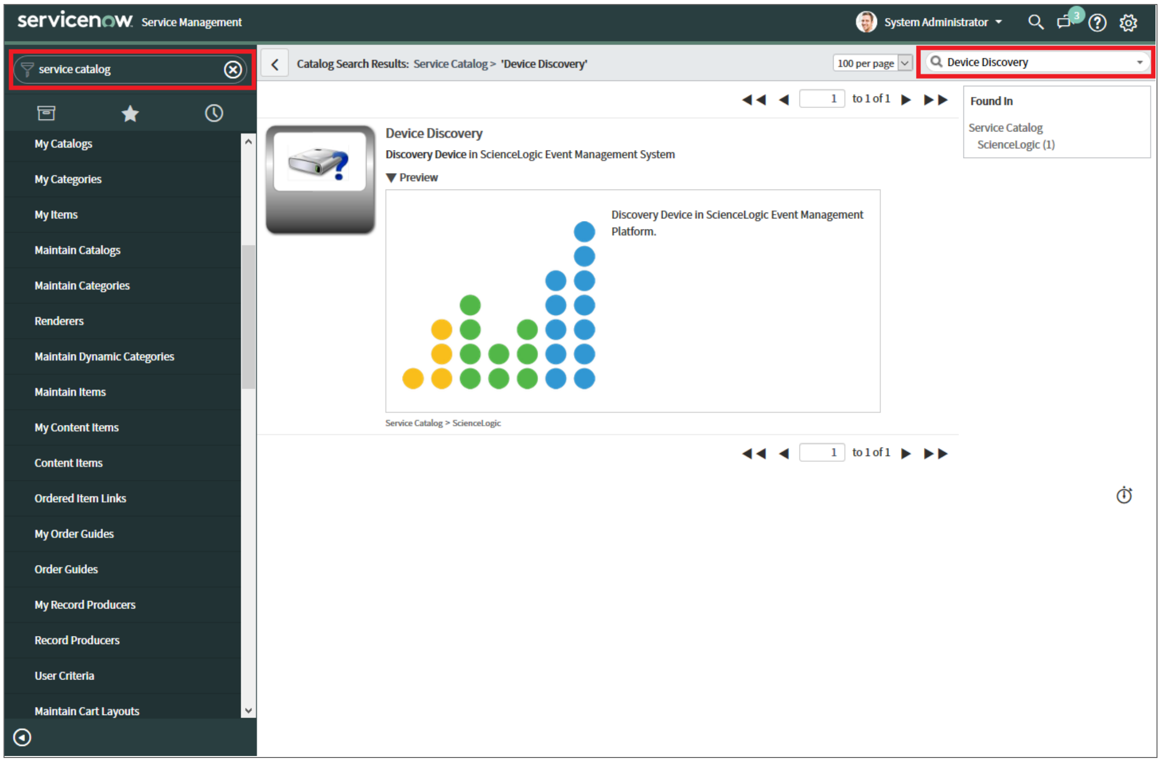
6. When the integration applications finish running, the Integration Service sends the status of those applications to ServiceNow, and you can ***run a Discovery Sync in ServiceNow***.

Running a Discovery Sync in ServiceNow

The Discovery Sync process starts an SL1 discovery session from ServiceNow and syncs the newly discovered SL1 devices and their data with ServiceNow. You can choose to discover standard devices or virtual devices.

To run a Discovery Sync from the Service Catalog page:

1. In ServiceNow, search for "service catalog" in the filter navigator.
2. Navigate to the **Service Catalog** page (Self-Service > Service Catalog), type "device discovery" in the ***Search catalog*** field at the top right, and press **[Enter]**. The **Device Discovery** catalog entry appears:



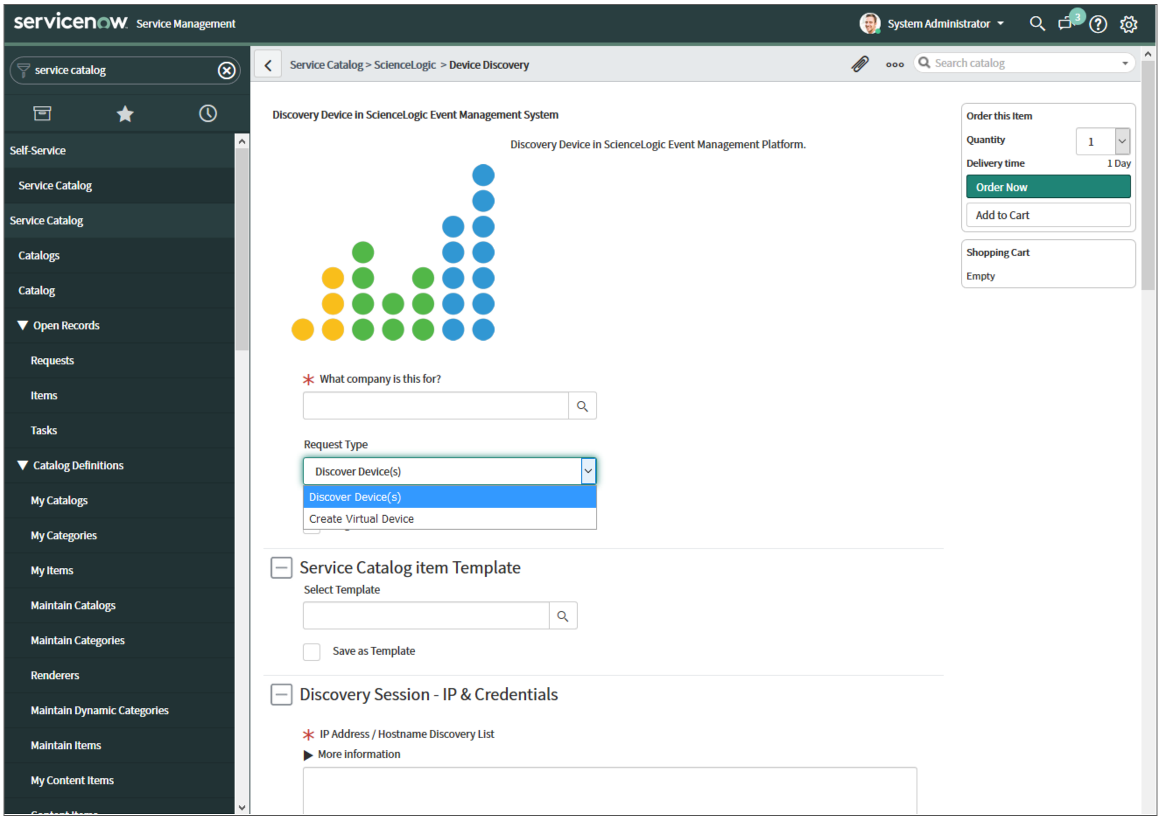
Device Discovery

Discovery Device in ScienceLogic Event Management System

Discovery Device in ScienceLogic Event Management Platform.

NOTE: Previous versions of the "ScienceLogic SL1: CMDB & Incident Automation Application" (also called the Certified or Scoped Application) created two separate service requests: **Create Virtual Device** and **Device Discovery**. Both features have been combined into the **Device Discovery** service request.

3. Click **Device Discovery**. The **Device Discovery** service request appears:



Service Catalog > ScienceLogic > Device Discovery

Discovery Device in ScienceLogic Event Management System

Discovery Device in ScienceLogic Event Management Platform.

Order this Item

Quantity 1

Delivery time 1 Day

Shopping Cart

Empty

* What company is this for?

Request Type

Discover Device(s)

Discover Device(s)

Create Virtual Device

Service Catalog item Template

Select Template

□ Save as Template

Discovery Session - IP & Credentials

* IP Address / Hostname Discovery List

▶ More information

4. In the ***What company is this for?*** field, specify the company. The ***Region*** field updates automatically based on the company you select.

5. In the ***Request Type*** field, select *Discover Device(s)* or *Create Virtual Device*, depending on the type of device you want to discover.

   - If you selected *Discover Device(s)*, go to step 6.
   - If you selected *Create Virtual Device*, go to step 7.

6. If you selected *Discover Device(s)* in the ***Request Type*** field, complete the following fields:

   - ***Log All***. Select this option if you want the discovery session to use verbose logging. When you select this option, SL1 logs details about each IP address or hostname specified in the ***IP Address/Hostname Discovery List*** field, even if the results are "No device found at this address."
   - ***Select Template***. To use a template that contains your device discovery information, select the template from the dropdown.

TIP: You can save the current device discovery as a template by checking ***Save as Template***. A template saves all of the discovery settings except for the IP addresses. You can access existing templates on the **Catalog Template** page in ServiceNow (ScienceLogic > Automations > Catalog Templates).

   - ***IP Address/Hostname Discovery List***. Provide a list of IP addresses, hostnames, or fully-qualified domain names for SL1 to scan during discovery:

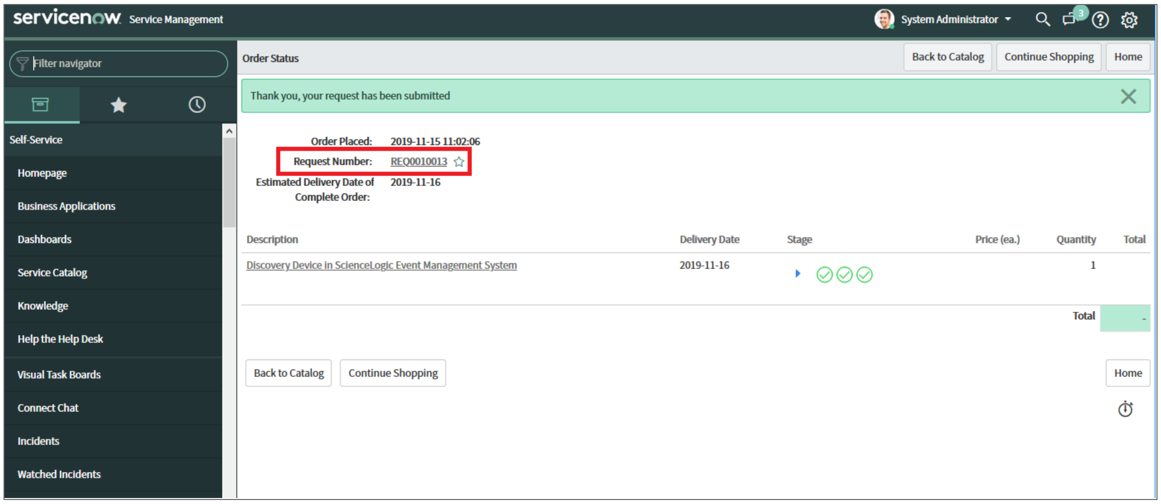
    - One or more *single IPv4 addresses* separated by commas and a new line. Each IP address must be in standard IP notation and cannot exceed 15 characters. For example, "10.20.30.1, 10.20.30.2, 10.20."
    - One or more *ranges of IPv4 addresses* with "-" (dash) characters between the beginning of the range and the end of the range. Separate each range with a comma. For example, "10.20.30.1 – 10.20.30.254".
    - One or more IP address ranges in *IPv4 CIDR notation*. Separate each item in the list with a comma. For example, "192.168.168.0/24".
    - One or more hostnames (fully-qualified domain names). Separate each item in the list with a comma.

- ***Credentials***. Select one or more SNMP credentials to allow SL1 to access a device's SNMP data.
- ***Discover Non-SNMP***. Specifies whether or not SL1 should discover devices that don't respond to SNMP requests.
- ***Model Devices***. Determines whether or not the devices that are discovered with this discovery session can be managed through SL1.
- ***DHCP***. Specifies whether or not the specified range of IPs and hostnames use DHCP. If you select this option, SL1 performs a DNS lookup for the device during discovery and each time SL1 retrieves information from the device.
- ***Device Model Cache TTL (h)***. Amount of time, in hours, that SL1 stores information about devices that are discovered but not modeled, either because the ***Model Devices*** option is not enabled or because SL1 cannot determine whether a duplicate device already exists. The cached data can be used to manually model the device from the **Discovery Session** window.
- ***Collection Server***. Select an existing collector to monitor the discovered devices. Required.
- ***What company is this for?***. Specify the company that will use this discovery data. Click the magnifying glass icon to locate a company.
- ***Add Devices to Device Groups***. Select one or more existing device groups to which you want to add the discovered devices.
- ***Apply Device Template*** . Select an existing device template if needed. As SL1 discovers a device in the IP discovery list, that device is configured with the selected device template.
- ***Initial Scan Level***. For this discovery session only, specifies the data to be gathered during the initial discovery session.
- ***Scan Throttle***. Specifies the amount of time a discovery process should pause between each specified IP address (specified in the ***IP Address/Hostname Discovery List*** field). Pausing discovery processes between IP addresses spreads the amount of network traffic generated by discovery over a longer period of time.
- ***Scan Default Ports***. Select this option to scan the default ports: 21,22,23,25,80. If you de-select this option, you can specify a different list of ports in the ***Custom Port Scan*** field that appears.
- ***Port Scan All IPs***. For the initial discovery session only, specifies whether SL1 should scan all IP addresses on a device for open ports.

- ***Port Scan Timeout***. For the initial discovery session only, specifies the length of time, in milliseconds, after which SL1 should stop trying to scan an IP address for open ports and begin scanning the next IP address (if applicable).
- ***Interface Inventory Timeout (ms)***. Specifies the maximum amount of time that the discovery processes will spend polling a device for the list of interfaces. After the specified time, SL1 will stop polling the device, will not model the device, and will continue with discovery. The default value is 600,000 ms (10 minutes).
- ***Maximum Allowed Interfaces***. Specifies the maximum number of interfaces per devices. If a device exceeds this number of interfaces, SL1 stops scanning the device, will not model the device, and will continue with discovery. The default value is 10,000.
- ***Bypass Interface Inventory***. Select this option if you do *not* want SL1 to attempt to discover interfaces for each device in the discovery session.

7. If you selected *Create Virtual Device* in the ***Request Type*** field, complete the following fields:
    - ***Name***. Type a name for the virtual device.
    - ***Virtual Device Class***. Specify the device class of the virtual device. Click the magnifying glass icon to locate any classes aligned with your organization.
    - ***Collector Group***. Specify the SL1 collector group to use for the Discovery Sync. Click the magnifying glass icon to locate any collector groups aligned with your organization.

8. Click **[Order Now]**. On the **Order Status** page that appears, make a note of value in the ***Request Number*** field:



Thank you, your request has been submitted

Order Placed: 2019-11-15 11:02:06

Request Number: REQ0010013

Estimated Delivery Date of Complete Order: 2019-11-16

Description	Delivery Date	Stage	Price (ea.)	Quantity	Total
Discovery Device in ScienceLogic Event Management System	2019-11-16	✓✓✓		1	
				Total	-

9. In the Integration Service user interface, go to the **Integrations** page and run the "Sync Service Requests from ServiceNow to SL1" integration application.
10. When the application completes, go to **Self-Service > My Requests** in ServiceNow.
11. Click the **RITM** record link to go to the **Requested Item** page. The **State** field should update to *Closed Complete* and the request should be assigned to itself.
12. In the Integration Service user interface, go to the **Integrations** page and run the "Sync Devices from SL1 to

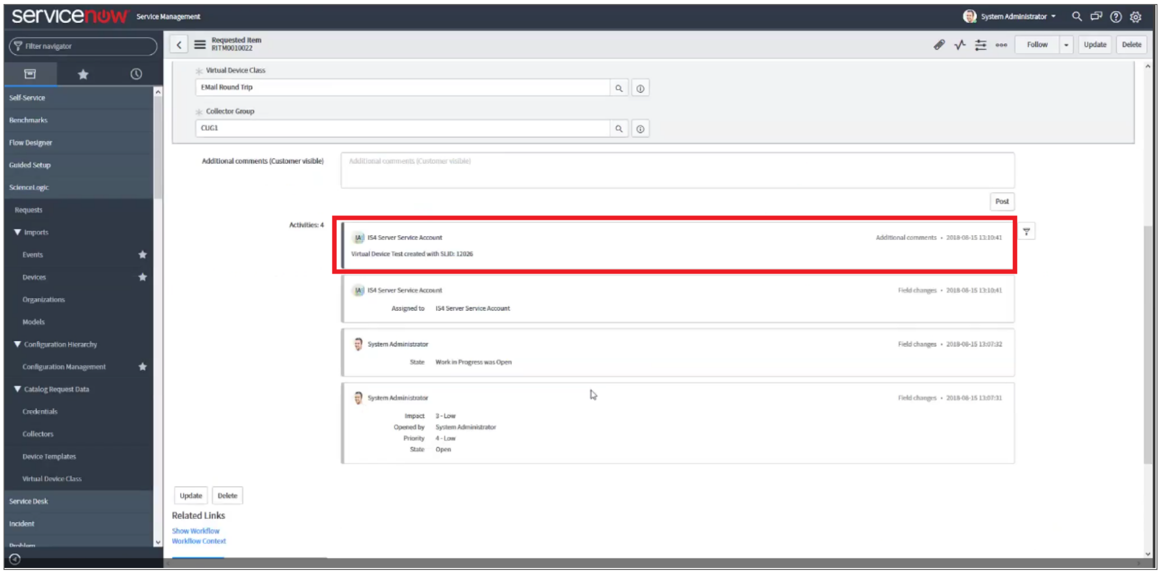
ServiceNow" integration application to make sure that the device or devices were discovered.

13. For a standard device discovery, go to ServiceNow and perform the following:

    - Scroll down to the **Activities** pane to verify that you have a comment stating the discovery completed.
    - In SL1, navigate to the **Discovery Control Panel** page (Registry > Manage > Discovery) and verify that SL1 created a new discovery session with that ID.

14. For a virtual device discovery, go to ServiceNow and perform the following:

    - Scroll down to the **Activities** pane to verify that you have a comment stating "Virtual Device <name> Created with SLID: <new id>"



Requested Item
RITM0010022

Virtual Device Class: EMail Round Trip

Collector Group: CUG1

Additional comments (Customer visible)

Activities: 4

IS4 Server Service Account — Additional comments • 2018-08-15 13:10:41
Virtual Device Test created with SLID: 12026

IS4 Server Service Account — Field changes • 2018-08-15 13:10:41
Assigned to: IS4 Server Service Account

System Administrator — Field changes • 2018-08-15 13:07:32
State: Work in Progress was Open

System Administrator — Field changes • 2018-08-15 13:07:31
Impact: 3 - Low
Opened by: System Administrator
Priority: 4 - Low
State: Open

- In SL1, navigate to the **Device Manager** page (Registry > Device Manager) and verify that SL1 created a new device with that device ID.

Discovering One or More Devices from ServiceNow to SL1

If you want to quickly select one or more CIs in ServiceNow for monitoring in SL1, you can use the *Monitor Device List* option from the **Configuration Items** list view, or the *Monitor Device* option from the Configuration Item detail view.

This feature uses registered events in ServiceNow that are queued to ServiceNow Event Management to trigger actions in the Integration Service. Also, this method is just an example of one of many ways to trigger a registered event. For more information about registered events, including examples of other triggering events you can define in ServiceNow, see the *ServiceNow Registered Events* appendix.

You will need to create a discovery template for a discovery process created on the **Service Catalog** page before you can discover devices using that template on the **Configuration Items** page. A template saves all of the discovery settings except for the IP addresses. You can access existing templates on the **Catalog Template** page in ServiceNow (ScienceLogic > Automations > Catalog Templates).

To discover one or more devices from ServiceNow:

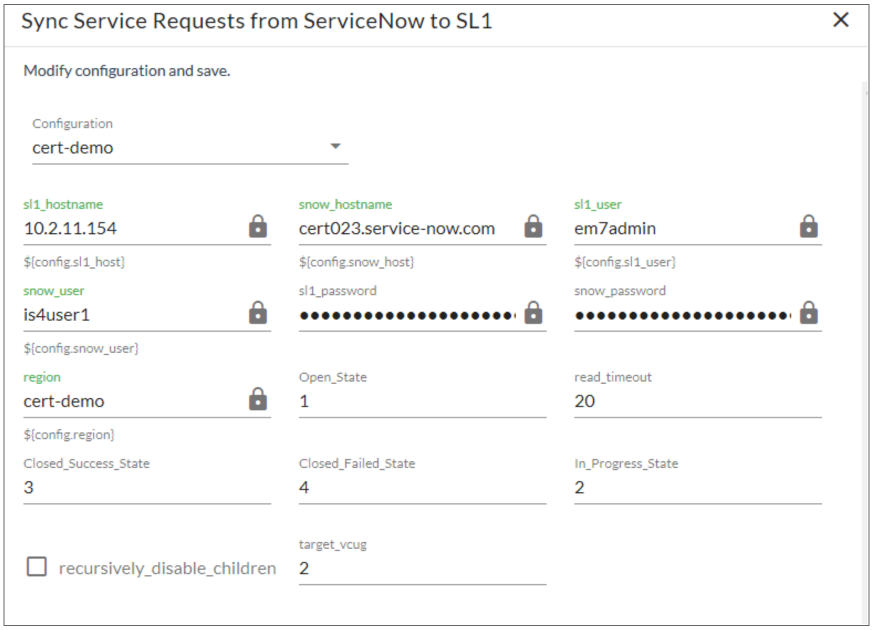
1. In ServiceNow, navigate to the **Configuration Items** page.
2. From the list view, select the CI or CIs (devices) that you want to discover.

> **NOTE:** A CI in ServiceNow must be aligned with a company in ServiceNow, or the service request will be canceled. Also, that company must be associated with a ScienceLogic Region.

3. Right-click anywhere in the window and select *Monitor Device List* from the pop-up menu. A **Select Discovery Template** dialog box appears.

> **TIP:** You can also select a specific CI from the list view and click the *Monitor Device* option from the Configuration Item detail view. You will also need to use an existing template for this process.

