



---

# ServiceNow Incident Synchronization PowerPack

Version 3.0.0

---

# Table of Contents

<b>Introduction to the ServiceNow Incident Synchronization PowerPack</b> .....	<b>7</b>
Architecture Overview for ServiceNow Synchronization PowerPacks .....	9
SL1 and ServiceNow Terminology .....	10
Dependency Map for ServiceNow Synchronization PowerPacks .....	10
Prerequisites for ServiceNow Synchronization PowerPacks .....	11
Contents of the Synchronization PowerPack .....	12
PowerFlow Applications .....	12
PowerFlow Application (Internal) .....	12
<b>Installing and Configuring the Incident Synchronization PowerPack</b> .....	<b>13</b>
Downloading, Importing, and Installing the ServiceNow Incident Synchronization PowerPack .....	14
Downloading the Synchronization PowerPack .....	14
Importing the Synchronization PowerPack .....	14
Installing the Synchronization PowerPack .....	15
Allowing Cross-Scoped Access in ServiceNow .....	16
Installing the "ScienceLogic SL1 : CMDB & Incident Automation" Application in ServiceNow .....	18
Installing the ServiceNow Base Pack PowerPack in SL1 .....	19
Validating Network Communications .....	20
Checking DNS .....	20
Checking HTTPS and JSON .....	21
HTTP Codes .....	21
Creating a ServiceNow Group .....	22
Creating a ServiceNow User .....	24
<b>Configuring Applications for the Incident Synchronization PowerPack</b> .....	<b>26</b>
Overview of the Run Book Automation for Incident Sync .....	28
Configuring Incident Sync .....	30
Creating and Aligning a Configuration Object .....	30
Creating a Configuration Object .....	31
Aligning a Configuration Object .....	33
Creating a ServiceNow Credential in SL1 .....	35
Enabling the Run Book Automation Policies .....	38
Enabling and Customizing the Run Book Action Policy .....	39
Customizing the Snippet Code in the Input Parameters Pane .....	40
Customizing Logging in the Run Book Action .....	43
Sending Custom Data to ServiceNow Using the Passthrough Option .....	43
Passing Custom Data to ServiceNow .....	44
Snippet Code Example .....	46
Configuring the "ServiceNow: Click to Create Incident" Automation Policy .....	47
Viewing Incidents and Events .....	49
Incident Topology Suppression .....	51
Hyperlinking Events .....	56
ServiceNow Hyperlinking .....	57
Viewing the Incident Import Table in ServiceNow .....	57
ServiceNow, ScienceLogic Event, and Incident Priority Matrix .....	59
Adding Additional Fields to the Transform Map .....	60
<b>Troubleshooting the Incident Synchronization PowerPack</b> .....	<b>66</b>
Initial Troubleshooting Steps .....	67
SL1 PowerFlow .....	67
ServiceNow .....	67
Resources for Troubleshooting .....	67
Useful PowerFlow Ports .....	67

Helpful Docker Commands .....	68
Viewing Container Versions and Status .....	68
Restarting a Service .....	68
Stopping all PowerFlow Services .....	68
Restarting Docker .....	68
Viewing Logs for a Specific Service .....	69
Clearing RabbitMQ Volume .....	69
Viewing the Process Status of All Services .....	69
Deploying Services from a Defined Docker Compose File .....	70
Dynamically Scaling for More Workers .....	70
Completely Removing Services from Running .....	70
Diagnosis Tools .....	70
Retrieving Additional Debug Information (Debug Mode) .....	71
Frequently Asked Questions .....	72
Why are Incidents not getting created in ServiceNow? .....	73
What if my Incident does not have a CI? .....	74
What if the PowerFlow user interface is unresponsive and Incidents are not being generated in ServiceNow? .....	74
Why are Incident numbers not populated in SL1 on Incident creation in ServiceNow? .....	74
Why am I not getting any Incidents after disabling the firewall? .....	74
<b>Checklists for Deployment .....</b>	<b>75</b>
CMDB-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow .....	76
Sync Devices from SL1 to ServiceNow .....	76
Discover Devices from ServiceNow in SL1 .....	77
Sync Business Services from SL1 to ServiceNow .....	77
Sync File Systems from SL1 to ServiceNow .....	77
Sync Network Interfaces from SL1 to ServiceNow .....	77
Sync Installed Software from SL1 to ServiceNow .....	78
Sync Maintenance Schedules from ServiceNow to SL1 .....	78
Sync Advanced Topology from SL1 to ServiceNow .....	78
CMDB-Only ServiceNow Integration with Single SL1 and Domain-Separated ServiceNow .....	80
Sync Devices from SL1 to ServiceNow .....	80
Discover Devices from ServiceNow in SL1 .....	81
Sync Business Services from SL1 to ServiceNow .....	81
Sync File Systems from SL1 to ServiceNow .....	82
Sync Network Interfaces from SL1 to ServiceNow .....	82
Sync Installed Software from SL1 to ServiceNow .....	82
Sync Maintenance Schedules from ServiceNow to SL1 .....	82
Sync Advanced Topology from SL1 to ServiceNow .....	83
CMDB-Only ServiceNow Integration with Multiple SL1 Systems, no Domain-Separated ServiceNow .....	84
Sync Devices from SL1 to ServiceNow .....	85
Discover Devices from ServiceNow in SL1 .....	85
Sync Business Services from SL1 to ServiceNow .....	86
Sync File Systems from SL1 to ServiceNow .....	86
Sync Network Interfaces from SL1 to ServiceNow .....	86
Sync Installed Software from SL1 to ServiceNow .....	87
Sync Maintenance Schedules from ServiceNow to SL1 .....	87
Sync Advanced Topology from SL1 to ServiceNow .....	87
Incident-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow .....	88
<b>Certified Application Objects .....</b>	<b>89</b>
Roles .....	90
Tables .....	91

Table Columns (cmdb_ci) .....	91
Table Columns (core_company) .....	92
Table Columns (cmdb_group) .....	92
Script Includes .....	92
Event Registry .....	92
Scripted Actions .....	93
Data Lookup Definitions .....	93
System Properties .....	93
Catalog Item .....	94
Catalog UI Policies .....	94
Variable Sets .....	94
Catalog Client Scripts .....	95
Workflows .....	95
Scripted REST Resources .....	96
Transform Maps .....	98
Transform Scripts .....	98
<b>ServiceNow API Endpoints .....</b>	<b>99</b>
Business Services .....	101
HTTP Method .....	101
Pagination .....	101
Resource Path .....	101
Default Resource Path .....	101
Example (Request URL) .....	101
Example (Response) .....	102
Change Requests .....	103
HTTP Method .....	103
Resource Path .....	103
Default Resource Path .....	103
Fixed Internal Query .....	104
Example .....	104
Example (Response) .....	104
Classification version 1 .....	105
HTTP Method .....	105
Pagination .....	105
Resource Path .....	105
Default Resource Path .....	105
Fixed Internal Query .....	106
Example .....	106
Example (Response) .....	106
Classification version 2 .....	108
HTTP Method .....	108
Pagination .....	108
Resource Path .....	108
Default Resource Path .....	108
Fixed Internal Query .....	109
Example .....	109
Example (Response) .....	109
CMDB Group .....	111
HTTP Method .....	111
Pagination .....	111
Resource Path .....	111
Default Resource Path .....	111

Example (Request URL)	111
Example (Body)	111
Example (Response)	112
Companies	113
HTTP Method	113
Pagination	113
Resource Path	113
Default Resource Path	113
Fixed Internal Query	114
Example	114
Example (Response)	114
Device Identification Engine	116
HTTP Method	116
Pagination	116
Resource Path	116
Default Resource Path	116
Example (Request URL)	116
Example (Body)	116
Example Business Service (Body)	117
Example (Response)	118
Discovery Dependents	119
HTTP Method	119
Pagination	119
Resource Path	119
Default Resource Path	119
Fixed Internal Query	120
Example	120
Example (Response)	120
File Systems	121
HTTP Method	121
Pagination	121
Resource Path	121
Default Resource Path	121
Fixed Internal Query	122
Example	122
Example (Response)	122
Import Set	123
HTTP Method	123
Resource Path	123
Default Resource Path	123
Example (Request URL)	123
Example (Body)	123
Incidents	125
HTTP Method	125
Pagination	125
Resource Path	125
Default Resource Path	125
Fixed Internal Query	126
Example	126
Example (Response)	126
Installed Software	128
HTTP Method	128

Pagination .....	128
Resource Path .....	128
Default Resource Path .....	128
Fixed Internal Query .....	129
Example (Request URL) .....	129
Manufacturer .....	131
HTTP Method .....	131
Pagination .....	131
Resource Path .....	131
Default Resource Path .....	131
Example (Request URL) .....	131
Example (Body) .....	131
Example (Response) .....	132
Model .....	133
HTTP Method .....	133
Resource Path .....	133
Default Resource Path .....	133
Example (Request URL) .....	133
Example (Body) .....	133
Example (Response) .....	134
Network Adapters .....	135
HTTP Method .....	135
Pagination .....	135
Resource Path .....	135
Default Resource Path .....	135
Fixed Internal Query .....	136
Example (Request URL) .....	136
Example (Response) .....	136
Service Requests .....	137
HTTP Method .....	137
Pagination .....	137
Resource Path .....	137
Default Resource Path .....	137
Fixed Internal Query .....	138
Example .....	138
Example (Response) .....	138

---

# Chapter

# 1

## Introduction to the ServiceNow Incident Synchronization PowerPack

---

### Overview

This chapter describes the *ServiceNow Incident Synchronization PowerPack*, which is the ScienceLogic integration with the ServiceNow Incident Management Module.