4. Select a discovery template to use for the current discovery.
5. Click **[OK]** to use the template. ServiceNow generates a new service request for **Device Discovery** for each CI.
6. In the Integration Service user interface, select the "Sync Service Requests from ServiceNow to SL1" integration application from the **Integrations** page and click **[Configure]** (⚙) on the Application detail page. The **Configuration** pane appears:



Sync Service Requests from ServiceNow to SL1 ✕

Modify configuration and save.

Configuration: cert-demo

sl1_hostname: 10.2.11.154 🔒
${config.sl1_host}

snow_hostname: cert023.service-now.com 🔒
${config.snow_host}

sl1_user: em7admin 🔒
${config.sl1_user}

snow_user: is4user1 🔒
${config.snow_user}

sl1_password: •••••••••••••••••••• 🔒

snow_password: •••••••••••••••••••• 🔒

region: cert-demo 🔒
${config.region}

Open_State: 1

read_timeout: 20

Closed_Success_State: 3

Closed_Failed_State: 4

In_Progress_State: 2

☐ recursively_disable_children

target_vcug: 2

7. Complete the following fields, as needed:

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

   - ***Open_State***. The State ID from ServiceNow that specifies which Requested Items (RITMs) to pull and process. The default is 1.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
   - ***Closed_Success_State***. The State ID for a successfully created virtual device. The State ID for a successful run changes from 1 to 2 and then ends with 4. The default is 3.
   - ***Closed_Failed_State*** The State ID for failed discoveries or failed virtual device creation, usually caused by invalid payloads. The State ID for a failed run changes from 1 to 2 and then ends with 4. The default is 4.
   - ***In_Progress_State***. The State ID for RITMs for a running discovery. The default is 2.
   - ***recursively_disable_children***. Leave this field blank.
   - ***target_vcug***. Leave this field blank.

8. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
9. Click **[Run]** (▶) to run the integration application.
10. Go to the **Integrations** page and run the "Sync Devices from SL1 to ServiceNow" integration application to make sure that the device or devices were discovered.

Decommissioning Devices

If you want to quickly select one or more CIs in ServiceNow for to remove from monitoring (or "decommission") in SL1, you can use the *Device Monitoring Removal list* option from the **Configuration Items** list view, or the *Monitoring Removal* option from the Configuration Item detail view.

You then use the "Sync Service Requests from ServiceNow to SL1" integration application to decommission the devices that you no longer want to monitor. Running this integration application takes the list of synced devices in the service request and moves them to an SL1 Virtual Collector Group (VCUG). The "Sync Service Requests from ServiceNow to SL1" integration application was formerly named "Sync Discovery Session Requests from ServiceNow to SL1".

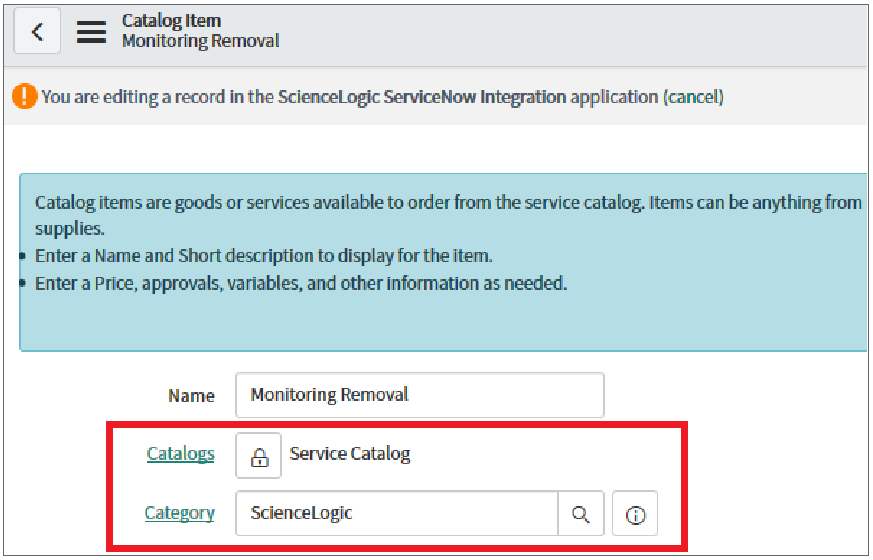
> WARNING: If you move a parent device to a new VCUG, then all of its children move as well. If you move a child directly, only the child moves.

This feature uses registered events in ServiceNow that are queued to ServiceNow Event Management to trigger actions in the Integration Service. Also, this method is just an example of one of many ways to trigger a registered event. For more information about registered events, including examples of other triggering events you can define in ServiceNow, see the ***ServiceNow Registered Events*** appendix.

Activating the ServiceNow Service Request for Monitoring Removal

To activate the ServiceNow service request for Device Decommission:

1. In ServiceNow, search for "Maintain Items" in the filter navigator.
2. Go to **Service Catalog > Catalog Definitions > Maintain Items** and type "ScienceLogic" in the Category field.
3. Open the "Monitoring Removal" service request and ensure that the **Catalogs** and **Category** fields are complete. Add the relevant information if the fields are blank. For example:



Catalog Item
Monitoring Removal

You are editing a record in the ScienceLogic ServiceNow Integration application (cancel)

Catalog items are goods or services available to order from the service catalog. Items can be anything from supplies.

- Enter a Name and Short description to display for the item.
- Enter a Price, approvals, variables, and other information as needed.

Name: Monitoring Removal

Catalogs: Service Catalog

Category: ScienceLogic

NOTE: Do not set the **Category** to a *Change Request*.

4. If you need to update these fields, click the "To edit this record click ***here***" link at the top of the detail page.
5. Update the fields and click the **[Update]** button to save your changes.
6. From the **Catalog Items** page, click the check box for the **Monitoring Removal** service request and click the **[Activate]** button at the bottom of the **Catalog Items** window.
7. Navigate to the relevant catalog for the service request. For example, if you selected *Service Catalog*, then type "Service Catalog" in the filter navigator, or select **Self-Service > Service Catalog** to view the new service requests.

Removing Devices from Monitoring

To decommission Configuration Items (devices) in ServiceNow that you no longer want to monitor:

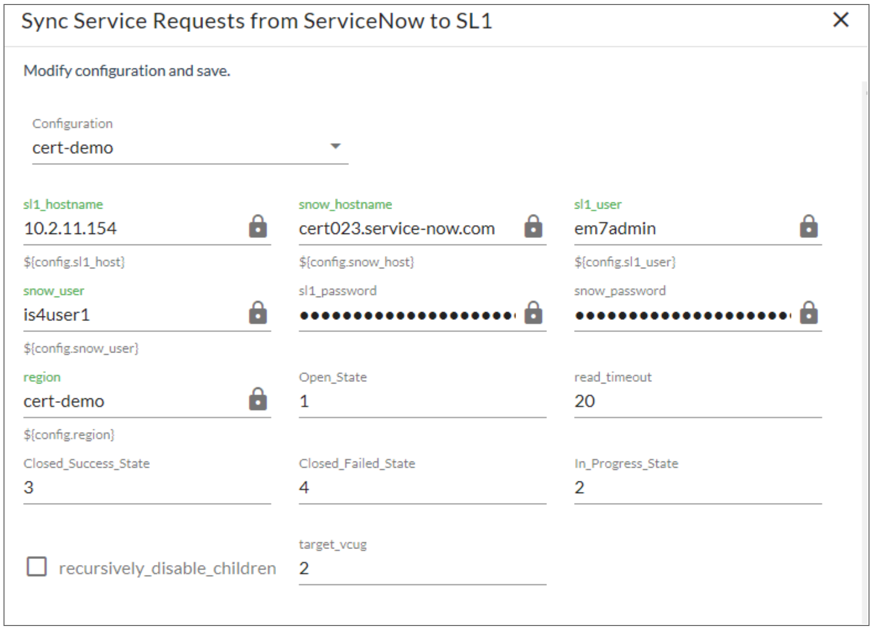
1. In ServiceNow, navigate to the **Configuration Items** window.
2. From the list view, select the CI or CIs (devices) that you want to decommission.

NOTE: A CI in ServiceNow must be aligned with a company in ServiceNow, or the service request will be canceled. Also, a company must be associated with a ScienceLogic Region.

3. Right-click anywhere on the window and select *Device Monitoring Removal list* from the pop-up menu. A dialog box appears.
4. Click **[OK]** to remove the CI or CIs from monitoring. ServiceNow generates a new service request for **Monitoring Removal** for each CI.

TIP: You can also select a specific CI from the list view and click the *Monitoring Removal* option from the Configuration Item detail view.

5. In the Integration Service user interface, select the "Sync Service Requests from ServiceNow to SL1" integration application from the **Integrations** page and click **[Configure]** (⚙) on the application detail page. The **Configuration** pane appears:



Sync Service Requests from ServiceNow to SL1

Modify configuration and save.

Configuration: cert-demo

sl1_hostname: 10.2.11.154
${config.sl1_host}

snow_hostname: cert023.service-now.com
${config.snow_host}

sl1_user: em7admin
${config.sl1_user}

snow_user: is4user1
${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

region: cert-demo
${config.region}

Open_State: 1

read_timeout: 20

Closed_Success_State: 3

Closed_Failed_State: 4

In_Progress_State: 2

☐ recursively_disable_children

target_vcug: 2

6. Complete the following field:

   - ***Configuration***. Select the relevant configuration object to align with this integration application. You cannot edit fields that are populated by the configuration object. Required.
   - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.
   - ***recursively_disable_children***. Check this option to move all child devices of the devices you are decommissioning to the VCUG. If this option is not checked and a parent device is in the disable request, the parent device will be skipped with a warning message.
   - ***target_vcug***. Specify the ID of the SL1 Virtual Collection Group (VCUG) you created to hold the devices on the **Collector Group Settings** page (System > Settings > Collector Groups). If this value is null, the integration application will attempt to pull the value from the ***target_vcug*** field in the "Delete Devices from SL1" integration application.

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

8. Click **[Run]** (▶) to run the integration application.

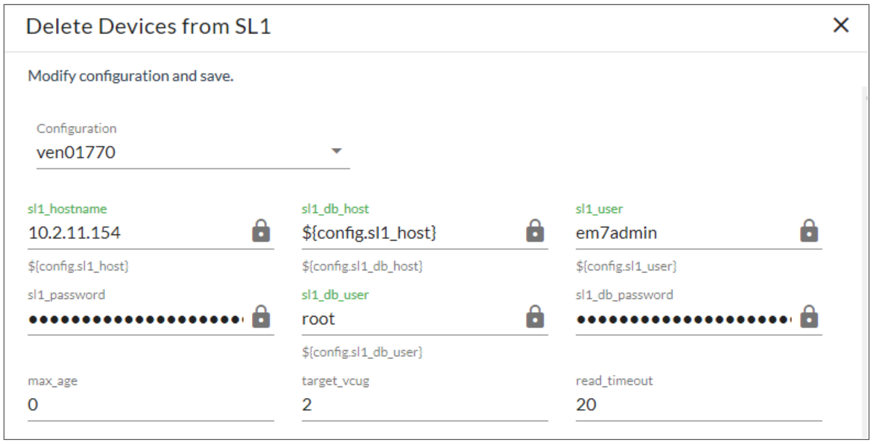
Deleting Devices

The "Delete Devices from SL1" integration application lets you delete devices in a specific Virtual Collector Group (VCUG) if those devices have not been modified in SL1 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 VCUG:

1. In the Integration Service user interface, run the "Sync Service Requests from ServiceNow to SL1" integration application to pull a list of decommissioned devices that you no longer want to monitor. For more information, see ***Decommissioning Devices***.

2. On the **Integrations** page, select the "Delete Devices from SL1" integration 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 — ${config.sl1_host}

sl1_db_host: ${config.sl1_host} — ${config.sl1_db_host}

sl1_user: em7admin — ${config.sl1_user}

sl1_password: ••••••••••••••••••••

sl1_db_user: root — ${config.sl1_db_user}

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 integration application. You cannot edit fields that are populated by the configuration object. Required.
    - ***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 SL1 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 integration applications to use the ***target_vcug*** value from the "Sync Service Requests from ServiceNow to SL1" integration application.

WARNING: If you specify a value to ***target_vcug*** here, the "Delete Devices from SL1" application will use that value instead of the ***target_vcug*** value from the "Sync Service Requests from ServiceNow to SL1" application.

    - ***read_timeout***. Specify the maximum amount of time in seconds the integration application should wait for a response before timing out. The default is 20 seconds.

4. Click **[Save]** and wait for the "App & Config modifications saved" pop-up message to appear. The **Configuration** pane automatically closes after this message appears.
5. Click **[Run]** (▶) to run the integration application.

Scheduling Integration Applications

Using the Integration Service user interface, you can configure integration 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:

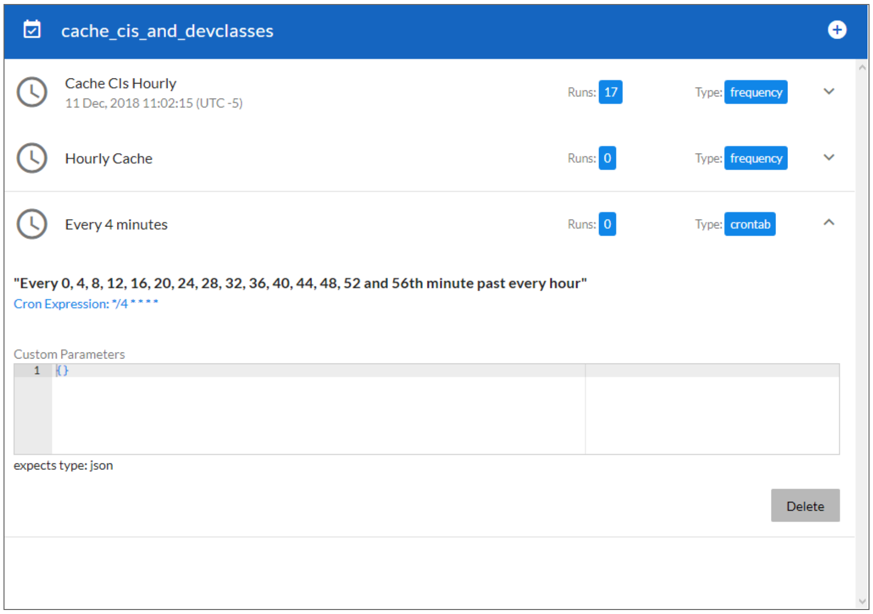
- "Cache ServiceNow CIs and SL1 Device Classes"
- "Sync Devices from SL1 to ServiceNow"
- "Sync Interfaces from SL1 to ServiceNow"

TIP: ScienceLogic recommends that you schedule these integration 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 integration application in the Integration Service user interface. When creating each schedule, you can specify the queue and the configuration file for that integration application.

To schedule an integration application:

1. On the **Integrations** page ( ), click **[Schedule]** for the integration application you want to schedule. The **Schedule** window appears, displaying any existing schedules for that application:



cache_cis_and_devclasses

Cache CIs Hourly
11 Dec, 2018 11:02:15 (UTC -5)
Runs: 17 Type: frequency

Hourly Cache
Runs: 0 Type: frequency

Every 4 minutes
Runs: 0 Type: crontab

"Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52 and 56th minute past every hour"
Cron Expression: */4 * * * *

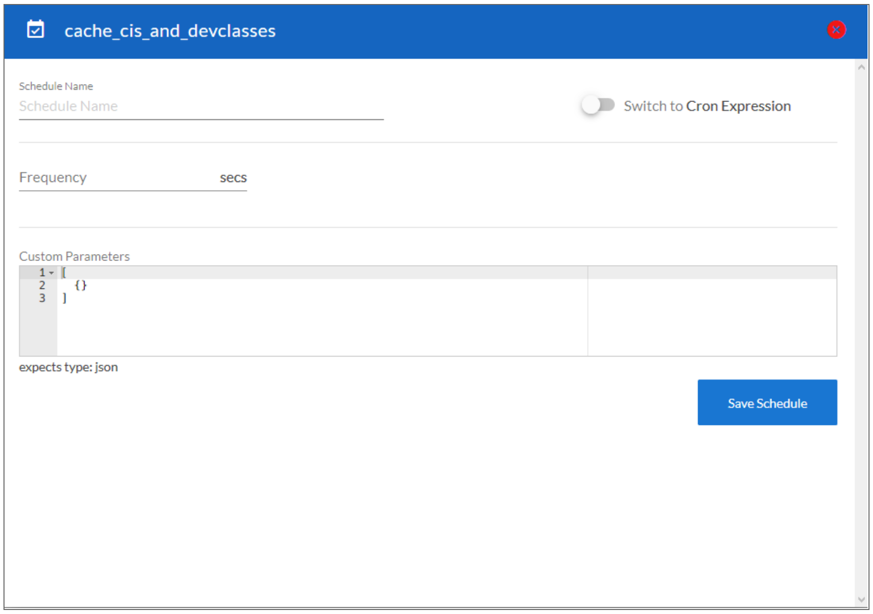
Custom Parameters

```
{}
```

expects type: json

NOTE: If you set up a schedule using a cron expression, the details of that schedule display in a more readable format in this list. For example, if you set up a cron expression of ***/4 * * * ***, the schedule on this window includes the cron expression along with an explanation of that expression: "*Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, and 56th minute past every hour*".

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

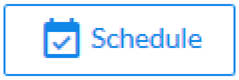


```
[
  {}
]
```

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

   - ***Schedule Name***. Type a name for the schedule.
   - ***Switch to***. Use this toggle to switch between a cron expression and setting the frequency in seconds.
     - ***Cron expression***. Select this option to schedule the integration 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 integration.
   - ***Custom Parameters***. Type any JSON parameters you want to use for this schedule, such as information about a configuration file or mappings.

5. Click **[Save Schedule]**. The schedule is added to the list of schedules on the initial **Schedule** window. Also, on the **Integrations** page, the word "Scheduled" appears in the **Scheduled** column for this integration application, and the **[Schedule]** button contains a check mark:

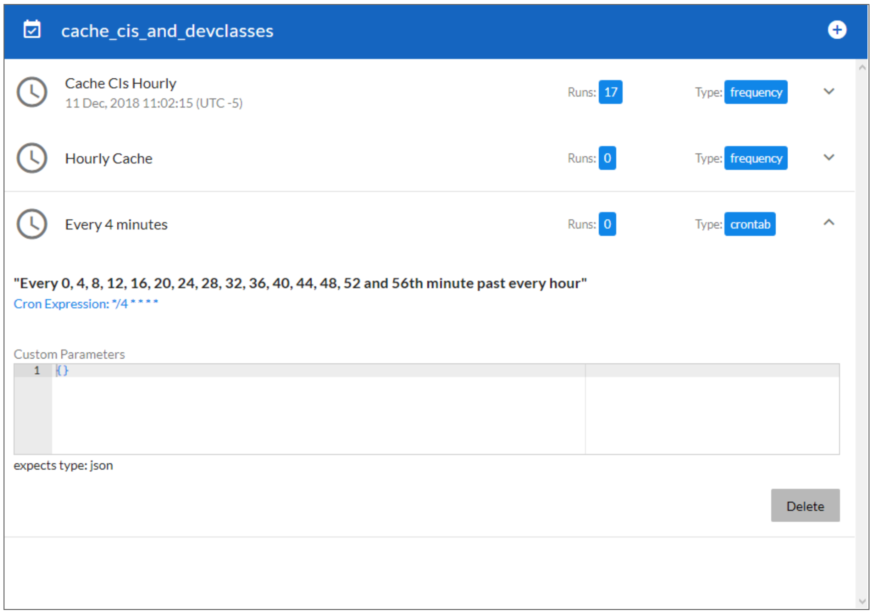


NOTE: After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

1. On the **Integrations** page , click **[Schedule]** for the integration 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:



"Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52 and 56th minute past every hour"
Cron Expression: */4 * * * *

Custom Parameters

```
1 {}
```

expects type: json

3. To delete the selected schedule, click **[Delete]**. The schedule is removed.

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

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" integration application.

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 SL1 to ServiceNow" integration application or in the ***mappings*** section of the "Generate Required CI Relations for ServiceNow" integration.

If you run the "Generate Required CI Relations for ServiceNow" integration 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 GQL payloads had bad data for parent and or child devices:

```
Warning: 10 bad payloads received from SL1. Re-run app in debug to troubleshoot.
```

If you 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

Troubleshooting the CMDB Synchronization PowerPack

Overview

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

This chapter covers the following topics:

Initial Troubleshooting Steps

The Integration Service 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 the Integration Service is talking. There might be additional configurations or actions enabled on ServiceNow or SL1 that result in unexpected behavior. For detailed information about how to perform the steps below, see *Resources for Troubleshooting*.

Integration Service

1. Run the following command:

   ```
   docker service ls
   ```

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

ServiceNow

1. Make a note of the ServiceNow version and Synchronization PowerPack version, if applicable.
2. Make a note of whether the user is running an update set or the Certified Application (also called the "ScienceLogic SL1: CMDB & Incident Automation" application).
3. Make a note of the ServiceNow integration application that is failing on the Integration Service.
4. Make a note of what step is failing in the integration 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 the Integration Service and troubleshooting commands for Docker, Couchbase, and the Integration Service API.

Useful Integration Service Ports

- **http://<*IP of Integration Service*>:8081**. Provides access to Docker Visualizer, a visualizer for Docker Swarm.
- **https://<*IP of Integration Service*>:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://<*IP of Integration Service*>:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by Integration Service workers.

- **https://<*IP of Integration Service*>/flower**. Provides access to Flower, a tool for monitoring and administrating Celery clusters.

Helpful Docker Commands

The Integration Service is a set of services that are containerized using Docker. For more information about Docker, see the Docker tutorial.

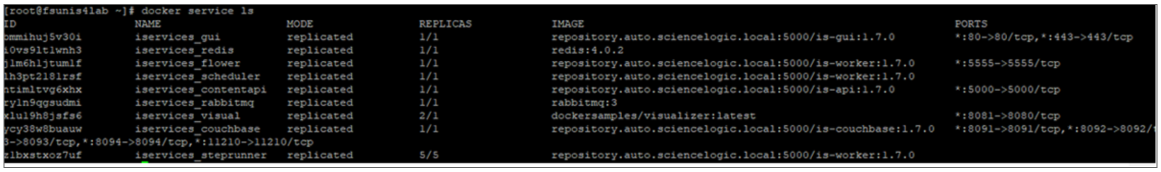
Use the following Docker commands for troubleshooting and diagnosing issues with the Integration Service:

Viewing Container Versions and Status

To view the Integration Service version, SSH to your Integration Service 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 the Integration Service:



```
[root@fsunis4lab ~]# docker service ls
ID                  NAME                  MODE                REPLICAS            IMAGE                                                         PORTS
ommihuj5v30i        iservices_gui         replicated          1/1                 repository.auto.sciencelogic.local:5000/is-gui:1.7.0          *:80->80/tcp,*:443->443/tcp
i0vs91tlwnh3        iservices_redis       replicated          1/1                 redis:4.0.2
jlm6hljtumlf        iservices_flower      replicated          1/1                 repository.auto.sciencelogic.local:5000/is-worker:1.7.0       *:5555->5555/tcp
lh3pt218lrsf        iservices_scheduler   replicated          1/1                 repository.auto.sciencelogic.local:5000/is-worker:1.7.0
ntimltvg6xhx        iservices_contentapi  replicated          1/1                 repository.auto.sciencelogic.local:5000/is-api:1.7.0          *:5000->5000/tcp
ryln9qgsudmi        iservices_rabbitmq    replicated          1/1                 rabbitmq:3
xlul9h8jsfs6        iservices_visual      replicated          2/1                 dockersamples/visualizer:latest                               *:8081->8080/tcp
ycy38w8buauw        iservices_couchbase   replicated          1/1                 repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0    *:8091->8091/tcp,*:8092->8092/
3->8093/tcp,*:8094->8094/tcp,*:11210->11210/tcp
zlbxstxoz7uf        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 Integration Service Services

Run the following command to stop all Integration Service services:

```
docker stack rm iservices
```

Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

NOTE: Restarting Docker does not clear the queue.

Viewing Logs for a Specific Service

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

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

Some common examples include the following:

```
docker service logs –f iservices_couchbase

docker service logs –f iservices_steprunner

docker service logs –f iservices_contentapi
```

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

Clearing RabbitMQ Volume

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

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

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

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

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

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

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

Re-deploy the Integration Service by running the following command:

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

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

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

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

Dynamically Scaling for More Workers

Run the following command:

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

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnosis Tools

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

- ***Docker PowerPack***. This PowerPack monitors your Linux-based Integration Service server with SSH (the Integration Service ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your Integration Service server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor the Integration Service, see the "Using SL1 to Monitor the Integration Service" chapter in the ***Integration Service Platform*** manual.
- ***Flower.*** This web interface tool can be found at the /flower endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the Integration Service 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 Integration Service API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see ***Retrieving Additional Debug Information***.
- ***Application Logs***. All applications generate a log file specific to that application. These log files can be found at /var/log/iservices and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.

- ***Step Logs***. Step logs display the log output for a specific step in the application. These step logs can be accessed via the Integration Service user interface by clicking on a step in an integration 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 the Integration Service 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 integration applications with "loglevel": 10, those integration 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, POST the following to the API endpoint:

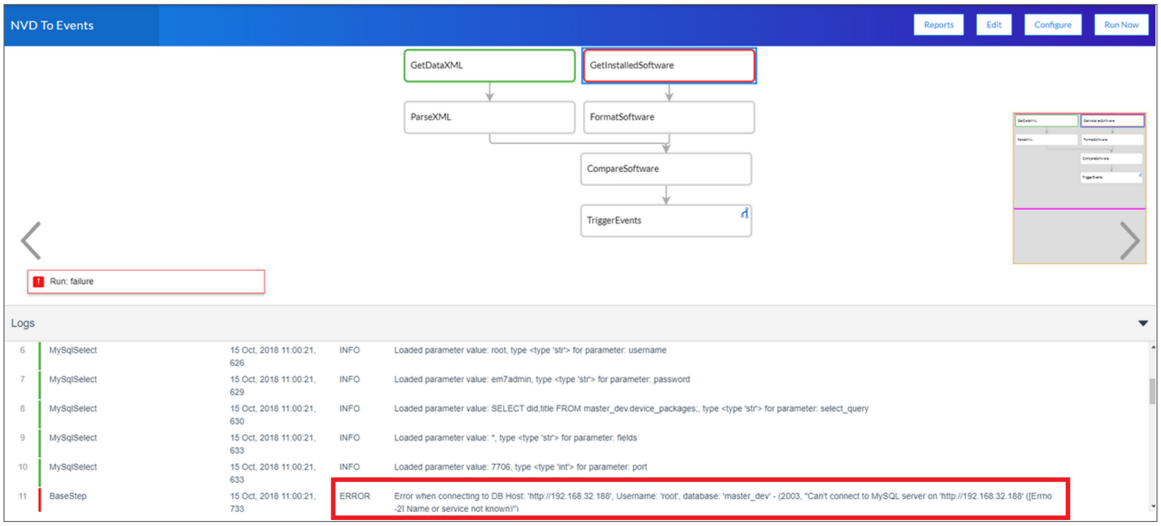
```
https://<integration_service>/api/v1/applications/run
```

Request body:

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

After running the integration application in Debug Mode, go back to the Integration Service user interface and review the step logs to see detailed debug output for each step in the integration application. When run in Debug Mode, the step log output shows additional debug statements such as "Saved data for next step", which displays the data being sent from one step to the next.

This information is especially helpful when trying to understand why an integration application or step failed:



```
6   MySqlSelect   15 Oct, 2018 11:00:21, 626   INFO    Loaded parameter value: root, type <type 'str'> for parameter: username
7   MySqlSelect   15 Oct, 2018 11:00:21, 629   INFO    Loaded parameter value: em7admin, type <type 'str'> for parameter: password
8   MySqlSelect   15 Oct, 2018 11:00:21, 630   INFO    Loaded parameter value: SELECT did,title FROM master_dev.device_packages;, type <type 'str'> for parameter: select_query
9   MySqlSelect   15 Oct, 2018 11:00:21, 633   INFO    Loaded parameter value: *, type <type 'str'> for parameter: fields
10  MySqlSelect   15 Oct, 2018 11:00:21, 633   INFO    Loaded parameter value: 7706, type <type 'int'> for parameter: port
11  BaseStep      15 Oct, 2018 11:00:21, 733   ERROR   Error when connecting to DB Host: 'http://192.168.32.188', Username: 'root', database: 'master_dev' - (2003, "Can't connect to MySQL server on 'http://192.168.32.188' ([Errno -2] Name or service not known)")
```

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

1. SSH to the Integration Service instance.
2. Run the following command:

```
curl -v -k -u isadmin:em7admin -X POST "https://<your_
hostname>/api/v1/applications/run" -H 'Content-Type: application/json' -H 'cache-
control: no-cache' -d '{"name": "interface_sync_sciencelogic_to_
servicenow","params": {"loglevel": 10}}'
```

Troubleshooting CMDB Sync

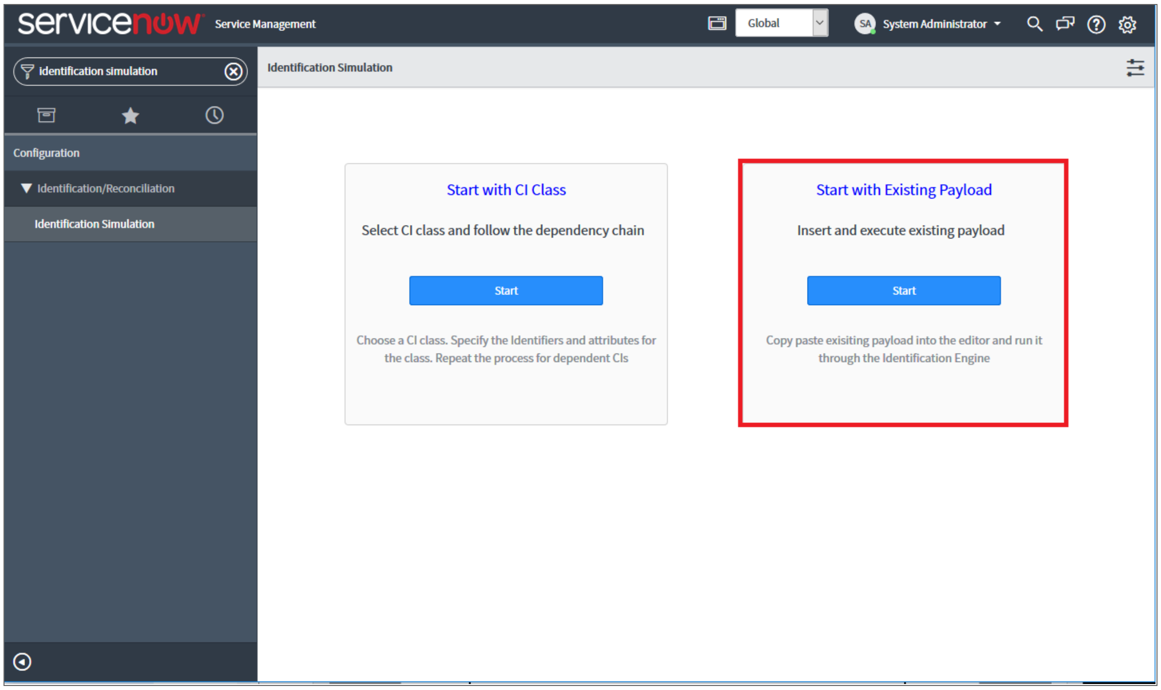
This section contains specific troubleshooting steps for the CMDB Synchronization PowerPack.

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**.


servicenow
Service Management
Global
SA System Administrator
identification simulation
Configuration
Identification/Reconciliation
Identification Simulation
Identification Simulation

Start with CI Class

Select CI class and follow the dependency chain

Start

Choose a CI class. Specify the Identifiers and attributes for the class. Repeat the process for dependent CIs

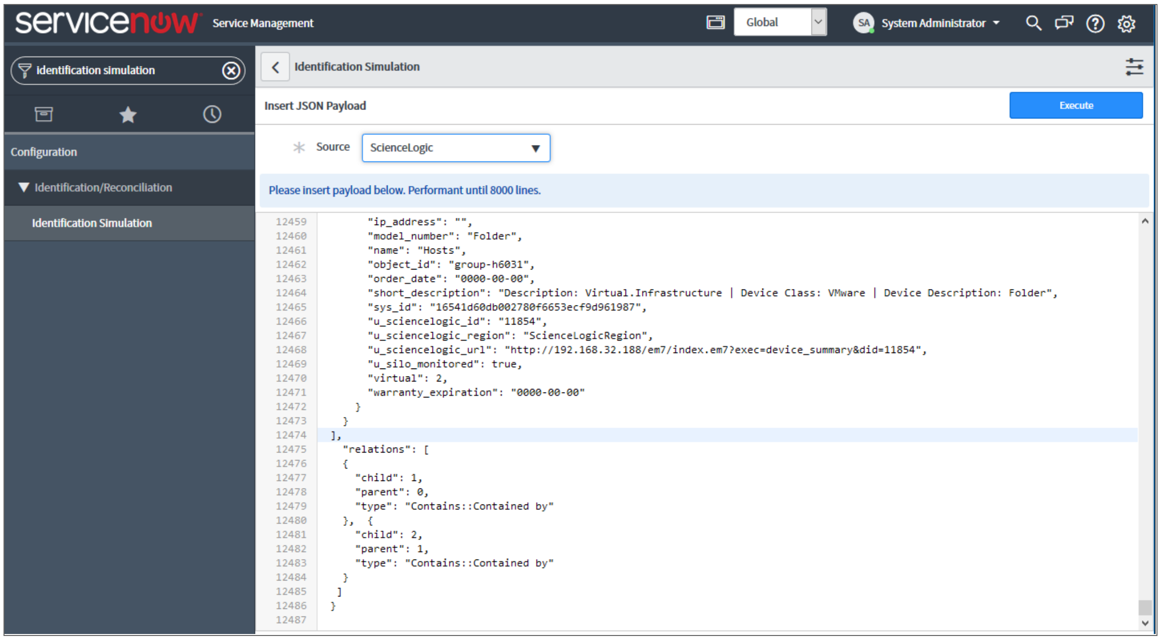
Start with Existing Payload

Insert and execute existing payload

Start

Copy paste exisiting payload into the editor and run it through the Identification Engine

7. On the **Identification Simulation** page, click the **[Start]** button in the **Start with Existing Payload** section. The Insert JSON Payload page appears:


servicenow
Service Management
Global
SA System Administrator
identification simulation
Configuration
Identification/Reconciliation
Identification Simulation
Identification Simulation
Insert JSON Payload
Execute

Source: ScienceLogic

Please insert payload below. Performant until 8000 lines.

```
          "ip_address": "",
          "model_number": "Folder",
          "name": "Hosts",
          "object_id": "group-h6031",
          "order_date": "0000-00-00",
          "short_description": "Description: Virtual.Infrastructure | Device Class: VMware | Device Description: Folder",
          "sys_id": "16541d60db002780f6653ecf9d961987",
          "u_sciencelogic_id": "11854",
          "u_sciencelogic_region": "ScienceLogicRegion",
          "u_sciencelogic_url": "http://192.168.32.188/em7/index.em7?exec=device_summary&did=11854",
          "u_silo_monitored": true,
          "virtual": 2,
          "warranty_expiration": "0000-00-00"
        }
      }
  ],
    "relations": [
    {
      "child": 1,
      "parent": 0,
      "type": "Contains::Contained by"
    },  {
      "child": 2,
      "parent": 1,
      "type": "Contains::Contained by"
    }
   ]
  }
```

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.

Enabling Debugging of the Configuration Item Payload

You must have administrator-level permissions in ServiceNow to access the system properties and enable debugging of the Configuration Item payload in the *ServiceNow Identification and Reconciliation module*.

To enable debugging of the Configuration Item 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 SL1 to ServiceNow" integration application. The system logs will have “identfication_engine” as the source, and the log messages will contain `identification_engine : Input`.
4. Copy the payload beginning from `{“items”` to the end of the message. For example:

```
Message: {"items":[{"className":"","values":{"discovery_
source":"ScienceLogic","mac_address":"9E:0F:04:0A:12:C7","name":"Postman Test
Server 1","x_sclo_scilogic_id":"1","serial_number":"gJ3Bwkzc8r","model_id":"","ip_
address":"10.10.10.102","manufacturer":"ScienceLogic, Inc.","ram":"16000","x_sclo_
scilogic_region":"Postman"},"lookup":[],"related":[]}],"relations":[]}
```

5. You can run this message through the ScienceLogic endpoint by putting the `{“items”}` bracket within `[]`. For example, send the following message to the endpoint `/api/x_sclo_scilogic/v1/sciencelogic/IdentificationEngine`:

```
Message: [{"items":[{"className":"","values":{"discovery_
source":"ScienceLogic","mac_address":"9E:0F:04:0A:12:C7","name":"Postman Test
Server 1","x_sclo_scilogic_id":"1","serial_number":"gJ3Bwkzc8r","model_id":"","ip_
address":"10.10.10.102","manufacturer":"ScienceLogic, Inc.","ram":"16000","x_sclo_
scilogic_region":"Postman"},"lookup":[],"related":[]}],"relations":[]}]
```

NOTE: The endpoint is different in a domain-separated environment.

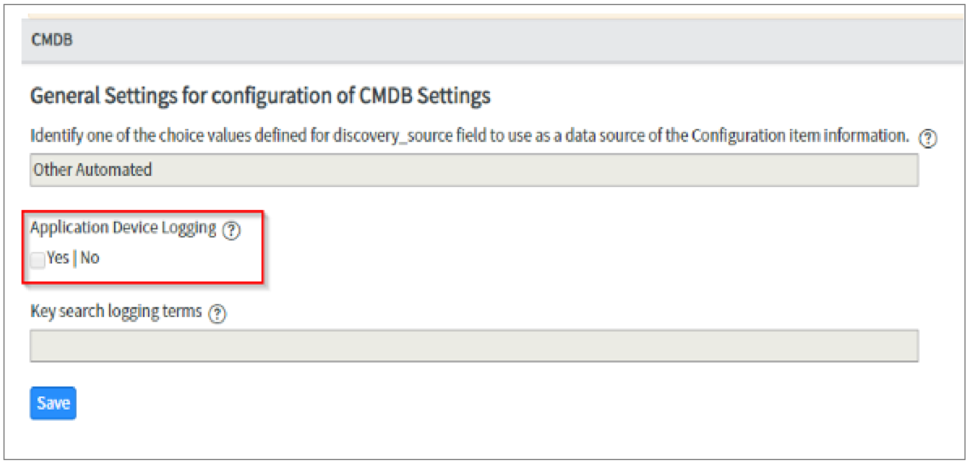
After the identification run is complete, the ServiceNow logs contain additional data about the run.

Enabling Logs in ServiceNow

You can enable device logging in ServiceNow to post errors to the logs in ServiceNow.

To enable ServiceNow device logging:

1. In ServiceNow, search for "device properties" in the filter navigator.
2. Select **ScienceLogic > Device Properties**. A **CMDB** record appears:



CMDB

General Settings for configuration of CMDB Settings

Identify one of the choice values defined for discovery_source field to use as a data source of the Configuration item information. ?

Other Automated

Application Device Logging ?

□ Yes | No

Key search logging terms ?

3. Select Yes in the **Application Device Logging** section and click **[Save]**.

Locating Missing Device Data after Device Sync

After you run a Device Sync for SL1 Devices and ServiceNow CIs, you might encounter data attributes that did not get synced between SL1 and ServiceNow, or that data might not be accurate.

To address these situations, you should always start by reviewing your device settings in SL1, your configuration settings in the Integration Service, and your CI settings in ServiceNow.

NOTE: The Device Sync settings are contained in the "Sync Devices from SL1 to ServiceNow" integration application in the Integration Service user interface.

Situations Where Device Data Might be Missing

After configuring and running a Device Sync in the Integration Service, you might find incidents where device attributes in SL1 or ServiceNow are incomplete or missing.

This topic covers the following situations where device data might be incomplete:

- In SL1, you might be using an older version of a PowerPack that does not gather the device data you were expecting.
- In SL1, one of more of the fields on your **Device Investigator** pages were not populated or updated properly.
- ServiceNow might not have received the required device data from SL1 due to configuration issues with the "Sync Devices from SL1 to ServiceNow" integration application in the Integration Service.

Example: Missing Serial Number for a vCenter Device

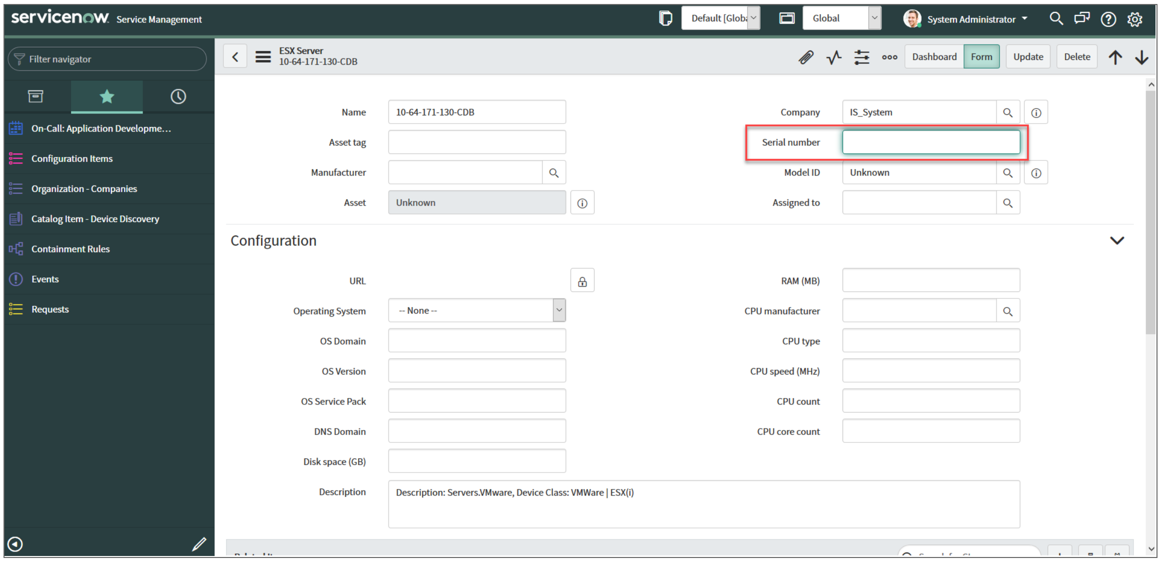
In this example, the **Serial Number** device attributes for VMware ESXi hosts in the "IS_System" Company are not getting properly synced between SL1 and ServiceNow.

The typical workflow for addressing this issue is the following:

1. In ServiceNow, identify the device attribute that did not sync properly with the corresponding CI record.
2. In SL1, locate the device and review the device details.
3. In SL1, review the Dynamic Application that collects data for the missing device attribute.
4. In the Integration Service, check the **Configuration** pane for the "Sync Devices from SL1 to ServiceNow" integration application.

Step 1: In ServiceNow, identify the attribute that did not sync properly

In ServiceNow, go to the **Configuration Items** page and select the CI record for one of the VMware ESXi hosts that you are syncing with SL1:



ESX Server
10-64-171-130-CDB

Name	10-64-171-130-CDB	Company	IS_System
Asset tag		Serial number	
Manufacturer		Model ID	Unknown
Asset	Unknown	Assigned to	

Configuration

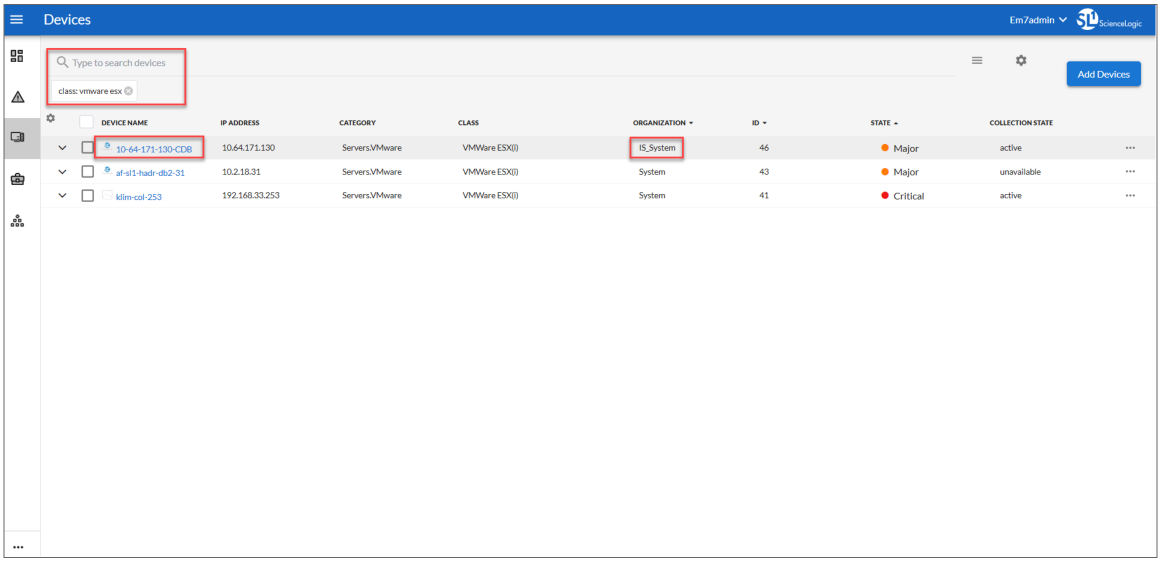
URL		RAM (MB)	
Operating System	-- None --	CPU manufacturer	
OS Domain		CPU type	
OS Version		CPU speed (MHz)	
OS Service Pack		CPU count	
DNS Domain		CPU core count	
Disk space (GB)			
Description	Description: Servers.VMware, Device Class: VMWare | ESX(i)		

In the above image, the ***Serial Number*** field for the ServiceNow CI (SL1 device) named "10-64-171-130-CDB" is empty. Make a note of that missing information and go to Step 2.

TIP: If you know the serial number for the synced device, you can type the serial number into the filter navigator in ServiceNow. If the serial number does not show up in your search, then you know that it has not been synced with Device Sync.

Step 2: In SL1, locate the SL1 device and review the device details

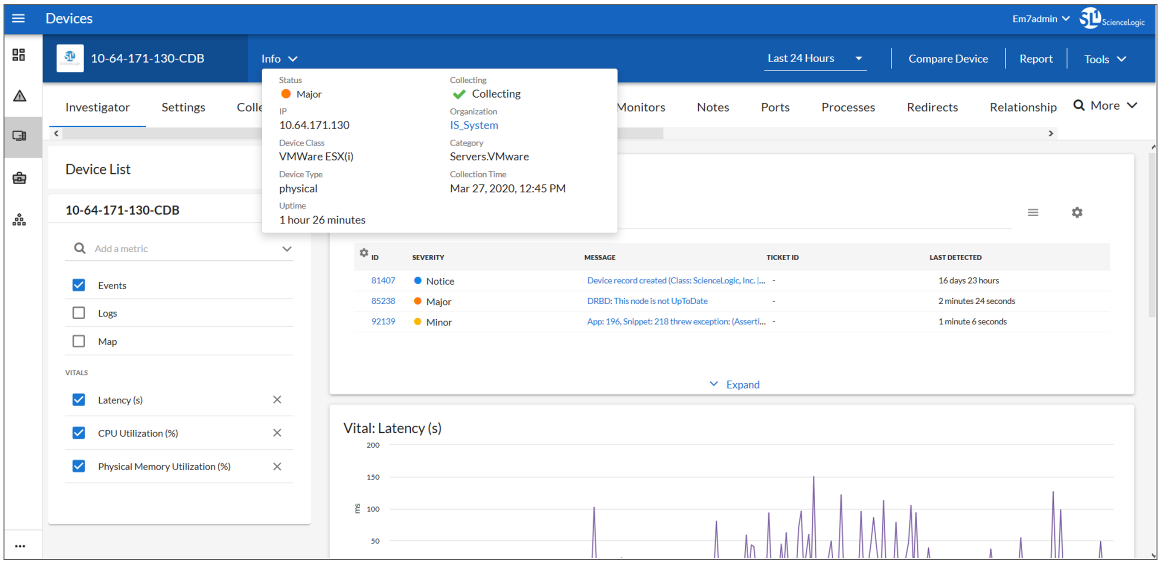
Now that you know what data is missing, go to the **Devices** page in SL1 and make sure that the device you are trying to sync exists in SL1. In this example, you are looking for VMware ESXi hosts, so filter for *VMware* on the **Devices** page:



DEVICE NAME	IP ADDRESS	CATEGORY	CLASS	ORGANIZATION	ID	STATE	COLLECTION STATE
10-64-171-130-CDB	10.64.171.130	Servers.VMware	VMWare ESX(i)	IS_System	46	● Major	active
af-sl1-hadr-db2-31	10.2.18.31	Servers.VMware	VMWare ESX(i)	System	43	● Major	unavailable
klim-col-253	192.168.33.253	Servers.VMware	VMWare ESX(i)	System	41	● Critical	active

You find a device named "10-64-171-130-CDB" from the "IS_System" Company, so you know that the device exists in SL1. Click the name of the device from your search results.

On the **Device Investigator** page for that device, click the **Info** drop-down to view the Device Class and other related information about the device:



Status: ● Major
Collecting: ✓ Collecting
IP: 10.64.171.130
Organization: IS_System
Device Class: VMWare ESX(i)
Category: Servers.VMware
Device Type: physical
Collection Time: Mar 27, 2020, 12:45 PM
Uptime: 1 hour 26 minutes

☑ Events
□ Logs
□ Map

VITALS

☑ Latency (s)
☑ CPU Utilization (%)
☑ Physical Memory Utilization (%)

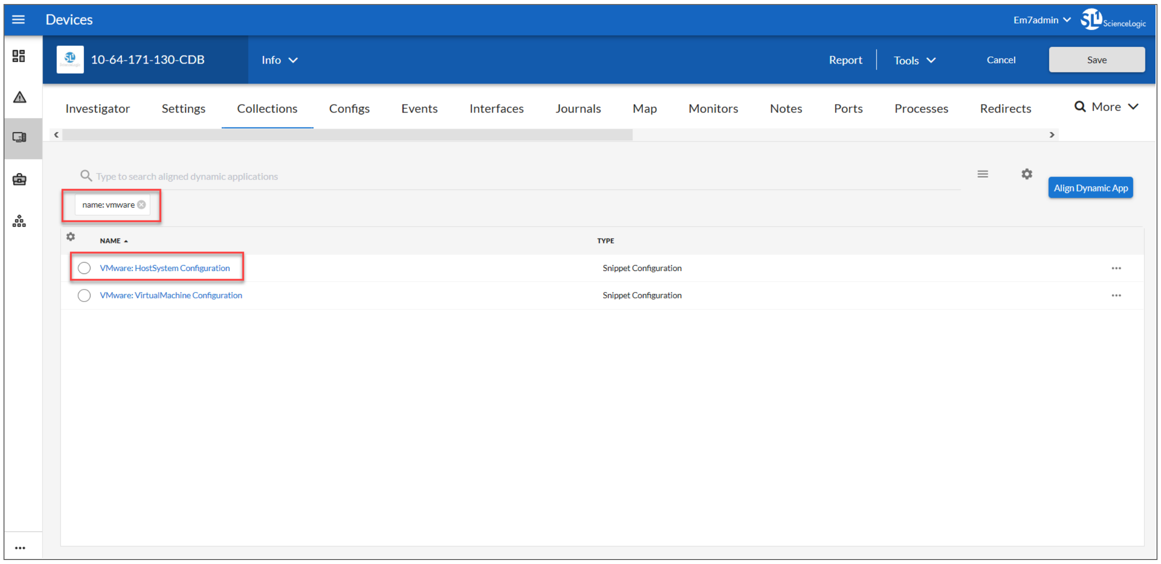
ID	SEVERITY	MESSAGE	TICKET ID	LAST DETECTED
81407	● Notice	Device record created (Class: ScienceLogic, Inc. |...	-	16 days 23 hours
85238	● Major	DRBD: This node is not UpToDate	-	2 minutes 24 seconds
92139	● Minor	App: 196, Snippet: 218 threw exception: (Asserti...	-	1 minute 6 seconds

If the fields on the **Info** drop-down are accurate, go to Step 3.

Step 3: In SL1, review the PowerPacks that collect data for that device

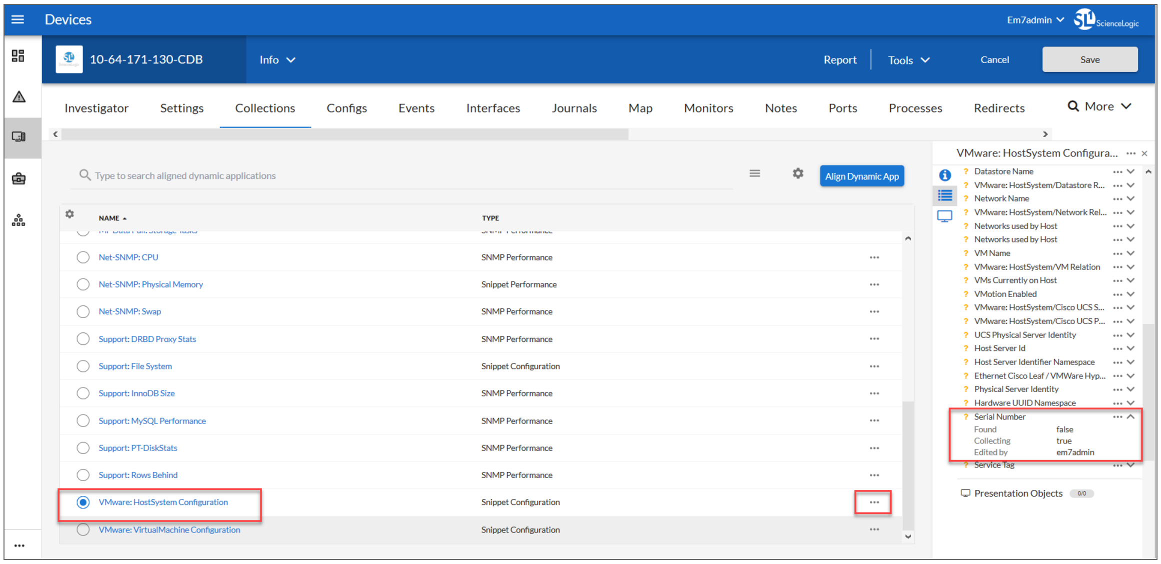
Because the device you are syncing is a VMware ESXi host, you should look for the "VMware: HostSystem Configuration"Dynamic Application from the *VMware: vSphere Base Pack* PowerPack. This Dynamic Application is responsible for capturing the ***Serial Number*** data for VMware devices.

Go to the **[Collections]** tab of the **Device Investigator** page for that device and search for the "VMware: HostSystem Configuration" Dynamic Application:



NAME	TYPE
VMware: HostSystem Configuration	Snippet Configuration
VMware: VirtualMachine Configuration	Snippet Configuration

Select the "VMware: HostSystem Configuration" Dynamic Application and make sure that it is aligned with the device. If needed, click the **Actions** button (···) and select *Enable Collection*. Also, make sure that the "Serial Number" collection object from the list at the right has been found and is collecting data:



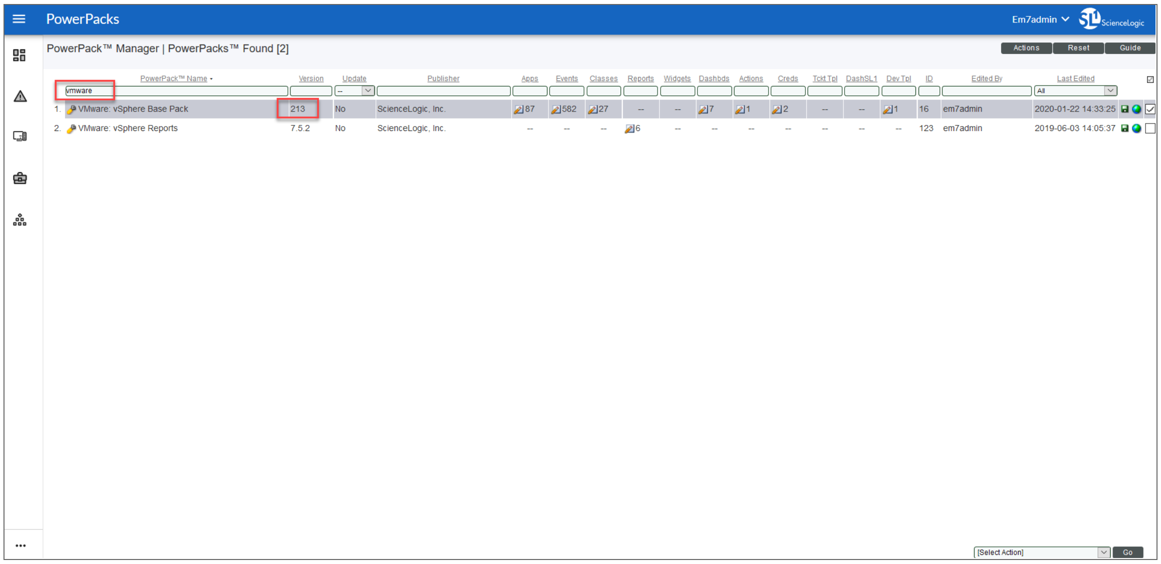
NAME	TYPE
Net-SNMP: CPU	SNMP Performance
Net-SNMP: Physical Memory	Snippet Performance
Net-SNMP: Swap	SNMP Performance
Support: DRBD Proxy Stats	SNMP Performance
Support: File System	Snippet Configuration
Support: InnoDB Size	SNMP Performance
Support: MySQL Performance	SNMP Performance
Support: PT-DiskStats	SNMP Performance
Support: Rows Behind	SNMP Performance
● VMware: HostSystem Configuration	Snippet Configuration
VMware: VirtualMachine Configuration	Snippet Configuration

- Datastore Name
- VMware: HostSystem/Datastore R...
- Network Name
- VMware: HostSystem/Network Rel...
- Networks used by Host
- Networks used by Host
- VM Name
- VMware: HostSystem/VM Relation
- VMs Currently on Host
- VMotion Enabled
- VMware: HostSystem/Cisco UCS S...
- VMware: HostSystem/Cisco UCS P...
- UCS Physical Server Identity
- Host Server Id
- Host Server Identifier Namespace
- Ethernet Cisco Leaf / VMWare Hyp...
- Physical Server Identity
- Hardware UUID Namespace
- Serial Number
- Service Tag

Serial Number	
Found	false
Collecting	true
Edited by	em7admin

You should also check the version of the PowerPack you are using. In this example, versions of the *VMware: vSphere Base Pack* PowerPack *before* version 213 did not collect Serial Number information.

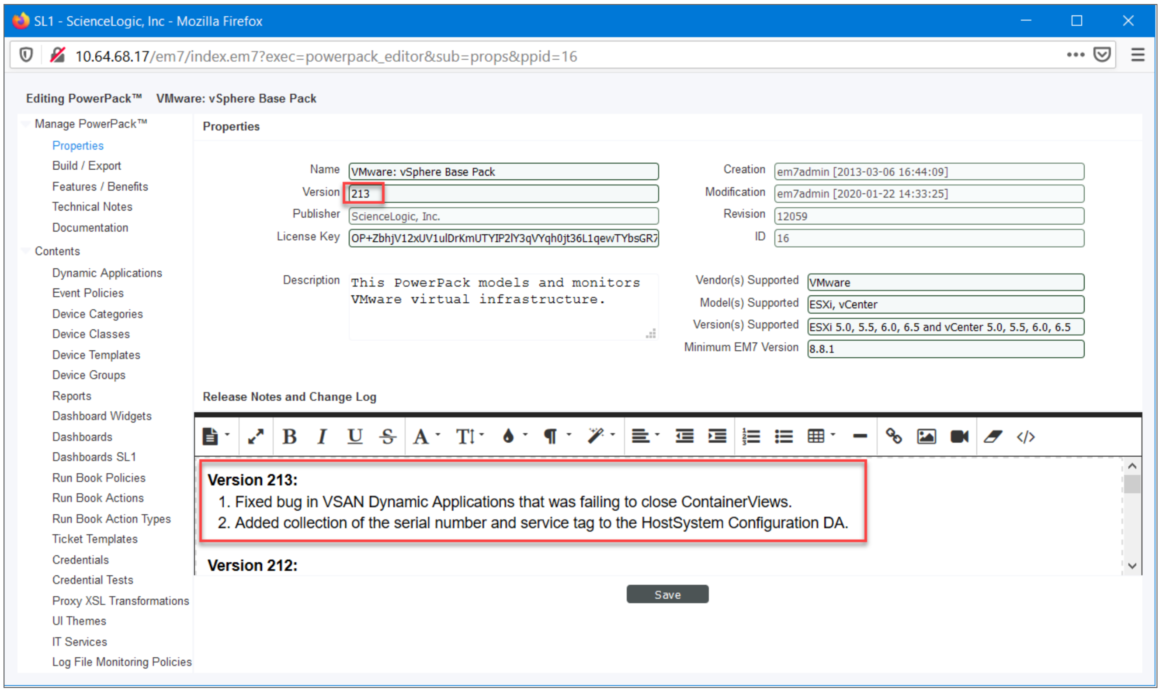
You can find the relevant PowerPack information by searching for the "VMware" PowerPack on the **PowerPacks** page (System > Manage > PowerPacks):



PowerPack™ Manager | PowerPacks™ Found [2]

	PowerPack™ Name	Version	Update	Publisher	Apps	Events	Classes	Reports	Widgets	Dashbds	Actions	Creds	Tckt Tpl	DashSL1	Dev Tpl	ID	Edited By	Last Edited
	vmware		--															All
1.	VMware: vSphere Base Pack	213	No	ScienceLogic, Inc.	87	582	27	--	--	7	1	2	--	--	1	16	em7admin	2020-01-22 14:33:25
2.	VMware: vSphere Reports	7.5.2	No	ScienceLogic, Inc.	--	--	--	6	--	--	--	--	--	--	--	123	em7admin	2019-06-03 14:05:37

Select the *VMware: vSphere Base Pack* PowerPack to view additional details:



Editing PowerPack™ VMware: vSphere Base Pack

Properties

Name: VMware: vSphere Base Pack
Version: 213
Publisher: ScienceLogic, Inc.
License Key: OP+ZbhjV12xUV1ulDrKmUTYIP2lY3qVYqh0jt36L1qewTYbsGR7

Description: This PowerPack models and monitors VMware virtual infrastructure.

Creation: em7admin [2013-03-06 16:44:09]
Modification: em7admin [2020-01-22 14:33:25]
Revision: 12059
ID: 16

Vendor(s) Supported: VMware
Model(s) Supported: ESXi, vCenter
Version(s) Supported: ESXi 5.0, 5.5, 6.0, 6.5 and vCenter 5.0, 5.5, 6.0, 6.5
Minimum EM7 Version: 8.8.1

Release Notes and Change Log

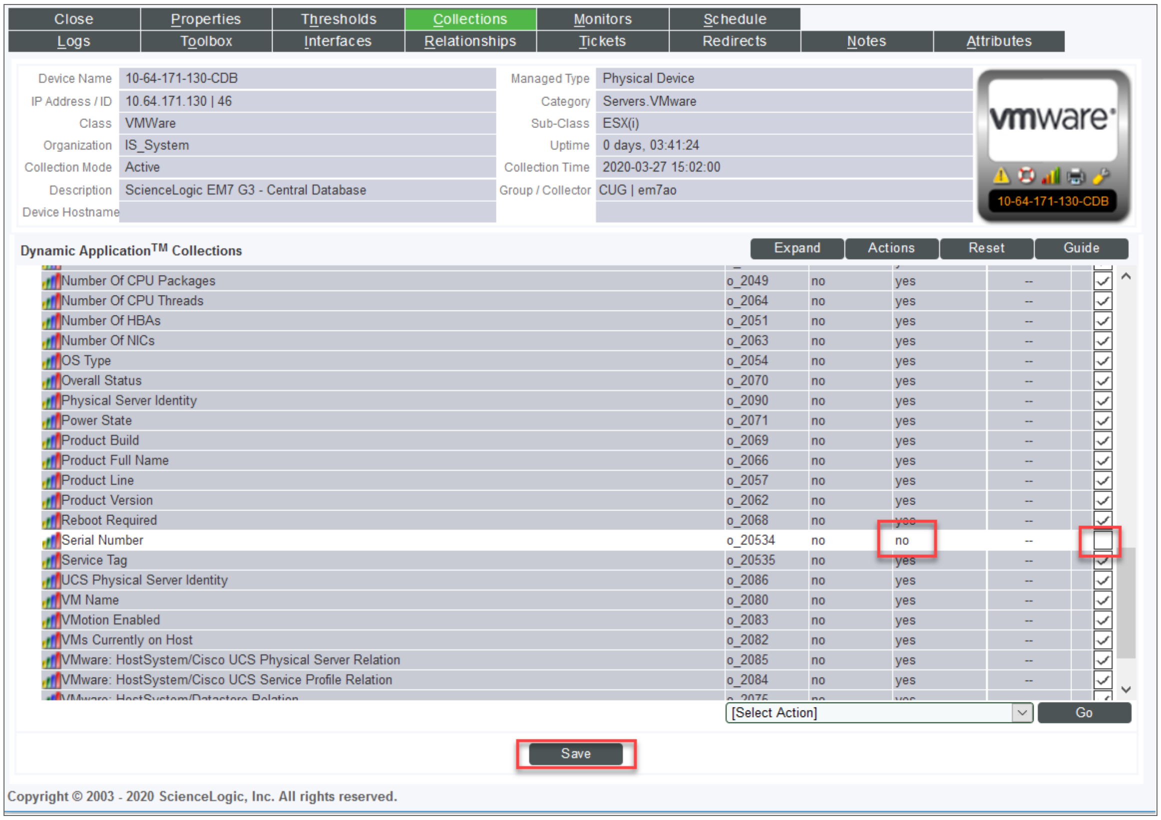
Version 213:

1. Fixed bug in VSAN Dynamic Applications that was failing to close ContainerViews.
2. Added collection of the serial number and service tag to the HostSystem Configuration DA.

Version 212:

You can also view the Dynamic Application information on the **Device Manager** page (Devices > Device Manager) by searching for "10-64-171-130-CDB" device and selecting its wrench icon (🔧) to open the **Device Properties** page.

On the **[Collections]** tab, expand the "VMware: HostSystem Configuration" Dynamic Application in the lower pane and scroll down to the "Serial Number" collection object:



Device Name	10-64-171-130-CDB	Managed Type	Physical Device
IP Address / ID	10.64.171.130 | 46	Category	Servers.VMware
Class	VMWare	Sub-Class	ESX(i)
Organization	IS_System	Uptime	0 days, 03:41:24
Collection Mode	Active	Collection Time	2020-03-27 15:02:00
Description	ScienceLogic EM7 G3 - Central Database	Group / Collector	CUG | em7ao
Device Hostname			

Dynamic Application^{TM} Collections

Number Of CPU Packages	o_2049	no	yes	--	☑
Number Of CPU Threads	o_2064	no	yes	--	☑
Number Of HBAs	o_2051	no	yes	--	☑
Number Of NICs	o_2063	no	yes	--	☑
OS Type	o_2054	no	yes	--	☑
Overall Status	o_2070	no	yes	--	☑
Physical Server Identity	o_2090	no	yes	--	☑
Power State	o_2071	no	yes	--	☑
Product Build	o_2069	no	yes	--	☑
Product Full Name	o_2066	no	yes	--	☑
Product Line	o_2057	no	yes	--	☑
Product Version	o_2062	no	yes	--	☑
Reboot Required	o_2068	no	yes	--	☑
Serial Number	o_20534	no	no	--	□
Service Tag	o_20535	no	yes	--	☑
UCS Physical Server Identity	o_2086	no	yes	--	☑
VM Name	o_2080	no	yes	--	☑
VMotion Enabled	o_2083	no	yes	--	☑
VMs Currently on Host	o_2082	no	yes	--	☑
VMware: HostSystem/Cisco UCS Physical Server Relation	o_2085	no	yes	--	☑
VMware: HostSystem/Cisco UCS Service Profile Relation	o_2084	no	yes	--	☑
VMware: HostSystem/Datastore Relation	o_2075	no	yes		

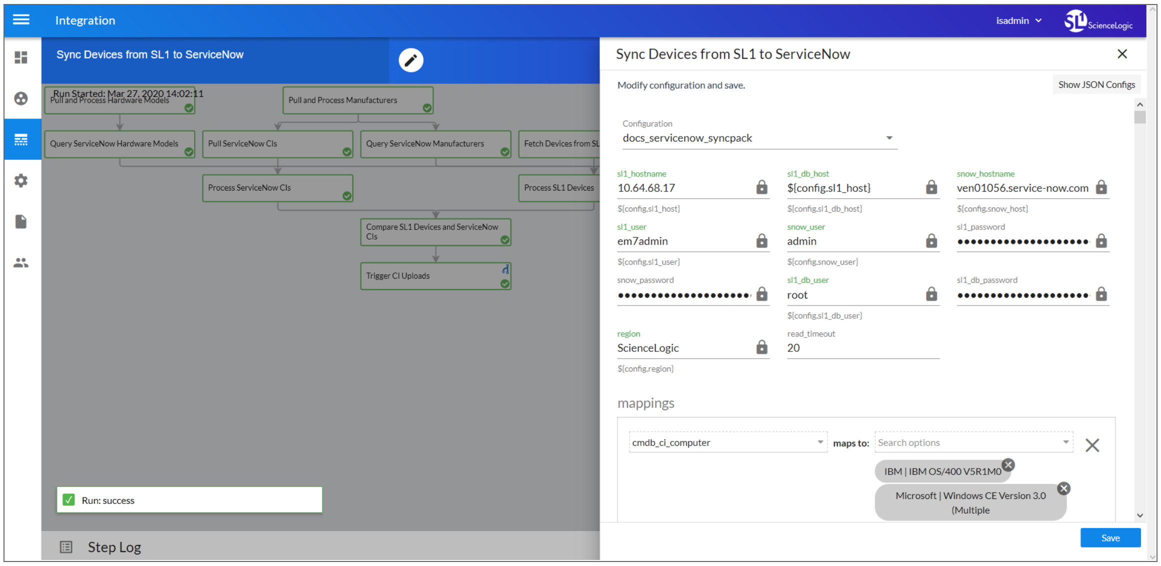
Copyright © 2003 - 2020 ScienceLogic, Inc. All rights reserved.

Make sure that the collection object for "Serial Number" is enabled and collecting data. If either the **Collecting** column displays a "no", as it does in the above image, click the empty check box to enable the collection object and click **[Save]**.

Step 4: In the SL1 Integration Service, check the Device Sync configuration

Finally, if you have not found any issues in SL1, check to make sure you have properly configured the values and mappings on the **Configuration** pane for the "Sync Devices from SL1 to ServiceNow" integration application.

In the Integration Service, go to the **Integrations** page, select the "Sync Devices from SL1 to ServiceNow" integration application, and click **Configure** (⚙). The **Configuration** pane appears:


Integration
isadmin
ScienceLogic
Sync Devices from SL1 to ServiceNow
Run Started: Mar 27, 2020 14:02:11
Pull and Process Hardware Models
Pull and Process Manufacturers
Query ServiceNow Hardware Models
Pull ServiceNow CIs
Query ServiceNow Manufacturers
Fetch Devices from SL
Process ServiceNow CIs
Process SL1 Devices
Compare SL1 Devices and ServiceNow CIs
Trigger CI Uploads
Run: success
Step Log
Show JSON Configs
Save

Sync Devices from SL1 to ServiceNow

Modify configuration and save.

Configuration: docs_servicenow_syncpack

sl1_hostname: 10.64.68.17 — ${config.sl1_host}

sl1_db_host: ${config.sl1_host} — ${config.sl1_db_host}

snow_hostname: ven01056.service-now.com — ${config.snow_host}

sl1_user: em7admin — ${config.sl1_user}

snow_user: admin — ${config.snow_user}

sl1_password: ••••••••••••••••••••

snow_password: ••••••••••••••••••••

sl1_db_user: root — ${config.sl1_db_user}

sl1_db_password: ••••••••••••••••••••

region: ScienceLogic — ${config.region}

read_timeout: 20

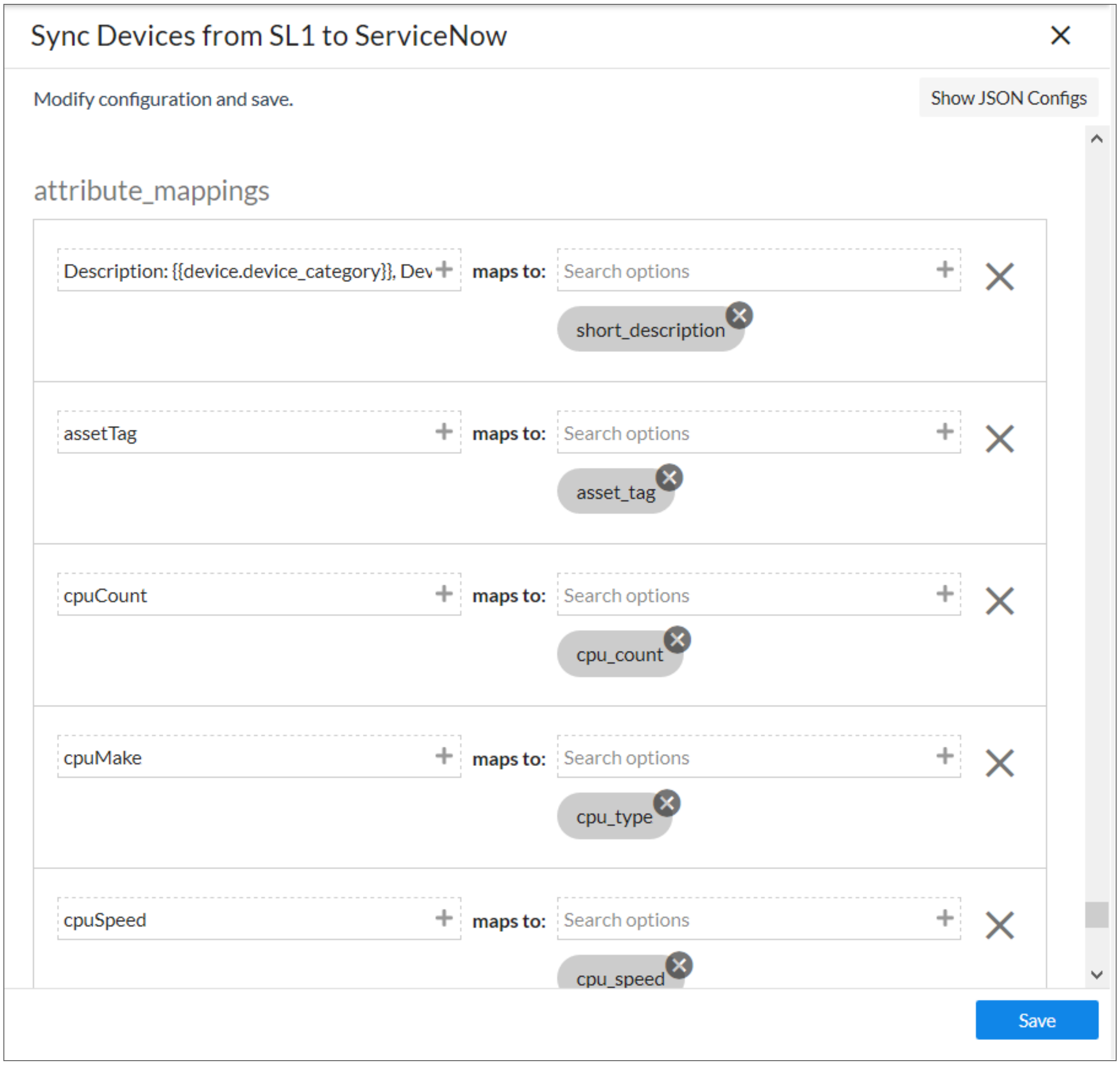
mappings

cmdb_ci_computer **maps to:** Search options

IBM | IBM OS/400 V5R1M0

Microsoft | Windows CE Version 3.0 (Multiple

Scroll down to the **attribute_mappings** section on the **Configuration** page. In this section you can add custom device attribute mappings:



Sync Devices from SL1 to ServiceNow

Modify configuration and save.

attribute_mappings

Description: {{device.device_category}}, Dev maps to: short_description

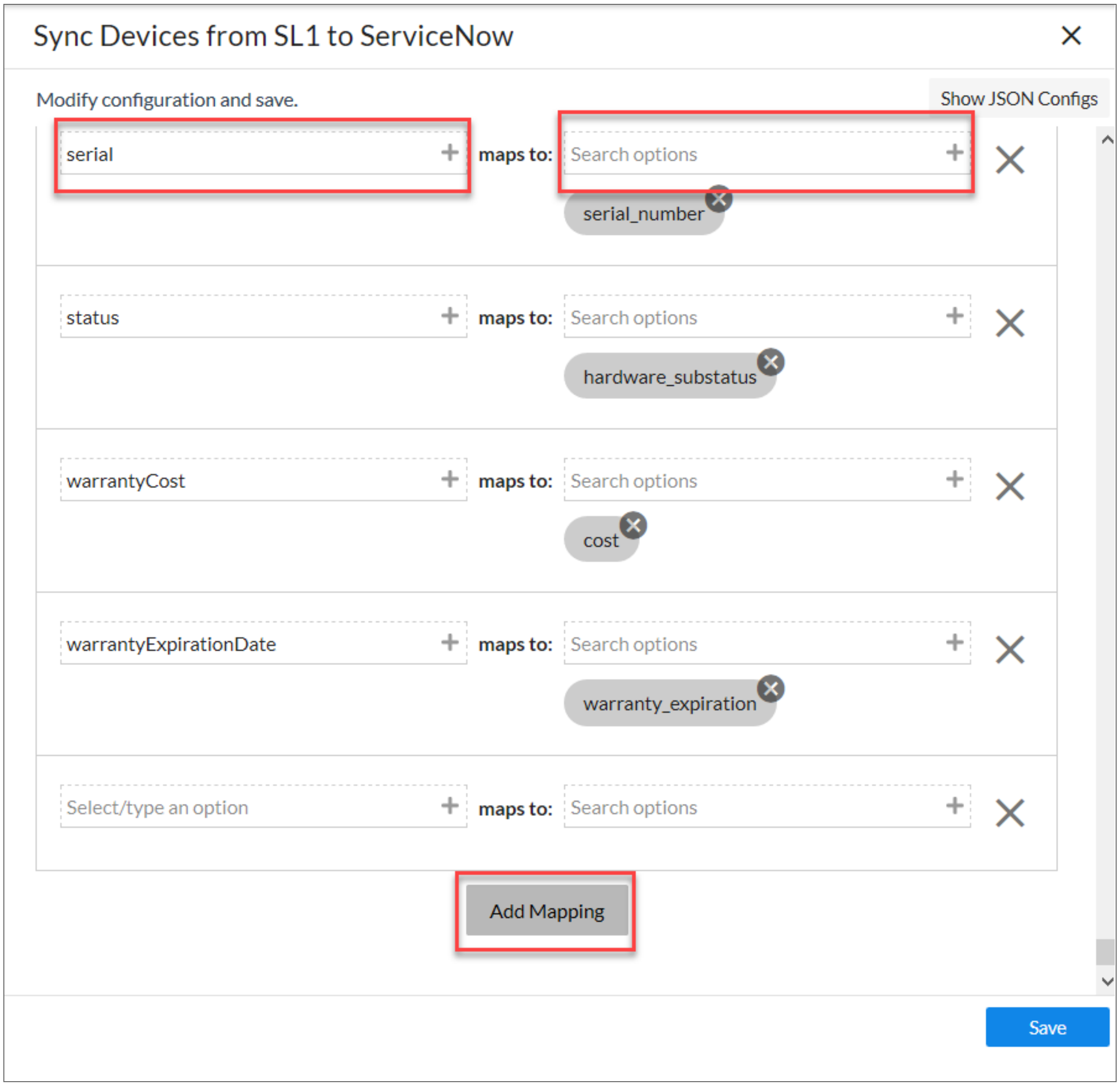
assetTag maps to: asset_tag

cpuCount maps to: cpu_count

cpuMake maps to: cpu_type

cpuSpeed maps to: cpu_speed

If you have not addressed the issue with the missing serial number data in the previous steps, you can click **[Add Mapping]** in this section and create a mapping for Serial Number. The SL1 values are on the left, and the ServiceNow values are on the right:



Sync Devices from SL1 to ServiceNow

Modify configuration and save.

serial maps to: serial_number

status maps to: hardware_substatus

warrantyCost maps to: cost

warrantyExpirationDate maps to: warranty_expiration

Select/type an option maps to: Search options

However, you find that the "serial" value from SL1 is already mapped to the "serial_number" value for ServiceNow, so you do not need to do any additional editing on the **Configuration** pane. The Serial Number value should sync properly the next time you run the "Sync Devices from SL1 to ServiceNow" integration application.

Appendix

A

Checklists for Deployment

Overview