This Synchronization PowerPack automatically logs, de-duplicates, correlates, updates, and appends ServiceNow Incidents, reducing the amount of time to resolve critical service issues. This Synchronization PowerPack covers the entire Incident life cycle, providing a bi-directional integration between SL1 events and ServiceNow Incidents, while providing a granular view into both the event and the associated Incident.

**WARNING:** You should not use the *ServiceNow Incident Synchronization PowerPack* and the *ServiceNow Events Synchronization PowerPack* on the same PowerFlow system.

**NOTE:** After the 2.1.0 platform release, the *Integration Service* was rebranded as *SL1 PowerFlow*, and the *Automation Builder* was rebranded as *SL1 PowerFlow builder*.

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

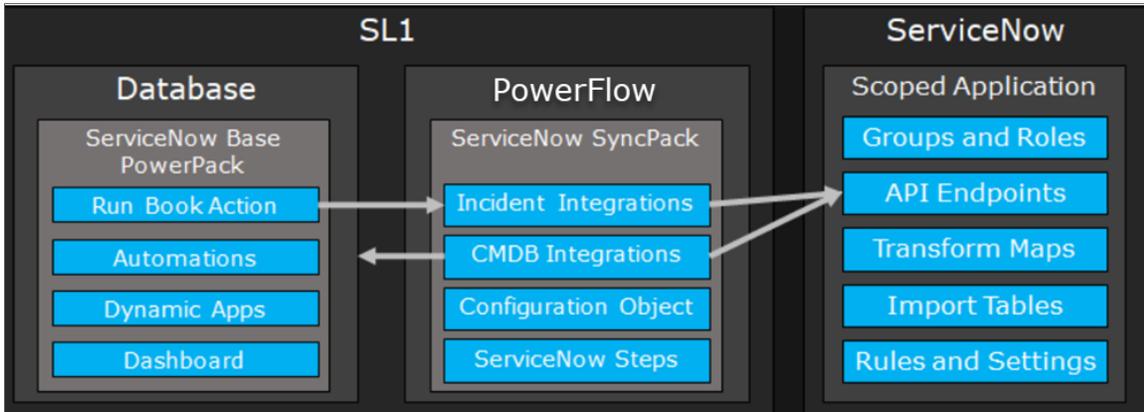
This chapter covers the following topics:

<a href="#">Architecture Overview for ServiceNow Synchronization PowerPacks</a> .....	9
<a href="#">SL1 and ServiceNow Terminology</a> .....	10
<a href="#">Dependency Map for ServiceNow Synchronization PowerPacks</a> .....	10

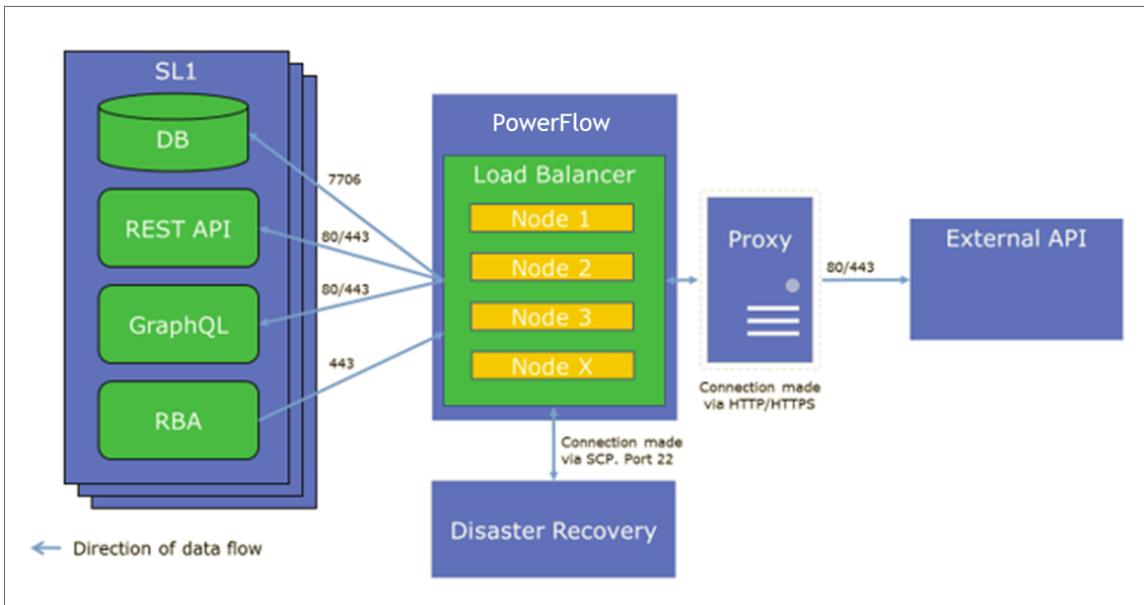
<i>Prerequisites for ServiceNow Synchronization PowerPacks</i> .....	11
<i>Contents of the Synchronization PowerPack</i> .....	12

# Architecture Overview for ServiceNow Synchronization PowerPacks

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



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



---

## SL1 and ServiceNow Terminology

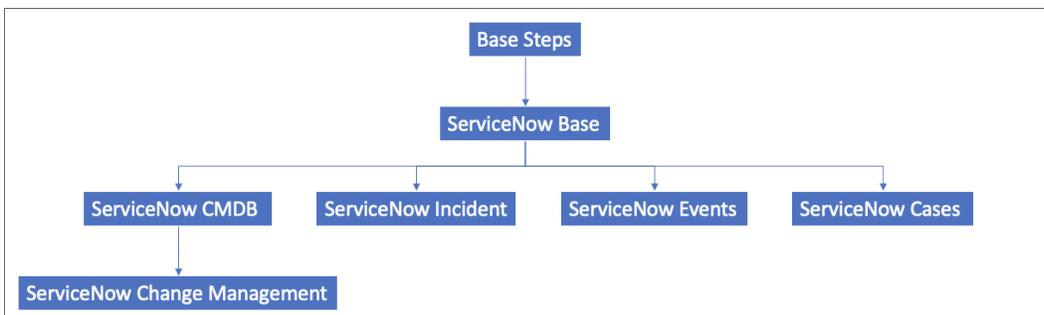
The following table lists the different names for the shared elements in SL1 and ServiceNow:

SL1	ServiceNow
Asset, Custom Attribute	Asset (ITAM)
Device	CI (Configuration Item)
Discovery Session	Service Request, Catalog Request
Event	Incident, Event, or Case (depending on the Synchronization PowerPack you are using)
Alert	Event
Organization	Company, Domain
Schedule, Maintenance Schedule	Change Request, Change Schedule
Topology, Relationships, Dynamic Component Mapping and Relationships (DCM+R)	Dependency View, Affected CIs

---

## 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 *SL1 PowerFlow 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 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 PowerFlow (formerly called 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 PowerFlow platform, 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 **SL1 PowerFlow for Developers** manual.
- If you are starting out with version 2.0.0 of the PowerFlow 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.

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

Source IP	PowerFlow Destination	PowerFlow Source Port	Destination Port	Requirement
PowerFlow	SL1 API	Any	TCP 443	SL1 API Access
PowerFlow	ServiceNow API	Any	TCP 443	ServiceNow API Access
SL1 Run Book Action	PowerFlow	Any	TCP 443	Send SL1 data to PowerFlow
PowerFlow	SL1 Database	Any	TCP 7706	SL1 Database Access

---

# Contents of the Synchronization PowerPack

This section lists the contents of the *ServiceNow Incident Synchronization PowerPack*.

## PowerFlow Applications

The following PowerFlow applications are included with the *ServiceNow Incident Synchronization PowerPack*:

- **Create or Update ServiceNow Incident from SL1 Event.** Reads SL1 events and creates or updates the corresponding ServiceNow Incident.
- **Sync Incident State from ServiceNow to SL1 Event.** Clears or updates SL1 events when the related ServiceNow Incident is updated.
- **Update ServiceNow Incident when SL1 Event is Acknowledged.** Updates the synced ServiceNow incident when the corresponding SL1 event is acknowledged.
- **Update ServiceNow Incident when SL1 Event is Cleared.** Updates the synced ServiceNow Incident when the corresponding SL1 event is cleared.

## PowerFlow Application (Internal)

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

- **Bulk Update SL1 Events.** Bulk updates SL1 events with a given payload.

## Installing and Configuring the Incident Synchronization PowerPack

---

### Overview

This chapter describes the how to install and configure the *ServiceNow Incident Synchronization PowerPack* and the components used by the Synchronization PowerPack.

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

1. In PowerFlow, download, import, and install the *ServiceNow Incident Synchronization PowerPack*.
2. In ServiceNow, enable cross-scoped access and install the "ScienceLogic SL1 : CMDB & Incident Automation" application.
3. In SL1, install the *ServiceNow Base Pack PowerPack*.
4. Validate network communications.
5. In ServiceNow, create a ServiceNow group and user account.