This appendix describes the checklists for deploying the Integration Service and the ServiceNow Synchronization PowerPacks, based on your environment and configuration.

This appendix includes the following topics:

CMDB-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow

1. Deploy the Integration Service ISO:
   - IP address, Netmask, Gateway, DNS, Hostname provided
   - Root password provided (this is the root user for the OS)
   - Start Docker services after installation:

     ```
     /opt/iservices/scripts/pull_start_iservices.sh
     ```

   - Validate that iservices are running:

     ```
     docker service ls
     ```

2. Activate the Configuration Management For Scoped Apps (CMDB) Plugin.
3. Install the ScienceLogic Certified Application and create a ServiceNow group and user account:
   - Username
   - Password
   - Web Service Access Only
   - GMT Time Zone
   - x_sclo_scilogic.Admin role assigned
4. Install the ServiceNow Synchronization PowerPacks on the Integration Service.
5. Create the Integration Service configuration object using the "ServiceNow SyncPack" configuration object as a template.

Sync Devices from SL1 to ServiceNow

1. Align the configuration object to the following integration applications:
   - Cache ServiceNow CIs and SL1 Device Classes
   - Sync Devices from SL1 to ServiceNow
2. Run "Cache ServiceNow CIs and SL1 Device Classes" to retrieve all device class information from SL1 and ServiceNow. This will populate the device class mapping in the following step. This integration should be run at least every 23 hours.
3. Configure class and attribute mappings in "Sync Devices from SL1 to ServiceNow".
4. Run "Generate Required CI Relations for ServiceNow" to see if you are missing any service rules or class mappings and create any required maps, containment rules, and hosting rules.
5. Run "Sync Devices from SL1 to ServiceNow" either manually or on a schedule. This integration should be run at least every 23 hours. See documentation for more information.

Discover Devices from ServiceNow in SL1

1. Align the configuration object to the following integration applications:
   - Sync Discovery Requirements
   - Sync Discovery Session Requests from ServiceNow to SL1
   - Sync Discovery Session Status from SL1 to ServiceNow
2. Set additional configuration variables for each of the integrations applications above in the respective **Configuration** pane.
3. Run "Sync Discovery Requirements" to sync all discovery-dependent information from SL1 to ServiceNow.
4. Make sure that the Discovery request RITM is successfully created and approved in ServiceNow using the provided Service Catalogs.
5. Run "Sync Discovery Session Requests from ServiceNow to SL1" either manually or on a schedule to create and execute the discovery session in SL1.
6. After the discovery session has completed in SL1, run "Sync Discovery Session Status from SL1 to ServiceNow" either manually or on a schedule to update the status of the RITM in ServiceNow.

NOTE: The following integration applications only sync CIs that are aligned with the devices that are already synced with ServiceNow. Before syncing any of the CIs below, you must first sync devices between SL1 and ServiceNow.

Sync Business Services from SL1 to ServiceNow

1. Align the configuration object to the following integration application:
   - Sync Business Services from SL1 to ServiceNow
2. Configure the service classification mappings in "Sync Business Services from SL1 to ServiceNow". These are defined in the **Configuration** pane.
3. Run "Sync Business Services from SL1 to ServiceNow" either manually or on a schedule.

Sync File Systems from SL1 to ServiceNow

1. Align the configuration object to the following integration application:
   - Sync File Systems from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs.
3. Run "Sync File Systems from SL1 to ServiceNow" either manually or on a schedule.

Sync Network Interfaces from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

    - Sync Network Interfaces from SL1 to ServiceNow

2. Determine additional filters for syncing network interfaces using the ***adapter_sync*** variable defined in the **Configuration** pane.
3. The parent CI must be synced in order to see these related CIs.
4. Run "Sync Network Interfaces from SL1 to ServiceNow" either manually or on a schedule. This integration application should be run at least every 23 hours if you would like to sync interface-level relationships with "Sync Advanced Topology from SL1 to ServiceNow".

Sync Installed Software from SL1 to ServiceNow

1. Align the configuration object to the following integration application:
    - Sync Software Packages from SL1 to ServiceNow
    - Sync Installed Software from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs.
3. First, run "Sync Software Packages from SL1 to ServiceNow". Verify that the integration has run successfully.
4. Run "Sync Installed Software from SL1 to ServiceNow" either manually or on a schedule.

Sync Maintenance Schedules from ServiceNow to SL1

1. Align the configuration object to the following integration application:
    - Sync Maintenance Schedules from ServiceNow to SL1
2. The parent CI must be synced in order to see these related CIs. The affected CI must have ***SL1 Monitored*** set to *True*.
3. If needed, configure the ***Request*** and ***Task state*** IDs in the **Configuration** pane. This is needed if the customer has custom IDs for certain change request or change task states.
4. Run "Sync Maintenance Schedules from ServiceNow to SL1" either manually or on a schedule.

Sync Advanced Topology from SL1 to ServiceNow

1. Align the configuration object to the following integration application:
    - Sync Advanced Topology from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs. Syncing Interface-level relationships (Layer 2, LLDP, and CDP) will require that "Sync Network Interfaces from SL1 to ServiceNow" is set to run at least every 23 hours.
3. Run "Sync Advanced Topology from SL1 to ServiceNow" either manually or on a schedule.

CMDB-Only ServiceNow Integration with Single SL1 and Domain-Separated ServiceNow

1. Deploy the Integration Service ISO:
   - IP address, Netmask, Gateway, DNS, Hostname provided
   - Root password provided (this is the root user for the OS)
   - Start Docker services after installation:

     ```
     /opt/iservices/scripts/pull_start_iservices.sh
     ```

   - Validate that iservices are running:

     ```
     docker service ls
     ```

2. Activate the Configuration Management For Scoped Apps (CMDB) Plugin.
3. Install the ScienceLogic Certified Application and create a ServiceNow group and user account:
   - Username
   - Password
   - Web Service Access Only
   - GMT Time Zone
   - x_sclo_scilogic.Admin role assigned
4. Install the ServiceNow Synchronization PowerPacks on the Integration Service.
5. Create the Integration Service configuration object using the "ServiceNow SyncPack" configuration object as a template.

Sync Devices from SL1 to ServiceNow

1. Align the configuration object to following integration applications:
   - Sync Organizations from SL1 to ServiceNow Companies
   - Cache ServiceNow CIs and SL1 Device Classes
   - Sync Devices from SL1 to ServiceNow
2. In "Sync Organizations from SL1 to ServiceNow Companies" **Configuration** pane, set the ***Source_of_Truth*** to *ServiceNow* and set the ***Domain_Separation*** flag to *True*.
3. Run "Sync Organizations from SL1 to ServiceNow Companies".
4. Run "Cache ServiceNow CIs and SL1 Device Classes" to retrieve all device class information from SL1 and ServiceNow. This will populate the device class mapping in the following step. This integration application should be run at least every 23 hours.

5. Configure class and attribute mappings in "Sync Devices from SL1 to ServiceNow". Set ***Domain_ Separation*** to *True*.
6. Run "Generate Required CI Relations for ServiceNow" to see if you are missing any service rules or class mappings, and then create any required maps, containment rules, and hosting rules.
7. Run "Sync Devices from SL1 to ServiceNow" either manually or on a schedule. This integration application should be run at least every 23 hours.

Discover Devices from ServiceNow in SL1

1. Align Configuration Object to following integration applications:
   - Sync Discovery Requirements
   - Sync Discovery Session Requests from ServiceNow to SL1
   - Sync Discovery Session Status from SL1 to ServiceNow
2. In "Sync Discovery Requirements" **Configuration** pane, set the ***Source_of_Truth*** to *ServiceNow*. Set ***Domain_Separation*** to *True*.
3. Run "Sync Discovery Requirements" to sync all discovery-dependent information from SL1 to ServiceNow and back to SL1.
4. Set additional configuration variables for each of the other integration applications above in the respective **Configuration** pane.
5. Discovery request RITM is successfully created and approved in ServiceNow using the provided Service Catalogs.
6. Run "Sync Discovery Session Requests from ServiceNow to SL1" either manually or on a schedule to create and execute the discovery session in SL1.
7. After the discovery session completes in SL1, run "Sync Discovery Session Status from SL1 to ServiceNow" either manually or on a schedule to update the status of the RITM in ServiceNow.

NOTE: Domain separation requires that "Sync Organizations from SL1 to ServiceNow Companies" is configured with ***Domain_Separation*** enabled and ***Source_of_Truth*** set to *ServiceNow*. In a domain-separated ServiceNow environment, this integration application must be properly configured and run successfully before syncing any additional CI items. SL1 organizations that are linked to a ServiceNow company will have the ***crm_id*** populated with the ServiceNow Company ***sys_id***.

NOTE: The following integrations only sync CIs that are aligned with the devices that are already synced with ServiceNow. Before syncing any of the CIs below, you must first sync devices between SL1 and ServiceNow.

Sync Business Services from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

- Sync Business Services from SL1 to ServiceNow

2. Configure the service classification mappings in "Sync Business Services from SL1 to ServiceNow". These are defined in the **Configuration** pane.
3. Ensure that ***Domain_Separation*** is set to *True* in the **Configuration** pane.
4. Run "Sync Business Services from SL1 to ServiceNow" either manually or on a schedule.

Sync File Systems from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

    - Sync File Systems from SL1 to ServiceNow

2. Ensure that ***Domain_Separation*** is set to *True* in the **Configuration** pane.
3. The parent CI must be synced in order to see these related CIs.
4. Run "Sync File Systems from SL1 to ServiceNow" either manually or on a schedule.

Sync Network Interfaces from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

    - Sync Network Interfaces from SL1 to ServiceNow

2. Determine additional filters for syncing network interfaces using the ***adapter_sync*** variable defined in the **Configuration** pane.
3. Ensure that ***Domain_Separation*** is set to *True* in the **Configuration** pane.
4. The parent CI must be synced in order to see these related CIs.
5. Run "Sync Network Interfaces from SL1 to ServiceNow" either manually or on a schedule. This integration application should be run at least every 23 hours if you would like to sync interface-level relationships with "Sync Advanced Topology from SL1 to ServiceNow".

Sync Installed Software from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

    - Sync Software Packages from SL1 to ServiceNow
    - Sync Installed Software from SL1 to ServiceNow

2. The parent CI must be synced in order to see these related CIs.
3. First, run "Sync Software Packages from SL1 to ServiceNow". Verify that the integration has run successfully.
4. Run "Sync Installed Software from SL1 to ServiceNow" either manually or on a schedule.

Sync Maintenance Schedules from ServiceNow to SL1

1. Align the configuration object to the following integration application:

- Sync Maintenance Schedules from ServiceNow to SL1

2. The parent CI must be synced in order to see these related CIs. The affected CI must have ***SL1 Monitored*** set to *True*.
3. If needed, configure the ***Request*** and ***Task state*** IDs in the **Configuration** pane. This is needed if the customer has custom IDs for certain change request or change task states.
4. Run "Sync Maintenance Schedules from ServiceNow to SL1" either manually or on a schedule.

Sync Advanced Topology from SL1 to ServiceNow

1. Align the configuration object to the following integration application:

    - Sync Advanced Topology from SL1 to ServiceNow

2. Ensure that ***Domain_Separation*** is set to *True* in the **Configuration** pane.
3. The parent CI must be synced in order to see these related CIs. Syncing Interface-level relationships (Layer 2, LLDP, and CDP) will require that "Sync Network Interfaces from SL1 to ServiceNow" is set to run at least every 23 hours.
4. Run "Sync Advanced Topology from SL1 to ServiceNow" either manually or on a schedule.

CMDB-Only ServiceNow Integration with Multiple SL1 Systems, no Domain-Separated ServiceNow

NOTE: Depending on the size of your SL1 stacks and the number of SL1 stacks you have, you may need to consider a “multi-tenant” configured IS. This is a more advanced deployment model. Please contact a ScienceLogic representative for more information.

1. Deploy the Integration Service ISO:
   - IP address, Netmask, Gateway, DNS, Hostname provided
   - Root password provided (this is the root user for the OS)
   - Start Docker services after installation:

     ```
     /opt/iservices/scripts/pull_start_iservices.sh
     ```

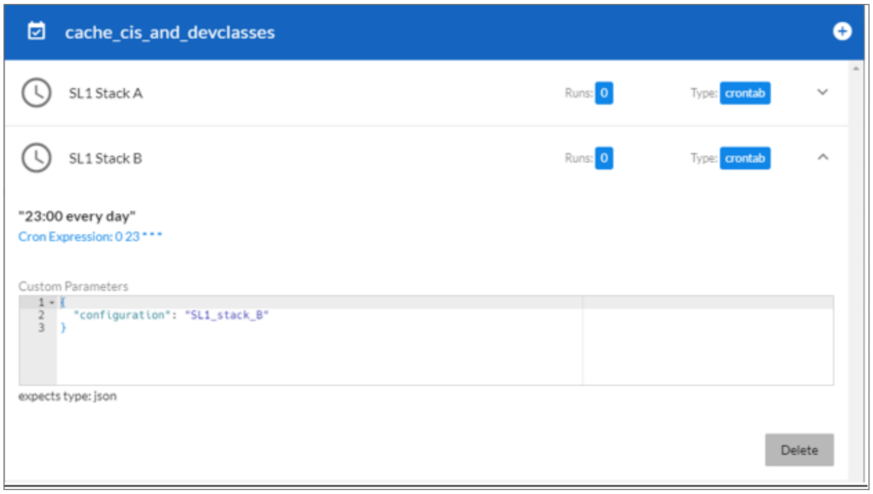
   - Validate that iservices are running:

     ```
     docker service ls
     ```

2. Activate the Configuration Management For Scoped Apps (CMDB) Plugin.
3. Install the ScienceLogic Certified Application and create a ServiceNow group and user account:
   - Username
   - Password
   - Web Service Access Only
   - GMT Time Zone
   - x_sclo_scilogic.Admin role assigned
4. Install the ServiceNow Synchronization PowerPacks on the Integration Service.
5. Create the Integration Service configuration object using the "ServiceNow SyncPack" configuration object as a template.

NOTE: The key difference between integrating a single SL1 stack on the Integration Service and integrating multiple SL1 stacks on the Integration Service is how you run the integration application. Running the integration application with multiple SL1 stacks involves creating an individual configuration object for each SL1 stack. Then, create an individual schedule for each configuration object. Each schedule should use a configuration object that is specific to a single SL1 stack.

When creating the schedule, populate the custom parameters with the configuration object ID. For example:



"23:00 every day"

Cron Expression: 0 23 * * *

Custom Parameters

```
{
  "configuration": "SL1_stack_B"
}
```

expects type: json

Sync Devices from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following Integrations following the note above:
   - Cache ServiceNow CIs and SL1 Device Classes
   - Sync Devices from SL1 to ServiceNow
2. Wait for the "Cache ServiceNow CIs and SL1 Device Classes" to retrieve all device class information from SL1 and ServiceNow. This will populate the device class mapping in the following step. This integration should be run at least every 23 hours.
3. Configure class and attribute mappings in "Sync Devices from SL1 to ServiceNow". If each SL1 stack has different class and attribute mapping requirements, you will need to specify the mappings in each schedule's custom parameters for this application.
4. Run "Generate Required CI Relations for ServiceNow" to see if you are missing any service rules or class mappings and create any required maps, containment rules, and hosting rules. By default, these will pull from the mappings set in "Sync Devices from SL1 to ServiceNow". If you have defined different mappings for each SL1 stack, you will need to also specify these mappings in each schedule's custom parameters for this application.
5. Run "Sync Devices from SL1 to ServiceNow" with a schedule where each schedule uses the configuration object for a SL1 stack. This integration application should be run at least every 23 hours.

Discover Devices from ServiceNow in SL1

1. Create a schedule for each SL1 stack for the following integration applications:
   - Sync Discovery Requirements
   - Sync Discovery Session Requests from ServiceNow to SL1

- Sync Discovery Session Status from SL1 to ServiceNow

2. If any additional configuration variables are needed, these will need to be set in the custom parameters section for each schedule
3. Run "Sync Discovery Requirements" on a schedule to sync all discovery-dependent information from SL1 to ServiceNow.
4. Verify that the discovery request RITM is successfully created and approved in ServiceNow using the provided Service Catalogs.
5. Run "Sync Discovery Session Requests from ServiceNow to SL1"on a schedule where each schedule uses the configuration object for a SL1 stack to create and execute the discovery session in SL1.
6. Once the discovery session has completed in SL1, run "Sync Discovery Session Status from SL1 to ServiceNow" via schedule where each schedule uses the configuration object for a SL1 stack to update the status of the RITM in ServiceNow.

The following integration applications only sync CIs that are aligned with the devices that are already synced with ServiceNow. Before syncing any of the CIs below, you must first sync devices between SL1 and ServiceNow.

Sync Business Services from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync Business Services from SL1 to ServiceNow
2. Configure service classification mappings in "Sync Business Services from SL1 to ServiceNow" application. These are defined in the **Configuration** pane.
3. Run "Sync Business Services from SL1 to ServiceNow" on a schedule where each schedule uses the configuration object for a SL1 stack.

Sync File Systems from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync File Systems from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs.
3. Run "Sync File Systems from SL1 to ServiceNow" on a schedule where each schedule uses the configuration object for a SL1 stack.

Sync Network Interfaces from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync Network Interfaces from SL1 to ServiceNow

2. Determine additional filters for syncing network interfaces using the ***adapter_sync*** variable defined in the **Configuration** pane.
3. The parent CI must be synced in order to see these related CIs.
4. Run "Sync Network Interfaces from SL1 to ServiceNow" on a schedule where each schedule uses the configuration object for a SL1 stack. This integration application should be run at least every 23 hours if you would like to sync interface-level relationships with "Sync Advanced Topology from SL1 to ServiceNow".

Sync Installed Software from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync Software Packages from SL1 to ServiceNow
   - Sync Installed Software from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs.
3. "Sync Software Packages from SL1 to ServiceNow" must run on a schedule before "Sync Installed Software from SL1 to ServiceNow". Verify that the software packages have been synced before continuing.
4. After the software packages have been synced, run "Sync Installed Software from SL1 to ServiceNow" on a schedule where each schedule uses the configuration object for a SL1 stack.

Sync Maintenance Schedules from ServiceNow to SL1

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync Maintenance Schedules from ServiceNow to SL1
2. The parent CI must be synced in order to see these related CIs. The affected CI must have the ***SL1 Monitored*** field set to *True*.
3. If needed, configure the ***Request*** and ***Task state*** IDs in the **Configuration** pane. This is needed if the customer has custom IDs for certain change request or change task states.
4. Run "Sync Maintenance Schedules from ServiceNow to SL1" on a schedule where each schedule uses the configuration object for a SL1 stack.

Sync Advanced Topology from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following integration application:
   - Sync Advanced Topology from SL1 to ServiceNow
2. The parent CI must be synced in order to see these related CIs. Syncing Interface-level relationships (Layer 2, LLDP, and CDP) will require that you set "Sync Network Interfaces from SL1 to ServiceNow" to run at least every 23 hours.
3. Run "Sync Advanced Topology from SL1 to ServiceNow" on a schedule where each schedule uses the configuration object for a SL1 stack.

Incident-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow

1. Deploy the Integration Service ISO:
    - IP address, Netmask, Gateway, DNS, Hostname provided
    - Root password provided (this is the root user for the OS)
    - Start Docker services after installation:

      ```
      /opt/iservices/scripts/pull_start_iservices.sh
      ```

    - Validate that iservices are running:

      ```
      docker service ls
      ```

2. Install the ScienceLogic Certified Application and create a ServiceNow group and user account:
    - Username
    - Password
    - Web Service Access Only
    - GMT Time Zone
3. Install the ServiceNow Synchronization PowerPacks on the Integration Service.
4. Create the Integration Service configuration object using the "ServiceNow SyncPack" configuration object as a template:
    - Align the configuration object to the following integration applications:
        - Create or Update ServiceNow Incident from SL1 Event
        - Update ServiceNow Incident when SL1 Event is Acknowledged
        - Update ServiceNow Incident when SL1 Event is Cleared
        - Sync Incident State from ServiceNow to SL1 Event
    - Only the following Integration should be run manually or scheduled:
        - Sync Incident State from ServiceNow to SL1 Event
5. Install the ServiceNow Base PowerPack and configure SL1:
    - Use the "ServiceNow RBA – Example" credential as a template to create a new credential that points to the Integration Service instance.
    - Align the newly-created credential to the "ServiceNow – Add/Update/Clear Incident" Run Book Action.
    - Ensure that all Run Book Actions and Run Book Policies are enabled.

Appendix

B

Certified Application Objects

Overview

This appendix describes the tables, endpoints, and roles that were created in ServiceNow as part of the "ScienceLogic SL1: CMDB & Incident Automation" application. This application is also known as the "Certified Application" or the "Scoped Application".

This appendix includes the following topics:

Roles

Two Roles were added with the ScienceLogic update set, Admin (x_sclo_scilogic.Admin) and User (x_sclo_scilogic.User). Both give access to SL1.

Role	Inherited Roles	Other Inherited Roles	Role Definition
x_sclo_scilogic.Admin			Role for ScienceLogic Service Accounts.
	itil		Can perform standard actions for an ITIL help desk technician. This is the default "Technician" role. Can open, update, close incidents, problems, changes, config management items. By default, only users with the itil role can have tasks assigned to them
		Dependency_view	A special role to be applied both on the $ngbsm UI page and on the BSMProcessor. This role is required to access the dependency views module. By default, ITIL includes this role to avoid regressions.
		cmdb_query_builder	Can access the CMDB Query Builder application to create, run, and save queries on the CMDB.
		template_editor	
		view_changer	Can switch active views.
		app_service_user	Can view and retrieve information using API from application service maps (cmdb_ci_service_discovered).
		certification	Can work on Certification tasks.
	import_transformer		Can manage Import Set Transform Maps and run transforms.
x_sclo_scilogic.User			General user account that allows read-only access to SL1.

Tables

Name	Label	Extends	Comments
x_sclo_scilogic_event	Event	(empty)	Event information
x_sclo_scilogic_event_severity	Event Severity Look Rules	Data Lookup Matcher Rules	Look up table for event Severity
x_sclo_scilogic_incident	Import Incident	Import Set Row	Import / staging events before transform to Event and Incident
x_sclo_scilogic_import_installed_software	Import Installed Software	Import Set Row	Import / staging events before transform to Software Instance
x_sclo_scilogic_org_ven_mfg	Import ORG VEN MFG	Import Set Row	Import / staging events before transform to core_company
x_sclo_scilogic_import_discovery_dependent	Import Discovery Dependent	Import Set Row	Import / staging events before transform to Discovery Dependent table
x_sclo_scilogic_discovery_dependent	Discovery Dependent	(empty)	Discovery Dependent Information
x_sclo_scilogic_catalog_item_templates	Catalog item Templates	(empty)	Templates use to fill out catalog items
x_sclo_scilogic_import_service_request	Import Service Request	Import Set Row	Import / staging events before transform to Service Requests

Table Columns (cmdb_ci)

Name	Label	Type	Comments
x_sclo_scilogic_id	SL1 ID	Integer	Unique ID
x_sclo_scilogic_region	SL1 Region	String	Unique String of SL1 Platform
x_sclo_scilogic_url	SL1 URL	URL	URL to SL1 Platform
x_sclo_scilogic_monitored	SL1 Monitored	True/False	Device currently synced with SL1 Platform

Table Columns (core_company)

Name	Label	Type	Comments
x_sclo_scilogic_id	SL1 ID	String	Unique ID
x_sclo_scilogic_region	SL1 Region	String	Unique String of SL1 Platform
x_sclo_scilogic_monitored	SL1 Monitored	True/False	Organization currently synced with SL1 Platform

Table Columns (cmdb_group)

Name	Label	Type	Comments
x_sclo_scilogic_id	SL1 ID	String	Unique ID
x_sclo_scilogic_region	SL1 Region	String	Unique String of SL1 Platform

Script Includes

Name	API Name	Comments
CatalogUtils	x_sclo_scilogic.catalogUtils	Catalog Script include scripts
ChangeUtils	x_sclo_scilogic.changeUtils	Change Script include scripts
DeviceUtils	x_sclo_scilogic.DeviceUtils	Device Script include scripts
EventUtils	x_sclo_scilogic.EventUtils	Event Script include scripts
GeneralUtils	x_sclo_scilogic.GeneralUtils	General Script include scripts

Event Registry

Suffix	Event name	Table	Comments
device_monitoring	x_sclo_scilogic.device_monitoring	Configuration Item [cmdb_ci]	Event for Device Monitoring
Remove_monitoring	x_sclo_scilogic.remove_monitoring	Configuration Item [cmdb_ci]	Event for Remove Monitoring

Scripted Actions

Name	Event name	Comments
Device Monitoring Catalog item	x_sclo_scilogic.device_monitoring	Action used to submit Catalog item via Event.
Device Removal Catalog item	x_sclo_scilogic.remove_monitoring	Action used to submit Catalog item via Event.

Data Lookup Definitions

Name	Source Table	Matcher Table	Comments
Event Severity	Import Incident [x_sclo_scilogic_incident]	Event Severity Lookup Rules [x_sclo_scilogic_event_severity]	Lookup for ScienceLogic Severity to Impact and Urgency

System Properties

Suffix	Name	Comments
CatalogItemDiscovery	x_sclo_scilogic.CatalogItemDiscovery	Unique value (sys_id)
CatalogItemRemove	x_sclo_scilogic.CatalogItemRemove	Unique value (sys_id)
closeCode	x_sclo_scilogic.closeCode	Value to use for Close Code for Incident Transform
Contact type	x_sclo_scilogic.Contact Type	Value to use for Contact type for Incident Transform
deviceLogging	x_sclo_scilogic.deviceLogging	Turn on Logging
deviceLoggingParam	x_sclo_scilogic.deviceLoggingParm	Add additional parameters beyond the default errors
discoverySource	x_sclo_scilogic.discoverySource	Discovery Source to be used by Integration Service
notResolved	x_sclo_scilogic.notResolved	Value of Reopened Incident
stateNew	x_sclo_scilogic.stateNew	Value of New Incident
StateResolved	x_sclo_scilogic.stateResolved	Value of Resolved Incident

Catalog Item

Name	Comments
Device Discovery	Role for ScienceLogic Service Accounts.
Monitoring Removal	General user account that allows read only access to ScienceLogic Application.

Catalog UI Policies

Catalog item	Short description	Comments
Device Discovery	Catalog Template	Updates form based on Select template
Device Discovery	Create Virtual Device	Updates form based on Request type
Device Discovery	Create Virtual Device (Retired)	
Device Discovery	Device Discovery	Updates form based on Request type
Device Discovery	Device Discovery (Retired)	
Monitoring Removal	Hide Overview variables not required	Hide variables not required for the Monitoring Removal request
Device Discovery	Port Scan	Hide scan ports that are not default
Device Discovery	Port Scan (Retired)	
Device Discovery	Region	Updates form based on Organization
Device Discovery	Region (Retired)	
Monitoring Removal	Region via Organization	Updates form based on Organization
Device Discovery	Save as Template	Updates form based on Save as template

Variable Sets

Title	Internal name	Comments
Create_virtual_device	create_virtual_device	
Discovery Overview	discovery_overview	

Title	Internal name	Comments
Discovery Sesion - Basic Settings	discovery_sesion_basic_settings	
Discovery Session - Detection and Scanning	discovery_session_detection_and_scanning	
Discovery Session - IP & Credentials	discovery_session_ip_credentials	
Monitoring Removal	monitoring _removal	
Service Catalog item Template	service_catalog_item_template	

Catalog Client Scripts

Name	Catalog item	Type	Comments
Hide Request Type Options	Monitoring Removal	onLoad	Shared variable hide options that don't apply
Hide Request Type Options	Device Discovery	onLoad	Shared variable hide options that don't apply
Region	Monitoring Removal	onChange	Update Region field based on Company Region
Region	Monitoring Removal	onChange	Update Region field based on Company Region

Workflows

Name	Table	Comments
SL1 Monitoring Removal	Requested Item [sc_req_item]	Workflow for Removal of devices from SL1 process
SL1 Discovery Session	Requested Item [sc_req_item]	Workflow for Discovery session process

Scripted REST Resources

Name			Comments
Business Services	/api/x_sclo_ scilogic/v1/sciencelogic/business_service	GET	This GET api will pull all ScienceLogic monitored Configuration items specific to Business Services class from the CMDB. It will be ordered via the sys_id field to ensure the same order every time.
CMDB Group	/api/x_sclo_scilogic/v1/sciencelogic/cmdb_ group	POST	Use this API to create cmdb_groups & add a CI to them.
Change Requests	/api/x_sclo_ scilogic/v1/sciencelogic/change_requests	GET	This GET api will pull Active Change Requests or Change Tasks based on the record_type supplied that have ScienceLogic monitored CI attached. It will be ordered via the sys_id field to ensure the same order every time.
Classification	/api/x_sclo_ scilogic/v1/sciencelogic/classification	GET	This GET api will pull all required CMDB information to build JSON payloads.
Companies	/api/x_sclo_ scilogic/v1/sciencelogic/companies	GET	This GET api will pull all Active Companies that are ScienceLogic monitored. It will be ordered via the sys_id field to ensure the same order every time.
Configuration Items	/api/x_sclo_ scilogic/v1/sciencelogic/configuration_Items	GET	This GET api will pull all ScienceLogic monitored Configuration items from the CMDB. It will be ordered via the sys_id field to ensure the same order every time.
Device IdentificationEngine	/api/x_sclo_ scilogic/v1/sciencelogic/IdentificationEngine	POST	Use this API to create or update configuration items within the CMDB via ScienceLogic.

Name			Comments
File Systems	/api/x_sclo_scilogic/v1/sciencelogic/file_systems	GET	This GET api will pull all ScienceLogic monitored Configuration items specific to File systems class from the CMDB. It will be ordered via the sys_id field to ensure the same order every time.
Import Set	/api/x_sclo_scilogic/v1/sciencelogic/import_set	POST	This POST API will post to the target import set table and create a record for each cmdb_ci.
Incidents	/api/x_sclo_scilogic/v1/sciencelogic/incidents	GET	This GET api will pull all incidents. It will be ordered via the sys_id field to ensure the same order every time.
Installed Software	/api/x_sclo_scilogic/v1/sciencelogic/installed_software	GET	This GET api will pull all Servicenow Software packages and installed instances from the CMDB. It will be ordered via the sys_id field to ensure the same order every time.
Manufacture	/api/x_sclo_scilogic/v1/sciencelogic/manufactures	POST	This POST API will pull all Manufactures.
Model	/api/x_sclo_scilogic/v1/sciencelogic/models	POST	This POST API will pull all Model.
Network Adapters	/api/x_sclo_scilogic/v1/sciencelogic/network_adapters	GET	This GET api will pull all ScienceLogic monitored Configuration items specific to Network Adapter class from the CMDB. It will be ordered via the sys_id field to ensure the same order every time.
Service Request	/api/x_sclo_scilogic/v1/sciencelogic/service_request	GET	This GET api will pull all ServiceRequest items from the CMDB associated with Device Discovery Catalog item. It will be ordered via the sys_id field to ensure the same order every time.
Classification	/api/x_sclo_scilogic/v2/sciencelogic/classification	GET	This GET api will pull all required CMDB information to build JSON payloads.

Transform Maps

Name	Source Table	Target Table	Comments
ScienceLogic Discovery Dependent	Import Discovery Dependent	Discovery Dependent	Import / staging table for Catalog Dependents
ScienceLogic Event	Import Incident	Event	Import / staging table for Events.
ScienceLogic Incident	Import Incident	Incident [incident]	Import / staging table for Incident
ScienceLogic Organization	Import ORG VEN MFG []	Company [core_company]	Import / staging table for Organization
ScienceLogic Service Request	Import Service Request []	Request Item [sc_req_item]	Import / staging table for Request item

Transform Scripts

Name	Transform Map	Order	Comments
onBefore	ScienceLogic Event	100	Check Action
onAfter	ScienceLogic Event	100	Check Action; Get Resolved Validation script include
onBefore	ScienceLogic Incident	100	Check Action, event workflow script include
onAfter	ScienceLogic Incident	100	Check Action, Affected CI script include

Appendix C

ServiceNow API Endpoints

Overview

This appendix describes the customized ServiceNow API Endpoints that were created for the Integration Service ServiceNow Synchronization PowerPacks. These scripted endpoints reduce the amount of REST calls that the Integration Service makes to ServiceNow.

Please note that for pagination, the following Query parameters are not required: `sysparm_offset` and `sysparm_limit`. The default settings are:

- `sysparm_offset=0`
- `sysparm_limit` = ServiceNow defines the default upper limits for data export. It will check the following properties at *System Properties* > *Import Export*: `glide.json.export.limit`, `glide.ui.export.limit`, and then `glide.ui.export.war.threshold`.

For example, if you have 200 total records and you want to pull the records in 100-record chunks, then the first pull would be `sysparm_offset=0 & sysparm_limit=100` and the second pull would be `sysparm_offset=100 & sysparm_limit=100`. For more information, see the ServiceNow documentation for Export Limits.

This appendix includes the following topics:

Business Services

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/business_service
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/business_service
```