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

<a href="#">Downloading, Importing, and Installing the <i>ServiceNow Incident Synchronization PowerPack</i></a>	14
<a href="#">Allowing Cross-Scoped Access in ServiceNow</a>	16
<a href="#">Installing the "ScienceLogic SL1: CMDB &amp; Incident Automation" Application in ServiceNow</a>	18
<a href="#">Installing the <i>ServiceNow Base Pack PowerPack</i> in SL1</a>	19
<a href="#">Validating Network Communications</a>	20
<a href="#">Creating a ServiceNow Group</a>	22
<a href="#">Creating a ServiceNow User</a>	24

---

# Downloading, Importing, and Installing the ServiceNow Incident Synchronization PowerPack

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

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

## Downloading the Synchronization PowerPack

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

To locate and download the Synchronization PowerPack:

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

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

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

## Importing the Synchronization PowerPack

To import a Synchronization PowerPack in the PowerFlow user interface:

1. On the **SyncPacks** page of the PowerFlow user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.

2. Click **[Browse]** and select the **.whl** file for the Synchronization PowerPack you want to install.

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

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

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

## Installing the Synchronization PowerPack

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

To install a Synchronization PowerPack in the PowerFlow user interface:

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

**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 Filter 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" application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

**TIP:** The most common error that occurs when installing a Synchronization PowerPack is that the Synchronization PowerPack dependencies are not installed. Review the "System Requirements" section of the release notes for that Synchronization PowerPack to ensure that you have installed all of the required applications for that Synchronization PowerPack. To view a list of additional PowerFlow and SL1 products that are required by the various Synchronization PowerPacks, see the SL1 PowerFlow Dependency Matrix.

---

## 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`

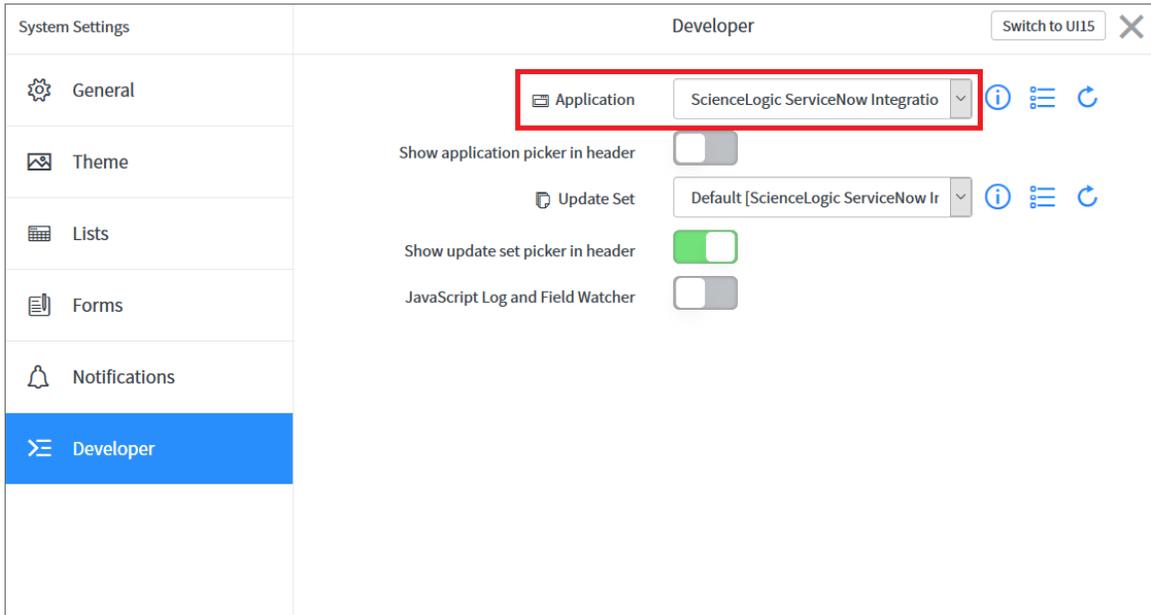
**Below is a rendering of the page up to the first error.**

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:

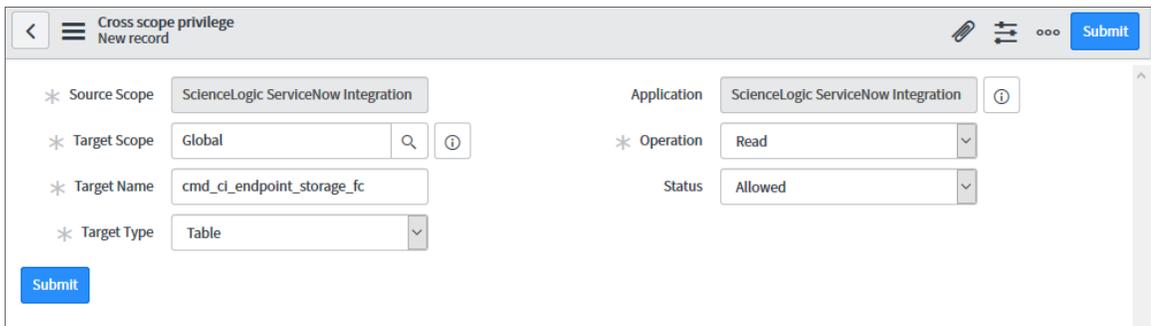
1. Log in to your ServiceNow instance.
2. Click the **Settings** icon (⚙️) and select the **Developer** tab. The **Developer System Settings** window appears:



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:



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](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

The *ServiceNow Incident Synchronization PowerPack* uses the "ScienceLogic SL1: CMDB & Incident Automation Application" to sync incident status update from ServiceNow back to SL1.

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:** PowerFlow instances running version 2.0.0 or later of the ServiceNow 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 applications to version 2.0.0 or later on all PowerFlow 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 the ServiceNow Base Pack PowerPack in SL1

The *ServiceNow Base Pack PowerPack* monitors the ServiceNow Incident and CMDB tables, and it returns information about Incident types, priorities, and states, displaying the information in an easy-to-consume dashboard. The PowerPack also returns information about the CI records that are actively being synced between SL1 and ServiceNow with PowerFlow, including basic CI metadata as well as an overall count.

Also, this PowerPack is a critical component of the Incident Sync Integration with ServiceNow, using Run Book Automations to integrate with the ServiceNow Incident Management module.

**TIP:** By default, installing a new version of a PowerPack overwrites all content in that PowerPack that has already been installed on the target system. You can use the **Enable Selective PowerPack Field Protection** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. For more information, see the **System Administration** manual.

To install the *ServiceNow Base Pack* PowerPack:

1. Search for and download the *ServiceNow Base Pack* PowerPack from the **PowerPacks** page on the [ScienceLogic Support Site](#).
2. In SL1, go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. Click the **Actions** menu and choose *Import PowerPack*. The **Import PowerPack** modal appears.
4. Click **[Browse]** and navigate to the *ServiceNow Base Pack* PowerPack file from step 1.

5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal page displays a list of the PowerPack contents.
6. Click **[Install]**. After the installation is complete, the *ServiceNow Base Pack* PowerPack appears on the **PowerPack Manager** page.

---

## Validating Network Communications

All communication between SL1 and ServiceNow is done through TCP port 443. To allow communication between SL1 and ServiceNow, the SL1 Database Server, Data Collector, or All-In-One Appliance must have external access to the ServiceNow instance. No inbound TCP ports are required to be open to the SL1 server. Outbound communication may use NAT or be direct.

**NOTE:** All firewall session-limiting policies must be disabled. If firewall session-limiting policies are enabled, HTTPS requests might be dropped by the firewall, resulting in data loss. Check with your security or firewall administrator to make sure there are no session limiting policies on TCP port 443 for your SL1 servers.

## Checking DNS

Because ServiceNow is a cloud-based service, DNS must be configured on all SL1 servers that communicate with your ServiceNow instance.

**NOTE:** ServiceNow instances are generally named as: ***your-instance.service-now.com***, where ***your-instance*** is the name of your ServiceNow server. The examples below use ***mycompany.service-now.com***. Your instance name will be unique to your subscription.

To validate that your SL1 server has proper DNS name resolution configured, test network connectivity and name resolution using the nmap command, which is available from the command line of any SL1 server:

```
nmap -sT -p 443 mycompany.service-now.com
```

If the test was successful, you will see a message similar to the following:

```
Starting Nmap 5.51 ( http://nmap.org ) at 2013-11-12 20:22 UTC
Nmap scan report for mycompany.service-now.com (199.91.136.100)
Host is up (0.067s latency).
PORT STATE SERVICE
443/tcp open  https
```

If domain name resolution fails, you will see a message similar to:

```
Failed to resolve given hostname/IP: mycompany.service-now.com.
```

## Checking HTTPS and JSON

You can administer a simple test to determine if the ServiceNow JSON Plug-in web service is configured and operating using the Basic Authentication method on your ServiceNow instance. To do so, run the following command from the ScienceLogic Central Database or All-In-One Appliance:

**NOTE:** In the example below, replace the `admin:admin` username and password key/value pair with your ServiceNow administrator username and password and `mycompany.service-now.com` with your ServiceNow instance name.

```
curl --location -vu admin:admin -H "Accept: application/json" -H "Content-Type: application/json" 'https://mycompany.service-now.com/api/now/table/incident'
```

If not successful, the following message appears:

```
HTTP/1.1 401 Unauthorized
```

If successful, a JSON encoded string starting with the "result" variable appears:

```
{"result":[{"upon_approval":"","location":"1083361cc611227501b682158cabf646"},...
```

## HTTP Codes

HTTP codes are necessary for identifying specific problems. The following table lists typical HTTP codes that might occur when testing the ServiceNow JSON Web Service.