This operation pulls all the fields from just the Business Service (**cmdb_ci_service**) table. The return is ordered by **sys_id**, so the results display in the same order every time. The results are filtered by the ***SL1 monitored*** and ***SL1 ID*** field on the ServiceNow side. This operation requires the region to be supplied by the requester, and it will only return region-supplied configuration items.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/ sciencelogic/business_
service
```

Example (Response)

```
{
    "results": [
        {
            "operational_status": "1",
            "sys_updated_on": "2019-02-06 19:32:34",
            "discovery_source": "Other Automated",
            "first_discovered": "2019-02-06 19:31:19",
            "sys_updated_by": "admin",
            "sys_created_on": "2019-02-06 19:31:19",
            "sys_domain": "global",
            "used_for": "Production",
            "sys_created_by": "is4user1",
            "sys_domain_path": "/",
            "install_status": "1",
            "name": "One Service to rule them",
            "subcategory": "Service",
            "busines_criticality": "1 - most critical",
            "last_discovered": "2019-02-06 19:31:19",
            "sys_class_name": "cmdb_ci_service",
            "sys_id": "52da95dcdb6323009f7dd7a0cf961918",
            "sys_class_path": "/!!/#C",
            "comments": "Postman",
            "sys_mod_count": "1",
            "x_sclo_scilogic_id": "1570",
            "model_id": "e8aaeb3f3763100044e0bfc8bcbe5d20",
            "cost_cc": "USD",
            "x_sclo_scilogic_monitored": "true",
            "category": "Business Service",
            "service_classification": "Technical Service",
            "x_sclo_scilogic_region": "ScienceLogic"
        }
    ],
    "sysparm_offset": 0,
    "sysparm_limit": 100,
    "return_count": 1,
    "total_count": 1
}
```

Change Requests

HTTP Method

GET

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/change_requests?record_type=change_
request&state=1&region=ScienceLogic
```

Default Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/change_requests?record_type=change_
request&state=1&region=ScienceLogic
```

This scripted API was built for pulling Change Requests or Change Tasks and formatting a JSON object response with the required information to create a maintenance schedule in SL1. The GET queries the **task_ci** table to find configuration items that are monitored by SL1 and are the correct record type. The GET operation returns all records with their configuration items in formatted JSON strings that include planned start and end time.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
record_type (required)	change_request
state	-5
region (required)	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status

Code	Value
200	OK
400	Query parameter \'region\' is not defined and is required.

Fixed Internal Query

State:

```
‘task.sys_class_name=’ + recordType + ‘task.state=’ + state + ‘^ci_item.x_sclo_
scilogic_monitored=true^ci_item.x_sclo_scilogic_region=’ + region
```

Non-State:

```
‘task.active=true^task.sys_class_name=’ + recordType + ‘ci_item.x_sclo_scilogic_
monitored=true^ci_item.x_sclo_scilogic_region=’ + region
```

Example

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/change_
requests?record_type=change_request&state=-5&region=ScienceLogic
```

Example (Response)

```
{
    "results": [
        {
            "sys_id": "48ebaba0db962f00dc44f00fbf961961",
            "number": "CHG0030001",
            "state_value": "-5",
            "state": "New",
            "short_description": "Test Change",
            "planned_start_date": "2019-01-01 06:00:01",
            "planned_end_date": "2019-01-01 18:00:01",
            "device": [
                {
                    "sys_id": "d83dac0adb4dab00dc44f00fbf961919",
                    "name": "Postman Test Server 11",
                    "id": "11",
                    "region": "ScienceLogic"
                }
            ]
        }
    ]
}
```

Classification version 1

NOTE: This API Endpoint has been deprecated. The last version of the "ScienceLogic SL1: CMDB & Incident Automation" application" (also called the Certified or Scoped application), that used this endpoint was version 1.0.18.

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/classification
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/classification
```

To support the identification and reconciliation framework, SL1 requires a large amount of information to know how to correctly fill out the JSON formatted string defined by the Identification Engine documentation. This operation uses the **getTableExtension()** function to find all the tables extended from the **cmdb_ci** table and then goes through each table one by one. This operation collects information about each class, such as which fields are required to identify and if it is considers another class to help find uniqueness. This operation then finds all the associated metadata. Finally, the operation pulls a list of all field names from the table. By default the **criterion_attributes** and **attributes** are not included and require "action=attributes" as a parameter in the API call to be passed.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
action	attributes

Attributes require x_sclo_scilogic.Admin be added to **sys_dictionary.*** (read) ACL to allow the API to access field names on each class table.

Fixed Internal Query

Example

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/sciencelogic/
classification
```

Example (Response)

```
{
  "results": [
    {
      "class_label": "Storage Area Network",
      "class_table": "cmdb_ci_san",
      "criterion_attributes": [
        ""
      ],
      "independent": "false",
      "containment_rule": [

      ],
      "hosting_rule": [

      ],
      "reference_rule": [

      ],
      "attributes": [
        "asset",
        "asset_tag",
        "assigned",
        "assigned_to",
        "assignment_group",
        "attributes",
        "can_print",
        "category",
        "change_control",
        "checked_in",
        "checked_out",
        "comments",
        "company",
        "correlation_id",
        "cost",
        "cost_cc",
        "cost_center",
        "delivery_date",
        "department",
        "discovery_source",
        "dns_domain",
        "due",
        "due_in",
        "fault_count",
        "first_discovered",
        "fqdn",
        "gl_account",
        "install_date",
        "install_status",
        "invoice_number",
        "ip_address",
        "justification",
        "last_discovered",
        "lease_id",
        "location",
        "mac_address",
        "maintenance_schedule",
        "managed_by",
        "manufacturer",
        "model_id",
        "model_number",
        "monitor",
        "name",
        "operational_status",
        "order_date",
        "owned_by",
        "po_number",
        "purchase_date",
        "san_id",
        "schedule",
        "serial_number",
        "short_description",
        "skip_sync",
        "start_date",
        "subcategory",
        "supported_by",
        "support_group",
        "sys_class_name",
        "sys_class_path",
        "sys_created_by",
        "sys_created_on",
        "sys_domain",
        "sys_domain_path",
        "sys_id",
        "sys_mod_count",
        "sys_updated_by",
        "sys_updated_on",
        "unverified",
        "vendor",
        "warranty_expiration",
        "x_sclo_scilogic_id",
        "x_sclo_scilogic_monitored",
        "x_sclo_scilogic_region",
        "x_sclo_scilogic_url"
      ]
    }
  ]
}
```

Classification version 2

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v2/sciencelogic/classification
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/classification
```

To support the identification and reconciliation framework, SL1 requires a large amount of information to know how to correctly fill out the JSON formatted string defined by the Identification Engine documentation. This operation uses the **getTableExtension()** function to find all the tables extended from the **cmdb_ci** table and then goes through each table one by one. This operation collects information about each class, such as which fields are required to identify and if it is considers another class to help find uniqueness. This operation then finds all the associated metadata. Finally, the operation pulls a list of all field names from the table. By default the **criterion_attributes** and **attributes** are not included and require "action=attributes" as a parameter in the API call to be passed.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
action	attributes
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

Attributes require x_sclo_scilogic.Admin be added to **sys_dictionary.*** (read) ACL to allow the API to access field names on each class table.

Fixed Internal Query

Example

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v2/sciencelogic/
classification
```

Example (Response)

```
{
  "results": [
    {
      "class_label": "Storage Area Network",
      "class_table": "cmdb_ci_san",
      "criterion_attributes": [
        ""
      ],
      "independent": "false",
      "containment_rule": [

      ],
      "hosting_rule": [

      ],
      "reference_rule": [

      ],
      "attributes": [
        "asset",
        "asset_tag",
        "assigned",
        "assigned_to",
        "assignment_group",
        "attributes",
        "can_print",
        "category",
        "change_control",
        "checked_in",
        "checked_out",
        "comments",
        "company",
        "correlation_id",
        "cost",
        "cost_cc",
        "cost_center",
        "delivery_date",
        "department",
        "discovery_source",
        "dns_domain",
        "due",
        "due_in",
        "fault_count",
        "first_discovered",
        "fqdn",
        "gl_account",
        "install_date",
        "install_status",
        "invoice_number",
        "ip_address",
        "justification",
        "last_discovered",
        "lease_id",
        "location",
        "mac_address",
        "maintenance_schedule",
        "managed_by",
        "manufacturer",
        "model_id",
        "model_number",
        "monitor",
        "name",
        "operational_status",
        "order_date",
        "owned_by",
        "po_number",
        "purchase_date",
        "san_id",
        "schedule",
        "serial_number",
        "short_description",
        "skip_sync",
        "start_date",
        "subcategory",
        "supported_by",
        "support_group",
        "sys_class_name",
        "sys_class_path",
        "sys_created_by",
        "sys_created_on",
        "sys_domain",
        "sys_domain_path",
        "sys_id",
        "sys_mod_count",
        "sys_updated_by",
        "sys_updated_on",
        "unverified",
        "vendor",
        "warranty_expiration",
        "x_sclo_scilogic_id",
        "x_sclo_scilogic_monitored",
        "x_sclo_scilogic_region",
        "x_sclo_scilogic_url"
      ]
    }
  ]
}
```

CMDB Group

HTTP Method

POST

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/cmdb_group
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/cmdb_group
```

This operation handles the intake of groups of devices from SL1 and converts the device groups to CMDB groups. This operation uses a standard formatted JSON string, and it checks for a ***sys_id*** of the group first by searching for a matching group. This process creates a group if a group is not supplied or found, and then it passes the JSON object to the ServiceNow CMDBGroupAPI, which sets the manual CI list of the group.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Example (Request URL

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/cmdb_group
```

Example (Body)

```
[
  {
    "items": [
      {
        "name": "test",
        "description": "",
        "group": "",
        "manualCIList": "d83dac0adb4dab00dc44f00fbf961919,2e6b7046db8dab00dc44f00fbf
        961929,7fb39667dba12380dc44f00fbf961901,77b39667dba12380dc44f00fbf961917,7bb
        39667dba12380dc44f00fbf96191c",
        "region": "Cisco",
        "id": "1"
      }
    ]
  }
]
```

Example (Response)

```
{
  "result": [
    {
      "idList": [

      ],
      "partialCIListDueToACLFlag": false,
      "nextBatchStart": 0,
      "result": true
    },
    {
      "idList": [

      ],
      "partialCIListDueToACLFlag": false,
      "nextBatchStart": 0,
      "result": true
    }
  ]
}
```

Companies

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/companies
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/companies
```

This operation supports Domain Separation enabled or not enabled. This operation pulls all the fields for from the company table that are not NULL values. The return is ordered by ***sys_id***, so the results display in the same order every time. The results are filtered by the ***SL1 Monitored*** and ***region*** values. The region mus be supplied by the requester, and it will only return region-specific companies.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
domainSep	false
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status

Code	*Value*
200	OK
400	Query parameter \'region\' is not defined and is required.

Fixed Internal Query

Domain:

```
'x_sclo_scilogic_region=’ + region + ‘^x_sclo_scilogic_monitored=true^sys_
domain!=global'
```

Non-Domain:

```
'x'_sclo_scilogic_monitored=true^x_sclo_scilogic_idISNOTEMPTY^x_sclo_scilogic_region'’
+ region
```

Example

```
https://<your instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/
companies?region=ScienceLogic&sysparm_offset=0&sysparm_limit=100
```

Example (Response)

```
{
  "results": [
    {
      "country": "USA",
      "notes": "What's on your digital horizon?",
      "city": "San Jose",
      "sys_updated_on": "2018-11-30 16:03:45",
      "sys_class_name": "core_company",
      "sys_id": "1ac84f95dbce2700dc44f00fbf9619c8",
      "sys_updated_by": "is4user1",
      "market_cap": "0",
      "street": "170 West Tasman Dr.",
      "sys_created_on": "2018-11-27 16:32:33",
      "state": "CA",
      "sys_created_by": "admin",
      "zip": "95134",
      "profits": "0",
      "revenue_per_year": "0",
      "sys_mod_count": "4",
      "x_sclo_scilogic_id": "1",
      "x_sclo_scilogic_monitored": "true",
      "phone": "18005532447",
      "name": "Cisco Systems, Inc.",
      "x_sclo_scilogic_region": "Cisco"
    }
  ],
  "sysparm_offset": 0,
  "sysparm_limit": 1,
  "return_count": 1,
  "total_count": 1
}
```

Device Identification Engine

HTTP Method

POST

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/IdentificationEngine
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/IdentificationEngine
```

This operation handles all creates and updates to the CMDB. This operation incorporates Identification Engine and uses the Identification and Reconciliation framework to properly import devices into the CMDB as a configurable discovery source. SL1 uses the classification GET to populate the JSON object.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
test	true

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/
IdentificationEngine?test=true
```

Example (Body)

```
[
  {
    "items": [
      {
        "className": "cmdb_ci_linux_server",
        "values": {
          "name": "Postman Test Server 1",
          "serial_number": "9876EFGH",
          "mac_address": "BF:D4:D6:6E:56:F1",
          "ip_address": "10.10.10.4",
          "ram": "16000",
          "x_sclo_scilogic_region": "ScienceLogic",
          "x_sclo_scilogic_id": "1"
        }
      }
    ]
  },
  {
    "items": [
      {
        "className": "cmdb_ci_linux_server",
        "values": {
          "name": "Postman Test Server 2",
          "serial_number": "HGFE6789",
          "mac_address": "87:54:3C:8C:2A:A3",
          "ip_address": "10.10.10.5",
          "ram": "16000",
          "x_sclo_scilogic_region": "ScienceLogic",
          "x_sclo_scilogic_id": "2"
        }
      }
    ]
  }
]
```

Example Business Service (Body)

```
[
{
"items": [
{
"className": "cmdb_ci_service",
"values": {
"name": "Integration Service",
"busines_criticality": "1 - most critical",
"used_for": "Production",
"operational_status": "1",
"service_classification": "Technical Service",
"comments": "Postman",
"x_sclo_scilogic_region": "ScienceLogic",
"x_sclo_scilogic_id": "1570"
}
},
{
"className": "cmdb_ci_linux_server",
"values": {
"name": "Postman Test Server",
"serial_number": "7MDvqrSNyd",
"manufacturer": "ScienceLogic, Inc.",
"model_id": "",
"mac_address": "EE:D6:0B:79:32:C7",
"ip_address": "10.10.10.224",
"ram": "16000",
"x_sclo_scilogic_region": "ScienceLogic",
"x_sclo_scilogic_id": "10"
}
}
],
"relations": [
{
"type": "Depends on::Used by",
"parent": 0,
"child": 1
}
]
}
]
```

Example (Response)

```
{
  "result": [
    {
      "items": [
        {
          "className": "cmdb_ci_linux_server",
          "operation": "NO_CHANGE",
          "sysId": "7fb39667dba12380dc44f00fbf961936",
          "identifierEntrySysId": "fb27f69cc3000200d8d4bea192d3ae67",
          "identificationAttempts": [
            {
              "identifierName": "Hardware Rule",
              "attemptResult": "SKIPPED",
              "attributes": [
                "serial_number",
                "serial_number_type"
              ],
              "searchOnTable": "cmdb_serial_number"
            },
            {
              "identifierName": "Hardware Rule",
              "attemptResult": "MATCHED",
              "attributes": [
                "serial_number"
              ],
              "searchOnTable": "cmdb_ci_hardware"
            }
          ]
        }
      ],
      "relations": [

      ]
    }
  ]
}
```

Discovery Dependents

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/discovery_dependent
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/discovery_dependent
```

This operation pulls all Discovery-dependent records that are tied to the ***region*** value, which is used for the catalog request process. Based on the request type, this operation returns a formatted JSON object. This operation pulls all the required information for both SL1 processes: Discovery Session and Create Virtual Device. Both requests require different information and are formatted accordingly.