Code	Definition
401	Unauthorized. Check that the username and password are correct and properly formatted.
403	Forbidden. ServiceNow understood the request, but either the URL is incorrect, or the user account does not have permission to see the requested object.
404	The ServiceNow server has not found anything matching the requested URL. Check to make sure there is data in the target table.
200	Success.
201	Success. Data is posted.

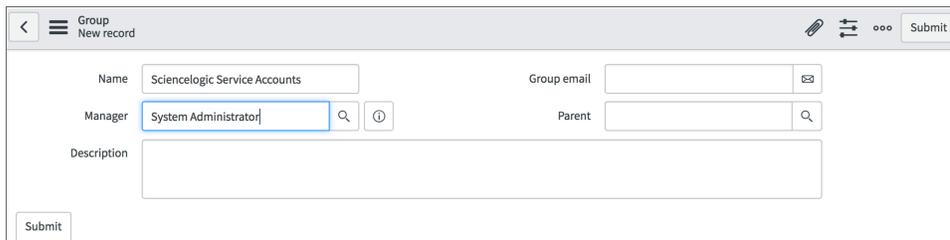
**TIP:** For more information about the ServiceNow JSON Web Service and the Table API, see [http://wiki.servicenow.com/index.php?title=Table\\_API](http://wiki.servicenow.com/index.php?title=Table_API). If you continue to have problems, please contact either ScienceLogic or ServiceNow customer support.

# Creating a ServiceNow Group

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

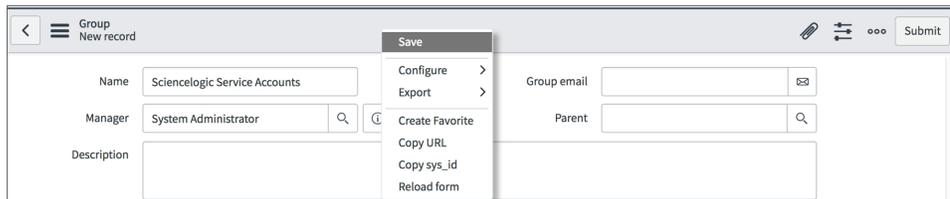
To create a ServiceNow Account for ScienceLogic Incident management:

1. In ServiceNow, search in the filter navigator for "groups".
2. On the **Groups** page (System Security > Groups), click **[New]**. A **New record** page appears.
3. In the **New record** page, type the group name and any additional information. **Name** is the only required field:



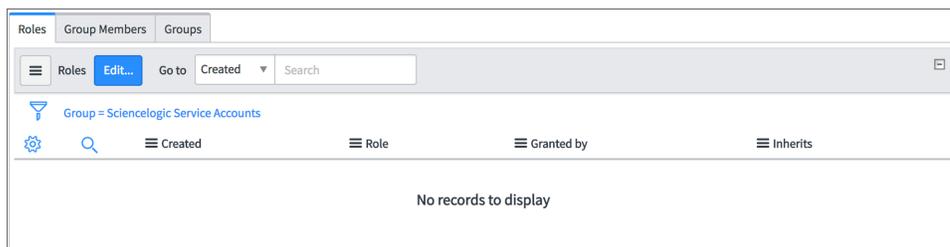
The screenshot shows the 'New record' form for a Group in ServiceNow. The form has a header with a back arrow, a hamburger menu, the text 'Group New record', and icons for edit, list, and a 'Submit' button. The form fields are: Name (Sciencelogic Service Accounts), Group email (empty), Manager (System Administrator), and Parent (empty). There is a large empty text area for Description. A 'Submit' button is located at the bottom left.

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



The screenshot shows the 'New record' form for a Group in ServiceNow. The form is the same as in the previous screenshot. A context menu is open over the header area, showing options: Save, Configure, Export, Create Favorite, Copy URL, Copy sys\_id, and Reload form.

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



The screenshot shows the 'Roles' tab for the Group 'Sciencelogic Service Accounts'. The page has a header with 'Roles', 'Edit...', 'Go to', 'Created', and a search bar. Below the header, there is a filter icon and a table with columns for 'Created', 'Role', 'Granted by', and 'Inherits'. The table is currently empty, displaying 'No records to display'.

6. Search for `x_sclo_scilogic.Admin` and move it to the **Roles List** column using the arrow buttons:

The screenshot shows a role assignment dialog. At the top, there are buttons for 'Add Filter' and 'Run filter' with a help icon. Below are three dropdown menus: '-- choose field --', '-- oper --', and '-- value --'. The main area is split into two panes: 'Collection' on the left and 'Roles List' on the right. The 'Collection' pane has a search bar and a list of roles: action\_designer, activity\_admin, activity\_creator, admin, agent\_admin, announcement\_admin, api\_analytics\_read, app\_service\_admin, app\_service\_user, approval\_admin, approver\_user, assessment\_admin, asset, assignment\_rule\_admin, atf\_test\_admin, and atf\_test\_designer. The 'Roles List' pane shows 'IS4 - Service Accounts' with 'x\_sclo\_scilogic.Admin' highlighted in a red box. Between the panes are right and left arrow buttons. At the bottom are 'Cancel' and 'Save' buttons.

7. Click **[Save]**. Your ServiceNow Group now has an assigned Role:

The screenshot shows the 'IS4 - Service Accounts' group configuration page. The 'Roles' tab is selected. The table below shows the assigned role:

Created	Role	Granted by	Inherits
2019-01-15 12:09:31	x_sclo_scilogic.Admin	(empty)	true

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

## Creating a ServiceNow User

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

To create a ServiceNow Account for ScienceLogic Incident management:

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



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

The screenshot shows a user interface for managing groups. On the left, under 'Collection', there is a search bar containing 'sciencelogic' and a list area showing '--None--'. On the right, under 'Groups List', there is a search bar and a list area containing 'Sciencelogic Service Accounts'. Between the two panels are two arrow buttons: a right-pointing arrow and a left-pointing arrow. At the bottom of the interface are 'Cancel' and 'Save' buttons. Below the 'Save' button, the text 'Name Sciencelogic Service Accounts' is displayed.

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

The screenshot shows a record view in ServiceNow. The breadcrumb navigation at the top reads 'Roles (18) > Groups (1) > Delegates > Subscriptions > Manage Subscriptions'. The 'Roles' tab is selected and highlighted in blue. Below the breadcrumb, there is a search bar and a list of roles. The first item in the list is 'Role', and the search bar contains the text '1 to 18 of 18'.

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

## Configuring Applications for the Incident Synchronization PowerPack

---

### Overview

This chapter describes the how to configure and run the various PowerFlow applications and SL1 Run Book Automations contained in the *ServiceNow Incident Synchronization PowerPack*.

A **PowerFlow application** is a JSON object that includes all the information required for executing an integration on the PowerFlow platform. In this Synchronization PowerPack, the applications are aligned with an SL1 **Run Book Automation** that you can configure to ensure that whenever SL1 detects a new, acknowledged, or cleared Event, a corresponding Incident is created or updated in ServiceNow.

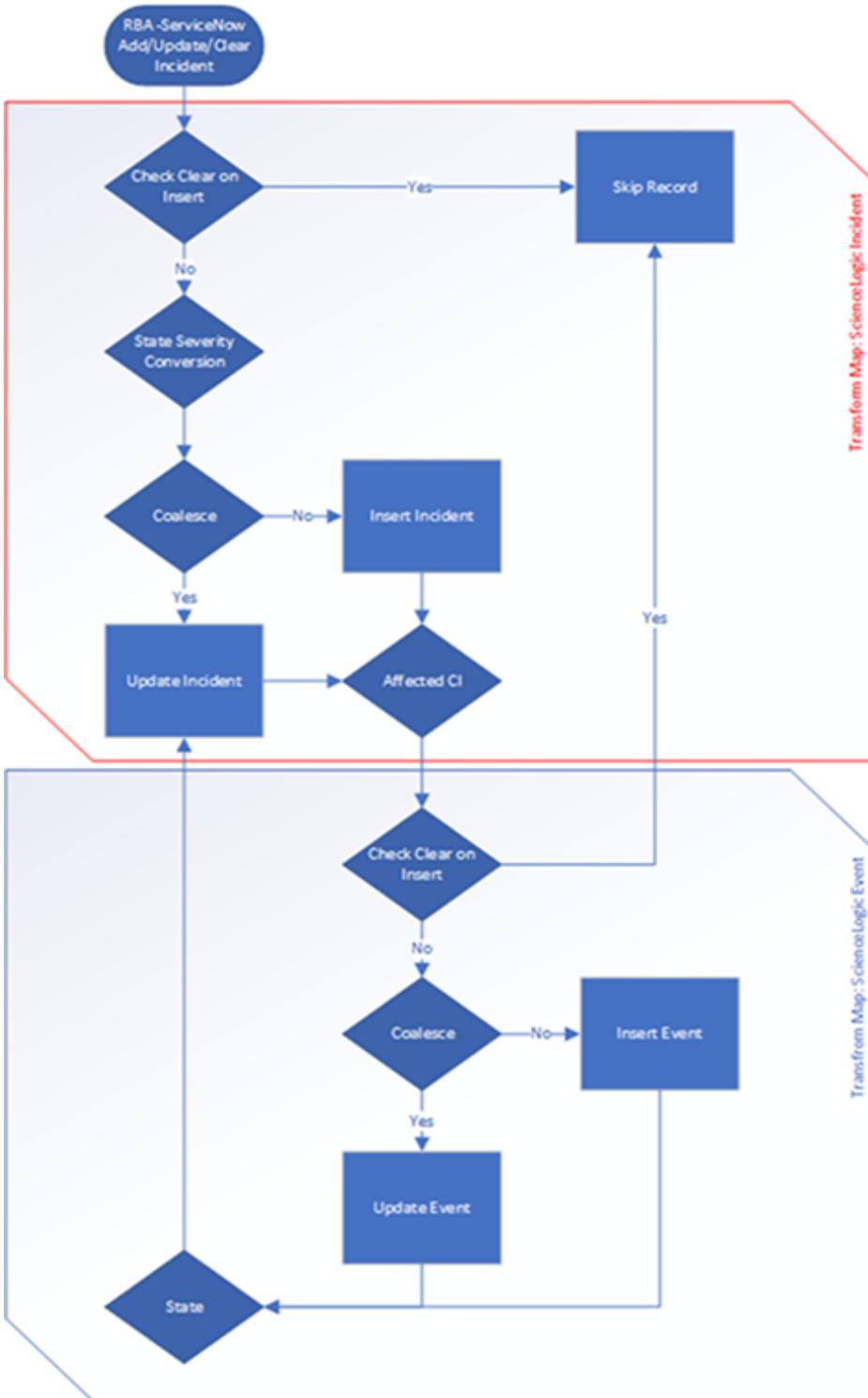
This chapter covers the following topics:

<i>Overview of the Run Book Automation for Incident Sync</i> .....	28
<i>Configuring Incident Sync</i> .....	30
<i>Creating and Aligning a Configuration Object</i> .....	30
<i>Creating a ServiceNow Credential in SL1</i> .....	35
<i>Enabling the Run Book Automation Policies</i> .....	38
<i>Enabling and Customizing the Run Book Action Policy</i> .....	39
<i>Sending Custom Data to ServiceNow Using the Passthrough Option</i> .....	43
<i>Configuring the "ServiceNow: Click to Create Incident" Automation Policy</i> .....	47
<i>Viewing Incidents and Events</i> .....	49
<i>Incident Topology Suppression</i> .....	51
<i>Hyperlinking Events</i> .....	56
<i>Viewing the Incident Import Table in ServiceNow</i> .....	57

<i>ServiceNow, ScienceLogic Event, and Incident Priority Matrix</i> .....	59
<i>Adding Additional Fields to the Transform Map</i> .....	60

# Overview of the Run Book Automation for Incident Sync

You can configure a Run Book Automation to ensure that whenever SL1 detects a new, acknowledged, or cleared event, a corresponding incident is created or updated in ServiceNow:



The "ServiceNow: Add/Update/Clear Incident" Run Book Action policy is responsible for sending the SL1 payload to PowerFlow. PowerFlow then sends that payload to ServiceNow and creates, updates, acknowledges, or clears an incident, as needed.

SL1 features three Run Book Automation policies that facilitate this process:

- ServiceNow: [Incident] Add/Update
- ServiceNow: [Incident] Event Acknowledged
- ServiceNow: [Incident] Event Cleared

**NOTE:** A fourth Run Book Automation policy, "ServiceNow: [Incident] Click to Create" lets you manually create an incident in ServiceNow by clicking the life-preserver icon (🛟) in SL1. For more information, see [Configuring the "ServiceNow: \[Incident\] Click to Create" Automation Policy](#).

**NOTE:** The "Sync Incident State from ServiceNow to SL1 Event" application does not have an associated Run Book Action that triggers Incident Sync. You must schedule this application to run every minute, or to a time suitable for your requirements. You can use a cron job to trigger this schedule, or you can use the PowerFlow user interface to schedule the application. For more information about scheduling, see the "Scheduling a PowerFlow Application" topic in the *Managing PowerFlow Applications* chapter of the **SL1 PowerFlow Platform** manual.

Each Run Book Automation policy calls a single Action in SL1. Ensure that the configuration object aligned with PowerFlow application points to the relevant SL1 system and ServiceNow instance. The Action then calls a PowerFlow application that determines the workflow to execute.

Events in SL1 frequently occur and resolve due to fluctuations in the network and other changing conditions. However, the Run Book Automation policies above use a de-duplication algorithm to ensure that only a single open ServiceNow incident exists per device. As a result, if a device already has an existing ServiceNow incident, the following updates are made to the ServiceNow incident record:

- The "Work Notes" activity log in the incident record is updated with information about the secondary event(s).
- If a secondary event is of a higher severity than the event that originally created the ServiceNow incident, then the **Impact**, **Urgency**, and **Priority** fields are updated automatically in the ServiceNow incident record. If the secondary event is of a lesser severity, those fields are not updated.
- If an event is cleared in SL1 and then later reoccurs before the incident has been "Closed" in ServiceNow, then the subsequent events appear in the original ServiceNow incident record for that device. If an incident record has been "Closed," then ServiceNow will create a new incident record when a cleared event reoccurs in SL1.
- By default, if an event is acknowledged in SL1, the ServiceNow incident record will be updated with the work notes and the acknowledging user. Clearing an SL1 event will move the ServiceNow incident record state to "Resolved". If all SL1 events associated with a ServiceNow incident record are clear, the ServiceNow incident record will, by default, move to a "Resolved" state.

**NOTE:** You can edit the Run Book Action Snippet code to adjust the behavior for changing states when an SL1 event is acknowledged or cleared.

---

## Configuring Incident Sync

This workflow covers how to set up the Run Book Automation to run Incident Sync:

1. [Create a configuration object in the PowerFlow user interface](#)
2. [Align the new configuration file with the following PowerFlow applications:](#)
  - o "Create or Update ServiceNow Incident from SL1 Event"
  - o "Sync Incident State from ServiceNow to SL1 Event"
  - o "Update ServiceNow Incident when SL1 Event is Acknowledged"
  - o "Update ServiceNow Incident when SL1 Event is Cleared"
3. [Create a ServiceNow credential in SL1](#)
4. [Enable the following Run Book Automation Policies:](#)
  - o "ServiceNow: Add/Update Incident"
  - o "ServiceNow: Event Acknowledged"
  - o "ServiceNow: Event Cleared"
3. [Enable and customize the "ServiceNow: Add/Update/Clear Incident" Run Book Action Policy](#)
4. Optionally, [send custom data to ServiceNow using the Passthrough Option](#)
5. Optionally, [enable and configure the "ServiceNow: Click to Create Incident" Policy](#)

---

## Creating and Aligning a Configuration Object

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

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

To use this Synchronization PowerPack, you will need to use an existing configuration object in the PowerFlow user interface or create a new configuration object. Next, you need to align that configuration object to the relevant applications that are triggered by the Run Book Actions in SL1.

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

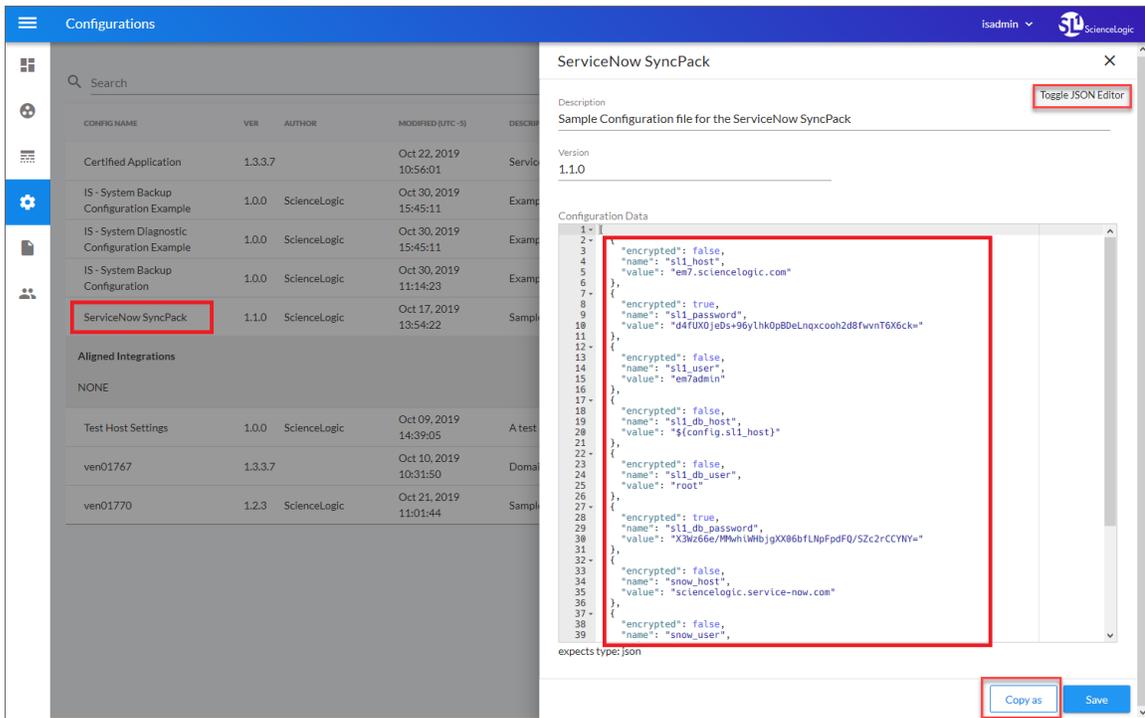
# Creating a Configuration Object

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*.

**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.

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

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



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

3. Click **[Copy as]**. The **Create Configuration** pane appears.
4. Complete the following fields:
  - **Friendly Name**. Name of the configuration object that will display on the **Configurations** page.
  - **Description**. A brief description of the configuration object.

- **Author.** User or organization that created the configuration object.
- **Version.** Version of the configuration object.

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

```
{
  "encrypted": false,
  "name": "sll_db_host",
  "value": "${config.sll_host}"
}
```

For example:

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

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

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

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

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

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

- **encrypted.** Specifies whether the value will appear in plain text or encrypted in this JSON file. If you set this to "true", when the value is uploaded, PowerFlow encrypts the value of the variable. The plain text value cannot be retrieved again by an end user. The encryption key is unique to each PowerFlow system. The value is followed by a comma.
- **name.** Specifies the name of the configuration file, without the JSON suffix. This value appears in the user interface. The value is surrounded by double-quotes and followed by a comma.
- **value.** Specifies the value to assign to the variable. The value is surrounded by double-quotes and followed by a comma.

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

## Aligning a Configuration Object

To run Incident Sync, you must "align" the configuration object to run with the following PowerFlow 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"

**NOTE:** If you want to link incidents with ServiceNow Configuration Items (CIs), you will need to run the "Sync Devices from SL1 to ServiceNow" application from the *ServiceNow CMDB Synchronization PowerPack*. If this is the first time you are running Case Sync, you will need to run the "Sync Devices from SL1 to ServiceNow" application twice to build the internal cache. For more information, see the *ServiceNow CMDB Synchronization PowerPack* manual.

To align the configuration object with the relevant applications:

1. On the **Applications** page of the PowerFlow user interface, open an "incident" application and click **[Configure]** (⚙️). The **Configurations** pane for that application appears:

**Create or Update ServiceNow Incident from SL1 Event** [Cancel] [Save]

Align configuration and save

Configuration  
scoped-app-conf

snow\_hostname: ven01056.service-no  
snow\_user: is4User1  
snow\_password: [masked]  
sl1\_hostname: 10.2.11.41

sl1\_user: em7admin  
sl1\_password: [masked]

eventDetails

1
---

expects type: json

region: QARegion10  
snow\_state: 2

eventDetails

1
---

expects type: json

correlation\_type: 5

discard      assignmentGroup      retry\_max: 0

2. From the **Configurations** drop-down, select the configuration object you want to use.
3. If needed, edit the retry values:
  - **retry\_max**. The maximum number of times PowerFlow will retry to execute the step before it stops retrying and logs a step failure. For example, if `retry_max` is 3, PowerFlow will retry after 1 second, then 2 seconds, then 4 seconds, and stop if the last retry fails. The default is 0.
  - **retry\_jitter**. Instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. The default is unselected.
  - **retry\_backoff**. Instead of using a defined interval between retries, PowerFlow will incrementally increase the interval between retries. The default is unselected.

- **retry\_backoff\_max**. The maximum time interval for the **retry\_backoff** option, in seconds. For example, This means, if you have `retry_max` set to 15, the delays will be 1, 2, 4, 8, 16, 32, 64, 120, 240, 480, 600, 600, 600, 600, and 600. The default is 600.

4. Click **[Save]** to align that configuration with the application.
5. Wait until you see the "App & Confg modifications saved" pop-up message before proceeding.
6. Repeat this process for every application you want to use.
7. Schedule the "Sync Incident State from ServiceNow to SL1 Event" PowerFlow application to run every minute, or to a time suitable for your requirements. For more information about scheduling applications, see the "Scheduling a PowerFlow Application" topic in the *Managing PowerFlow Applications* chapter of the **SL1 PowerFlow Platform** manual.

**NOTE:** The "Sync Incident State from ServiceNow to SL1 Event" application is the *only* application for Incident Sync that can be run manually or scheduled. The other three applications should only be triggered by Run Book Automations.

**NOTE:** The values for **eventDetails** and the other parameters that appear in the **Configuration** pane with a padlock icon (🔒) are populated either by the configuration object you aligned with the application or by the Run Book Action. Do not modify these values. If you encounter an error, make sure your Run Book Action is configured properly.

---

## Creating a ServiceNow Credential in SL1

To configure SL1 to communicate with ServiceNow, you must first create a SOAP/XML credential. This credential allows the Run Book Automation scripts and the Dynamic Applications in the *ServiceNow Base Pack* PowerPack to connect with your ServiceNow instance. These Run Book Automations are responsible for sending the SL1 event data to PowerFlow, which ultimately sends the data to a ServiceNow.

The **ServiceNow RBA - Example** credential from the *ServiceNow Base Pack* PowerPack is an example SOAP/XML credential that you can configure for your own use.

To configure the **ServiceNow RBA - Example** credential:

1. In SL1, go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **ServiceNow RBA - Example** credential and click its wrench icon (🔧). The **Edit SOAP/XML Credential** page appears:

The screenshot shows the 'Credential Editor' window for 'Edit SOAP/XML Credential #85'. The window has a blue title bar and a close button. Below the title bar are 'New' and 'Reset' buttons. The main area is divided into several sections:

- Basic Settings:** Profile Name (ServiceNow RBA - Example), Content Encoding ([ text/xml ]), Method ([ POST ]), HTTP Version ([ HTTP/1.1 ]), URL [ http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name ] (https://INTEGRATIONSERVICEHOSTNAME), HTTP Auth User (USERNAME), HTTP Auth Password (\*\*\*\*\*), Timeout (seconds) (5).
- Proxy Settings:** Hostname/IP, Port (0), User.
- CURL Options:** A list of options (CAINFO, CAPATH, CLOSEPOLICY, CONNECTTIMEOUT, COOKIE, COOKIEFILE, COOKIEJAR, COOKIELIST, CRLF, CUSTOMREQUEST, DNSCACHETIMEOUT) with right and left arrows.
- Soap Options:** Embedded Password [%P], Embed Value [%1], Embed Value [%2], Embed Value [%3], Embed Value [%4].
- HTTP Headers:** + Add a header.

At the bottom are 'Save' and 'Save As' buttons.

3. Complete the following fields:
  - **Profile Name.** Type a new name for the ServiceNow credential.
  - **Content Encoding.** Make sure *text/xml* is selected.
  - **Method.** Make sure *POST* is selected.
  - **HTTP Version.** Select *HTTP/1.1*.
  - **URL.** Type the URL for your PowerFlow instance.
  - **HTTP Auth User.** Type the username of your PowerFlow instance.
  - **HTTP Auth Password.** Type the password of your PowerFlow instance.
  - **Timeout.** Type "5".
4. Click **[Save As]**.
5. When the confirmation message appears, click **[OK]**.

- On the **Credential Management** page (System > Manage > Credentials), make a note of the value in the **ID** column for the credential you just created:

Credential Management   Credentials Found [1]												
Profile Name	Organization	RO Use	RW Use	DA Use	Type	Credential User	Host	Port	Timeout (ms)	ID	Last Edited	Edited By
1. ServiceNow RBA - Example	[all orgs]	--	--	--	SOAP/XML Host	USERNAME	INTEGRATIONSERVICEHOSTNAME	443	5000	107	2019-07-15 11:10:54	em7admin

You will use this value with the `s11_credential_id` parameter when you *edit the snippet code of the "ServiceNow: Add/Update/Clear" Run Book Action Policy*:

**Action Editor**

Policy Editor | Editing Action [455] Reset

Action Name: ServiceNow: Add/Update/Clear Incident

Action State: [ Disabled ]

Description: Adds and Updates Incidents in ServiceNow.