The basic catalog item Device Discovery is set up as information collection to support the process within SL1. The Service Catalog has been simplified to its most basic form. The Service Catalog moves the request into the correct state to be picked up by the GET Request and then waits for its return before completing the workflow.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	*Value*
200	OK
400	Query parameter \'region\' is not defined and is required.

Fixed Internal Query

Region Specific: 'region=' + region

Example

```
https://<your instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/discovery_dependent?region=del_test&sysparm_offset=0&sysparm_limit=100
```

Example (Response)

```
{
    "results": [
        {
            "sys_updated_on": "2019-08-28 18:03:50",
            "type": "credential",
            "type_label": "Credentials",
            "sys_id": "0491aae51b273f0045c8db1dcd4bcbc2",
            "hostname": "example.com",
            "sys_updated_by": "is4user1",
            "sys_created_on": "2019-08-28 18:03:50",
            "name": "AppDynamics Example",
            "id": "93",
            "category": "soapCredentials",
            "region": "del_test",
            "sys_created_by": "is4user1"
        }
    ],
    "sysparm_offset": 0,
    "sysparm_limit": 1,
    "return_count": 1,
    "total_count": 150
}
```

File Systems

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/file_systems
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/file_systems
```

This operation pulls all the fields from the File System table. The return is ordered by **sys_id**, so the results display in the same order every time. The results are filtered by the ***SL1 monitored*** and ***SL1 ID*** field on the ServiceNow side. This operation requires the region to be supplied by the requester, it returns only region-supplied configuration items.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	***Value***

200	OK
400	Query Parameter \'region\' is not defined and is required.

Fixed Internal Query

```
'x_sclo_scilogic_monitored=true^x_sclo_scilogic_idISNOTEMPTY^x_sclo_scilogic_region='
+ region
```

Example

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/file_
systems?region=ScienceLogic&sysparm_offset=0&sysparm_limit=100
```

Example (Response)

```
{
  "results": [
    {
      "operational_status": "1",
      "sys_updated_on": "2018-11-12 21:59:52",
      "media_type": "fixed",
      "sys_created_by": "admin",
      "sys_domain_path": "/",
      "sys_class_name": "cmdb_ci_file_system",
      "computer": "d83dac0adb4dab00dc44f00fbf961919",
      "x_sclo_scilogic_monitored": "true",
      "x_sclo_scilogic_region": "ScienceLogic",
      "sys_updated_by": "admin",
      "sys_created_on": "2018-11-12 21:59:06",
      "sys_domain": "global",
      "install_status": "1",
      "name": "/root",
      "subcategory": "File Share",
      "sys_id": "afd30ba0dbf5a380dc44f00fbf961951",
      "file_system": "ntfs",
      "sys_class_path": "/!!/!K/!!",
      "mount_point": "/root",
      "sys_mod_count": "3",
      "x_sclo_scilogic_id": "31",
      "label": "/root",
      "cost_cc": "USD",
      "category": "Resource"
    }
  ],
  "sysparm_offset": 0,
  "sysparm_limit": 100,
  "return_count": 1,
  "total_count": 1
}
```

Import Set

HTTP Method

POST

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/import_set
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/import_set
```

This operation handles the custom intake of import sets before it reaches the transform map staging table, such as x_sclo_scilogic_import_installed_software. This operations is currently only used for importing installed software (x_sclo_scilogic_import_installed_software).

Headers	
Key	***Value***
Accept	application/json
Content-Type	application/json

Parameters	
Key	***Value***
record_type (required)	x_sclo_scilogic_import_installed_software

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/import_set
```

Example (Body)

```
[
  {
  "records": [
    {
      "name": "acl-2.2.51-12.el7",
      "software": "671bafd8dba13700dc44f00fbf961953",
      "cmdb_ci": [
        "ff01a81edb1df300dc44f00fbf961947",
        "4011a81edb1df300dc44f00fbf961958",
        "f301a81edb1df300dc44f00fbf96193d",
        "7b01a81edb1df300dc44f00fbf961942",
        "c411a81edb1df300dc44f00fbf96195d",
        "7701a81edb1df300dc44f00fbf961922",
        "7b01681edb1df300dc44f00fbf9619e7",
        "fb01a81edb1df300dc44f00fbf961927"
      ],
      "active": true
    }
    ]
  }
]
```

Incidents

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/incidents
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/incidents
```

This operation pulls all records from the incident table that are created by a specific **user_id** and its related events. The results are ordered by the **sys_id** of the incident, so the results display in the same order every time. This operation is also based on the incident being in an active state. This operation returns a pre-set of data and does not return everything on the Incident and Event (x_sclo_scilogic_event) tables.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
user_id (required)	is4user1
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	***Value***

200	OK
400	Query Parameter \'user_id\' is not defined and is required.

Fixed Internal Query

```
‘sys_created_by=’ +  user_id + ‘active=true’
```

Example

```
https://<your Instance>.service-now.com/api/x_sclo_
scilogic/v1/sciencelogic/incidents?user_id=is4user1&sysparm_offset=0&sysparm_limit=100
```

Example (Response)

```
{
  "results": [
    {
      "sys_id": "0141807bdbb16300dc44f00fbf9619fc",
      "number": "INC0010135",
      "state": "2",
      "state_label": "In Progress",
      "events": [
        {
          "event_id": "16908",
          "device": {
            "sys_id": {

            }
          }
        },
        {
          "event_id": "16874",
          "device": {
            "sys_id": {

            }
          }
        },
        {
          "event_id": "16865",
          "device": {
            "sys_id": {

            }
          }
        }
      ]
    }
  ],
  "sysparm_offset": 0,
  "sysparm_limit": 1,
  "return_count": 1,
  "total_count": 44
}
```

Installed Software

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/installed_software
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/installed_software
```

This operation pulls all the fields from the software (cmdb_ci_spkg) table. The return is ordered by **sys_id**, so the results display in the same order every time. The results are filtered by the ***SL1 monitored*** field on the ServiceNow side. This operation requires the ***region*** to filter the installed software on devices.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	***Value***
200	OK

400	Query parameter \'region\' is not defined and are required.

Fixed Internal Query

'x_sclo_scilogic_monitored=true'

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/installed_
software?sysparm_offset=0&sysparm_limit=100&region=ScienceLogic
```

```
Example (Response){
"results": [
{
"operational_status": "1",
"operational_status_label": "Operational",
"sys_updated_on": "2019-05-01 06:00:09",
"install_count": "2",
"sys_updated_by": "system",
"sys_created_on": "2019-03-29 19:42:58",
"sys_domain": "global",
"sys_created_by": "admin",
"sys_domain_path": "/",
"install_status": "1",
"install_status_label": "Installed",
"name": "Test_31",
"subcategory": "Package",
"sys_class_name": "cmdb_ci_spkg",
"sys_class_name_label": "Software",
"sys_id": "1e9608fcdb2cb740dc44f00fbf961949",
"sys_class_path": "/!!/#$",
"key": "Test_31_:::_NULL",
"license_available": "-2",
"sys_mod_count": "1",
"x_sclo_scilogic_id": "31",
"model_id": "2c146728dbe8b740dc44f00fbf9619c6",
"model_id_label": "Unknown",
"cost_cc": "USD",
"cost_cc_label": "USD",
"x_sclo_scilogic_monitored": "true",
"package_name": "Test_31",
"category": "Software",
"x_sclo_scilogic_region": "AutoGenerateClass",
"installed_on": [
{
"sys_id": "5a271407dbfe6300dc44f00fbf96190f",
"id": "10",
"region": "ScienceLogic",
"monitored": "true"
},
{
"sys_id": "5a271407dbfe6300dc44f00fbf96190f",
"id": "10",
"region": "ScienceLogic",
"monitored": "true"
}
]
}
],
"sysparm_offset": 0,
"sysparm_limit": 100,
"return_count": 4,
"total_count": 4
}
```

Manufacturer

HTTP Method

POST

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/manufacture
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/manufactures
```

This operation does not populate any data into ServiceNow. Instead, this operation takes an array of manufacturer names and attempts to line them up with manufacturers already in ServiceNow. Then the operation returns the sys_id of manufacturers it was able find based on matching name. If the Normalization Data Services Client is active on the target instance, this operation uses those tables to find a matching company record; otherwise the operation will match on whether name and manufacturer is true on the core_company table.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/
manufacture
```

Example (Body)

```
{
"manufactures": [
"Cisco Systems Inc",
"Cisco Systems, Incorporated",
"CiscoSystems",
"American Power Conversion Inc.",
"APC Corp",
"Apc",
"IBM",
"IBM CORP",
"International Business Machines",
"Juniper Systems",
"Juniper Networks,Inc",
"Juniper Solutions"

]
}
```

Example (Response)

```
{
"result": {
"Cisco Systems Inc": "",
"Cisco Systems, Incorporated": "",
"CiscoSystems": "",
"American Power Conversion Inc.": "",
"APC Corp": "",
"Apc": "",
"IBM": "",
"IBM CORP": "",
"International Business Machines": "",
"Juniper Systems": "",
"Juniper Networks,Inc": "",
"Juniper Solutions": ""
}
}
```

Model

HTTP Method

POST

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/model
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/models
```

This operation does not populate any data into ServiceNow. Instead, this operation takes an array of model names and attempts to line them up with models already in ServiceNow and returns the sys_id of models it was able to find based on matching name.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/models
```

Example (Body)

```
{
"models": [
"4331 ISR",
"7206VXR",
"7609S",
"AS5300",
"ASR5000",
"Catalyst 3560G-24TS",
"Catalyst 4948",
"Catalyst 6509-CatOS",
"BIG-IP Viprion B4300",
"F5 BIG-IP DNS",
"BIG-IP Wide IP Container",
"BIG-IP Data Center Container"
]
}
```

Example (Response)

```
{
"result": {
"4331 ISR": "",
"7206VXR": "",
"7609S": "",
"AS5300": "",
"ASR5000": "",
"Catalyst 3560G-24TS": "",
"Catalyst 4948": "",
"Catalyst 6509-CatOS": "",
"BIG-IP Viprion B4300": "",
"F5 BIG-IP DNS": "",
"BIG-IP Wide IP Container": "",
"BIG-IP Data Center Container": ""
}
}
```

Network Adapters

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/network_adapters
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/network_adapters
```

This operation pulls all the fields from the network adapter table. The return is ordered by **sys_id**, so the results display in the same order every time. The results are filtered by the ***SL1 monitored*** and ***SL1 ID*** field on the ServiceNow side. This operation requires the region to be supplied by the requester, and it only returns region-supplied configuration items.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	***Value***

200	OK
400	Query Parameter \'region\' is not defined and is required.

Fixed Internal Query

```
‘x_sclo_scilogic_monitored=true^x_sclo_scilogic_idISNOTEMPTY^x_sclo_scilogic_region=’
+ region
```

Example (Request URL)

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/network_
adapters?region=ScienceLogic&sysparm_offset=0&sysparm_limit=100
```

Example (Response)

```
{
  "results": [
    {
      "operational_status": "1",
      "sys_updated_on": "2018-11-12 21:29:23",
      "sys_updated_by": "admin",
      "sys_created_on": "2018-11-12 21:27:48",
      "sys_domain": "global",
      "sys_created_by": "admin",
      "cmdb_ci": "d83dac0adb4dab00dc44f00fbf961919",
      "sys_domain_path": "/",
      "install_status": "1",
      "name": "eth0",
      "subcategory": "Network",
      "sys_class_name": "cmdb_ci_network_adapter",
      "sys_id": "33ac36acdbb5a380dc44f00fbf961963",
      "netmask": "255.255.255.0",
      "sys_class_path": "/!!/!8",
      "mac_address": "BF:D4:D6:6E:56:F1",
      "sys_mod_count": "3",
      "x_sclo_scilogic_id": "20",
      "ip_address": "10.10.10.4",
      "cost_cc": "USD",
      "x_sclo_scilogic_monitored": "true",
      "category": "Hardware",
      "x_sclo_scilogic_region": "ScienceLogic"
    }
  ],
  "sysparm_offset": 0,
  "sysparm_limit": 1,
  "return_count": 1,
  "total_count": 5
}
```

Service Requests

HTTP Method

GET

Pagination

Enabled

Resource Path

```
/api/x_sclo_scilogic/v1/sciencelogic/service_request
```

Default Resource Path

```
/api/x_sclo_scilogic/sciencelogic/service_request
```

This operation pulls all service requests that are tied to specific catalog item. Based on the request type it returns a formatted JSON object. It pulls all the required information for an SL1 Discovery session and creating a virtual device in SL1. Both requests require different information and are formatted accordingly.

The basic catalog item Device Discovery is set up as information collection to support the process within SL1. The Service Catalog has been simplified to its most basic form. The workflow moves the request into the correct state to be picked up by the GET request and then waits for its return before completing the workflow.

Headers	
Key	***Value***
Content-Type	application/json
Accept	application/json

Parameters	
Key	***Value***
region (required)	ScienceLogic
state	2
sysparm_offset	0
sysparm_limit	glide.json.export.limit, glide.ui.export.limit, glide.ui.export.war.threshold

HTTP Status	
Code	*Value*
200	OK
400	Query Parameter \'region\' is not defined and is required.

Fixed Internal Query

State:

```
‘request_item.active=true^request_item.cat_item=’ + catalog + ‘^sc_item_option.item_
option_new.name=Region^sc_item_option.value=’ + region
```

Non-State:

```
‘request_item.active=true^request_item.cat_item=’ + catalog + ‘^sc_item_option.item_
option_new.name=Region^sc_item_option.value=’ + region + ‘^request_item.state=’ +
state
```

Example

```
https://<your Instance>.service-now.com/api/x_sclo_scilogic/v1/sciencelogic/ service_
request?region=Cisco
```

Example (Response)

```
{
  "results": [
    {
      "number": "RITM0010018",
      "sysid": "00365de2db1a2340dc44f00fbf961941",
      "state": "2",
      "request_type": "Discover Device",
      "region": "Cisco",
      "log_all": "false",
      "ip_hostname_list": "167.132.14.15",
      "credentials": [
        {
          "Category": "Linux",
          "ID": "1"
        }
      ],
      "discover_non_snmp": "false",
      "model_devices": "true",
      "dhcp": "false",
      "device_model_cache_ttl_h": "2",
      "collection_server": "1",
      "organization": "1",
      "add_devices_to_device_groups": [
        "test"
      ],
      "device_template": "1",
      "initial_scan_level": "System Default (Recommended)",
      "scan_throttle": "System Default (Recommended)",
      "scan_ports": "21,22,23,25,80",
      "port_scan_all": "System Default (Recommended)",
      "port_scan_timeout": "System Default (Recommended)",
      "interface_inventory_timeout": "600000",
      "maximum_allowed_interfaces": "10000",
      "bypass_interface_inventory": "false"
    },
    {
      "number": "RITM0010016",
      "sysid": "194447e8db162f00dc44f00fbf96195b",
      "state": "2",
      "request_type": "Discover Device",
      "region": "Cisco",
      "log_all": "false",
      "ip_hostname_list": "192.168.1.1",
      "credentials": [
        {
          "Category": "Linux",
          "ID": "1"
        }
      ],
      "discover_non_snmp": "false",
      "model_devices": "false",
      "dhcp": "false",
      "device_model_cache_ttl_h": "2",
      "collection_server": "1",
      "organization": "1",
      "add_devices_to_device_groups": [

      ],
      "device_template": "1",
      "initial_scan_level": "System Default (Recommended)",
      "scan_throttle": "System Default (Recommended)",
      "scan_ports": "21,22,23,25,80",
      "port_scan_all": "System Default (Recommended)",
      "port_scan_timeout": "System Default (Recommended)",
      "interface_inventory_timeout": "600000",
      "maximum_allowed_interfaces": "10000",
      "bypass_interface_inventory": "false"
    },
    {
      "number": "RITM0010014",
      "sysid": "250dae2cdbd22f00dc44f00fbf961954",
      "state": "2",
      "request_type": "create_virtual_device",
      "region": "Cisco",
      "collection_server": "1",
      "virtual_device_class": "1"
    }
  ]
}
```

Appendix

D

ServiceNow Registered Events

Overview

This appendix describes the commands and data you can use to generate registered events in ServiceNow that are queued to ServiceNow Event Management. These events can trigger actions in the Integration Service, such as specifying one or more CIs for monitoring, or putting a CI into maintenance.

These events use the `gs.eventQueue` command, using the following format:

```
eventQueue(String name, Object instance, String parm1, String parm2)
```

You can use examples found in the following ServiceNow update sets to help you customize the `gs.eventQueue` command to specify which ServiceNow events can trigger Integration Service actions:

- ScienceLogic ServiceNow Integration (Catalog UI)
- ScienceLogic ServiceNow Integration (Maintenance Mode)
- ScienceLogic ServiceNow Integration (Maintenance Schedule) Example

You will need to install these update sets in ServiceNow.

NOTE: This appendix is recommended for advanced ServiceNow administrators.

This appendix includes the following topics:

Catalog Item Events

The following events are available through the "ScienceLogic ServiceNow Integration (Catalog UI)" update set in ServiceNow.

x_sclo_scilogic.device_monitoring

This event takes the selected Configuration Items in ServiceNow, files a catalog request using the template selected by the user, and submits the catalog request.

Trigger

Custom requirement supplied by ScienceLogic implementation or the Customer directly.

Command

```
gs.eventQueue('x_sclo_scilogic.device_monitoring', region, ip_list.toString(),
region.getUniqueValue() + "," + region.x_sclo_scilogic_region + "," + silo_template);
```

Event Fields

Field	***Description***
`x_sclo_scilogic.device_monitoring`	Unique name of the event.
`region`	The table to which the event applies.
`ip_list.toString()`	Parm1: The IP, or a comma-separated list of IP addresses, that is pulled from the `ip_address` field on the **cmdb_ci** table.
`getCompany.getUniqueValue(), silo_template`	Parm2: List of three requirements that the `sys_id` of the company associated with the Configuration Item and the catalog template selected through the user interface action.

Example

The UI action / UI page is available through the "ScienceLogic ServiceNow Integration (Catalog UI Action)" update set.

x_sclo_scilogic.remove_monitoring

This action takes the selected Configuration Item or Items and submits a request through the ServiceNow service catalog for each Configuration Item.

Trigger

Custom requirement supplied by ScienceLogic implementation or the Customer directly.

Command

```
gs.eventQueue('x_sclo_scilogic.remove_monitoring',current, current.getUniqueValue(),
current.company);
```

Event Fields

Field	*Description*
`x_sclo_scilogic.remove_monitoring`	Unique name of the event.
`current`	The table to which the event applies.
`current.getUniqueValue()`	Parm1: The `sys_id` of the Configuration Item that needs to be removed
`current.company);`	Parm2: The `sys_id` of the company that is associated with the Configuration Item.

Example

The UI action / UI page is available through the "ScienceLogic ServiceNow Integration (Catalog UI Action)" update set.

Maintenance Mode Events

The following event is available through the "ScienceLogic ServiceNow Integration (Maintenance Mode)" update set in ServiceNow.

x_sclo_scilogic.device_maintenance

This event submits a list of devices to be put in to Maintenance via the Mid Server.

Trigger

Custom requirement supplied by ScienceLogic implementation or the Customer directly.

Command

```
gs.eventQueue('x_sclo_scilogic.device_maintenance',current, action, affected_ci);
```

Event Fields

Field	***Description***
`x_sclo_scilogic.device_maintenance`	Unique name of the event.
`current`	The table to which the event applies.
`action`	Parm1: An array that includes action to be performed (enable_maintenance or disable_maintenance) and the `sys_id` of the task. Task is not required, but the action is.
`affected_ci`	Parm2: An array of device sys_ids that need to be enable or disabled maintenance mode.

Example

Business rule (ScienceLogic ServiceNow Integration (Maintenance Mode Business rule)).

Maintenance Schedule Events

The following event is available through the "ScienceLogic ServiceNow Integration (Maintenance Schedule) Example" update set in ServiceNow.

x_sclo_scilogic.device_maintenance_skd

This event submits a list of devices and creates or cancels scheduled maintenance via the Mid Server.

Trigger

Custom requirement supplied by ScienceLogic implementation or the Customer directly.

Command

```
gs.eventQueue('x_sclo_scilogic.device_maintenance_skd',current, 'schedule',
current.getUniqueValue());
```

Event Fields

Field	***Description***
`x_sclo_scilogic.device_maintenance_skd`	Unique name of the event.
`current`	The table to which the event applies.
`'schedule'`	Parm1: Accepts two variables: *schedule* or *cancel*. - **[schedule]**. Creates a schedule in SL1. - ***cancel***: Deletes any currently scheduled maintenance.
`current.getUniqueValue()`	Parm2: An array of device `sys_id` values that need to be enabled or disabled for maintenance mode.

Example

The UI action / UI page is available via the "ScienceLogic ServiceNow Integration (Maintenance Schedule) Example" update set.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

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 on this Site, information on this Site 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 in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

ScienceLogic

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

International: +1-703-354-1010