Organization: [ System ]

Action Type: ServiceNow: Create, Update, Clear Incident or Event (1.1)

Execution Environment: [ -- Default: ServiceNow Base Pack ]

Action Run Context: [ Database ]

Input Parameters

```
{
  "s11_credential_id": "107",
  "debug": false,
  "configuration": "",
  "queue": "",
  "discard_if_no_ci": false,
  "is_app_name_new": "incident_sync update_create",
  "is_app_name_ack": "event_acknowledged",
  "is_app_name_clear": "event_cleared",
  "servicenow_state_new": "1",
  "servicenow_state_ack": "",
  "servicenow_state_clear": "6",
  "correlation_type": "5",
  "assignment_group_new": "",
  "assignment_group_ack": ""
}
```

Save Save As

# Enabling the Run Book Automation Policies

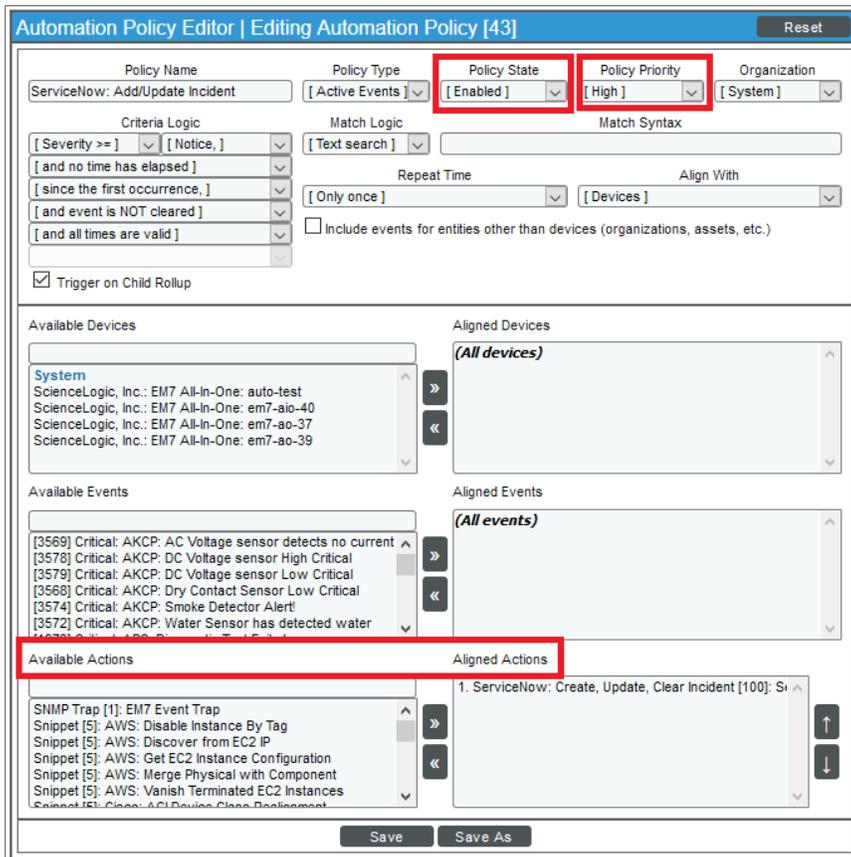
Before you can run the "ServiceNow: Add/Update/Clear Incident" Run Book Action, you must enable the relevant Run Book Automation policies in SL1 :

- ServiceNow: [(Cases, Events, or Incident)] - Add/Update
- ServiceNow: [(Events or Incident)] - Event Acknowledged
- ServiceNow: [(Cases, Events, or Incident)] - Event Cleared

**NOTE:** Versions 104 and 105 of the ServiceNow Base Pack PowerPack separated these Run Book Action policies by Cases, Events, and Incident, such as "ServiceNow: **[Events]** - Add/Update" and "ServiceNow: **[Cases]** - Add/Update". This update lets you choose the Run Book Actions based on if you are using the Cases, Events, or Incident Synchronization PowerPack.

To enable the three ServiceNow Run Book Automation policies:

1. In SL1 , go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Locate the "ServiceNow: [(Cases, Events, or Incident)] - Add/Update" automation policy and click its wrench icon (🔧). The **Automation Policy Editor** page appears:



3. Update the following fields:

- **Policy State.** Select *Enabled*.
- **Policy Priority.** Select *High* to ensure that this PowerFlow automation policy is added to the top of the queue.
- **Available Actions.** If it is not already selected, select the corresponding ServiceNow Run Book Action policy.

**TIP:** By default, the "ServiceNow: [Incidents] Add/Update" automation policy will create ServiceNow Incidents for **all** devices. You can limit the devices affected by making changes to the **Organization, Severity, Match Logic, Aligned Devices, and/or Aligned Events** fields.

**WARNING:** ScienceLogic highly recommends that you do not make changes to the **Policy Type, Repeat Time, or Align With** fields or the *And event is NOT acknowledged* setting.

4. Click **[Save]**.

5. Repeat steps 2-4 for the "ServiceNow: [(Events or Incident)] - Event Acknowledged" and "ServiceNow: [(Cases, Events, or Incident)] - Event Cleared" Run Book Automation policies.

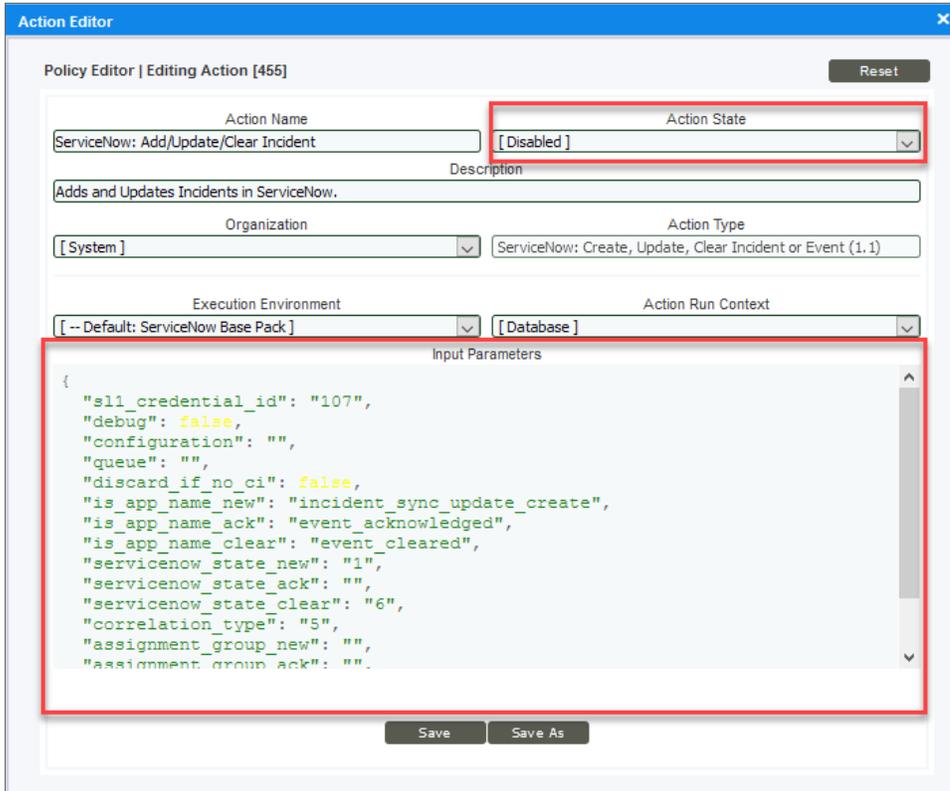
---

## Enabling and Customizing the Run Book Action Policy

The "ServiceNow: Add/Update/Clear Case", the "ServiceNow: Add/Update/Clear Event", and the "ServiceNow: Add/Update/Clear Incident" Run Book Action policies contain snippet code that you can customize to use with the relevant Synchronization PowerPack. You edit these values in the **Input Parameters** pane of the **Action Policy Editor** page for this policy.

To enable and customize the Run Book Action policy:

1. In SL1, go to the **Action Policy Manager** page (Registry > Run Book > Actions).
2. Locate the **ServiceNow: Add/Update/Clear** policy that you want to use and click its wrench icon (🔧). The **Action Policy Editor** page appears:



3. For the **Action State** field select *Enabled*.
4. For the **s11\_credential\_id** field in the **Input Parameters** pane, specify the credential ID from the **ID** column on the **Credential Management** page (System > Manage > Credentials). For example: "s11\_credential\_id": "107"
5. Edit the snippet code as necessary, using the information in the **Customizing the Snippet Code in the Input Parameters Pane** section, below. When you are finished, click [Save].

## Customizing the Snippet Code in the Input Parameters Pane

SL1 Run Book Action snippets are written in Python. In the event of a syntax error, the policies will no longer run. As a result, you must ensure that all edits adhere to Python standards. True and False options are case-sensitive and must not contain quotes.

**NOTE:** Previous SyncServer users had three separate Run Book Action scripts for add/update, acknowledge, and clear. These have been rolled into a single Run Book Action in this Synchronization PowerPack, but there are still three Automation policies.

You can customize the following values in the "ServiceNow: Add/Update/Clear" Run Book Action snippet code:

- **sl1\_credential\_id**. Specifies the ID of the credential object. You can find this value in the **ID** column of the **Credentials** page (System > Manage > Credentials of SL1). For example: "sl1\_credential\_id": "107"
- **debug**. A true/false value that determines if the action is logged in SL1 and if the application is run in Debug Mode on PowerFlow. Troubleshooting logs are written to `/data/tmp/servicenow_rba.log`.
- **configuration**. Specifies the ID of the configuration object used on PowerFlow. The configuration ID is all lower-case, with spaces in the configuration object "friendly" name replaced by underscores. For example: "configuration": "servicenow\_syncpack\_configs"

**NOTE:** To find the configuration ID with the API, make a GET request on this endpoint:  
`https://<powerflow_hostname>/api/v1/configurations.`

- **queue**. Specifies the worker queue on which the application runs. Leave this as default.
- **discard\_if\_no\_ci**. Specifies whether PowerFlow should create cases, events, or incidents in ServiceNow for devices that do not have a matching CI record. The default is *false*. Your options include:
  - *true*. If a device is not mapped to a CI, PowerFlow will *not* create a case, event, or incident in ServiceNow, and SL1 is not updated. The following log message appears: "No CI found".
  - *false*. If a device is not mapped to a CI, PowerFlow *will* create a case, event, or incident in ServiceNow and update SL1.

**NOTE:** The following parameters are specific to the "ServiceNow: Add/Update/Clear Incident" Run Book Action.

- **servicenow\_state\_new**:
  - 1. Incident state is "New". This is the default value.
  - 2. Incident state is "In Progress".
  - 3. Incident state is "On Hold".
  - 6. Incident state is "Resolved".
  - 7. Incident state is "Closed".
  - 8. Incident state is "Canceled".

- **servicenow\_state\_ack:**

- 1. Incident state is "New". There is no default value.
- 2. Incident state is "In Progress".
- 3. Incident state is "On Hold".
- 6. Incident state is "Resolved".
- 7. Incident state is "Closed".
- 8. Incident state is "Canceled".

- **servicenow\_state\_clear:**

- 1. Incident state is "New".
- 2. Incident state is "In Progress".
- 3. Incident state is "On Hold".
- 6. Incident state is "Resolved". This is the default value.
- 7. Incident state is "Closed".
- 8. Incident state is "Canceled".

- **correlation\_type.** Correlates an SL1 Event with a ServiceNow Incident. For this value, which is also called the "Correlation ID", you can choose a number of different ways in which an incident can be created. The default setting is "correlation\_type": "5", which means that this Run Book Action correlates all events by device ID and event policy ID, and if the event matches and the state is active, the Action updates the existing incident. The Action creates a *new* incident if the event does *not* match by device ID and event policy ID. As a result, the Correlation Type helps determine which events get rolled up under an incident. All Run Book Action scripts should use the same Correlation Type value, otherwise correlation will fail.

The possible Correlation Type values include the following:

- 1. Correlate all duplicate incidents by SL1 ID only.
- 2. Correlate all duplicate incidents by event policy ID only.
- 3. Correlate all duplicate incidents by device ID only.
- 4. Correlate all duplicate incidents by Interface ID only. This correlation requires that the SL1 event has an interface aligned. If there is no interface aligned to the event, the returned Interface ID will be 0.
- 5. Correlate all duplicate incidents by device ID and event policy ID. This is the default Correlation Type value.
- 6. Correlate all duplicate incidents by device ID, event policy ID, and event sub entity ID.
- 7. Correlate all duplicate incidents by device ID, event policy ID, region, and event ID. Please note that using this correlation type will result in all events triggering new incidents.
- 8. Correlate all events to the root device ID of the component device into one incident. Please note that using this correlation type might lead to undesirable behavior, as not all events generated on a child device indicate a problem on the root device. As a result, ScienceLogic recommends that you only use this correlation type against specific event policies.

- You can assign the assignment group to one of the new, acknowledged, or cleared incidents that are mapped. To disable this feature, ensure that no values are set. To assign an assignment group, set the variable value to the **sys\_id** of the ServiceNow Assignment Group. In the following example, the assignment group is assigned to incidents that are *cleared*:

```
"assignment_group_new": "",  
"assignment_group_ack": "",  
"assignment_group_clear": "sys_id"
```

## Customizing Logging in the Run Book Action

You can customize the following logging-related items in the "ServiceNow: Add/Update/Clear" Run Book Action snippet code:

- `logfile = /data/tmp/ServiceNow_add_update_clear_incident.log`
  - Location for logging output.
  - Will be created if it does not exist.
  - Will be appended with each Run Book job.
  - Is case-sensitive.
- `do_debug_logging = True`
  - True is on, False is off.
  - Is case-sensitive.
  - For troubleshooting, these can be enabled or changed.
  - Writes logs to **/data/tmp/servicenow\_rba.log**.

---

## Sending Custom Data to ServiceNow Using the Passthrough Option

You can use the "ServiceNow: [Incident] Add/Update" Run Book Automation and the "ServiceNow: Add/Update/Clear Incident" Run Book Action to "pass through" custom data about incidents to ServiceNow. For example, you might want to use the passthrough functionality to overwrite the impact and urgency of a ServiceNow Incident, which is the only way to change the priority of the Incident.

**NOTE:** You can also use the "ServiceNow: [Cases] Add/Update" Run Book Automation and the "ServiceNow: Add/Update/Clear Cases" Run Book Action to "pass through" custom data about cases to ServiceNow. Similarly, you can use the "ServiceNow: [Events] Add/Update" Run Book Automation and the "ServiceNow: Add/Update/Clear Event" Run Book Action to "pass through" custom data about events to ServiceNow.

To pass custom data to ServiceNow:

- Create a new Run Book Action that pulls the relevant data and adds it to a dictionary called EM7\_RESULT.
- Add the new Run Book Action to the "ServiceNow: [(Cases, Events, or Incident)] Add/Update " Run Book Automation Policy, ahead of the "ServiceNow: Add/Update/Clear (Case, Event, or Incident)" Run Book Action so that the new Action runs first, and then is consumed by the ServiceNow Action.

## Passing Custom Data to ServiceNow

The following procedure describes how to configure the passthrough functionality, using the "ServiceNow: [Incident] Add/Update" Run Book Automation and the "ServiceNow: Add/Update/Clear Incident" Run Book Action as examples.

To pass custom data to ServiceNow:

1. In SL1, go to the **Action Policy Manager** page (Registry > Run Book > Actions) and click **[Create]** to create a new Run Book Action policy:

The screenshot shows the 'Policy Editor | Creating New Action' interface. It features a 'Reset' button in the top right corner. The form is divided into several sections:

- Action Name:** Example Passthrough EM7\_RESULT
- Action State:** [Enabled]
- Description:** Passthrough data using EM7\_RESULT dictionary
- Organization:** [System]
- Action Type:** Run a Snippet
- Snippet Credential:** (None)
- Action Run Context:** Database
- Execution Environment:** [--Default Environment]
- Snippet Code:** EM7\_RESULT = { "work\_notes": "This is a new note" }

A 'Save' button is located at the bottom center of the form.

2. Complete the following fields:
  - **Action Name.** Type a unique name for the Action.
  - **Action State.** Select *Enabled*.
  - **Action Type.** Select *Run a Snippet*.
  - **Execution Environment.** Select *ServiceNow Base Pack*.
  - Complete the other fields as needed.

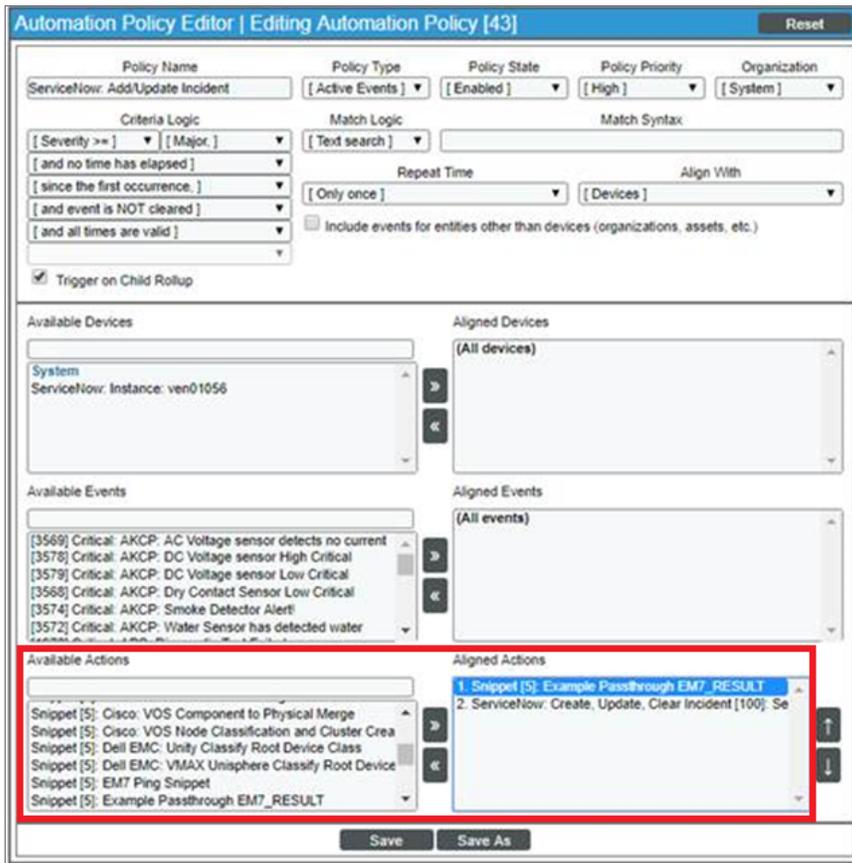
3. In the **Snippet Code** pane, add the snippet code you want to include for the EM7\_RESULT dictionary. For example, the following snippet code lets you override the ServiceNow Incident work notes with a hardcoded note:

```
EM7_RESULT = {"work_notes": "This is a new note"}
```

Additional notes about the structure of the EM7\_RESULT dictionary:

- EM7\_RESULT = is required for the dictionary, and the formatting of the keys should match the example above.
  - All keys defined in the EM7\_RESULT dictionary need to map to field IDs on the **ScienceLogic Events** table in ServiceNow.
  - You can hard-code the values in the EM7\_RESULT dictionary, or you can use variables and functions, like the "Snippet Code Example", below.
  - As a best practice, avoid sending null passthrough values to ServiceNow. If you must send 'null' or 'NULL' values to ServiceNow, pass through that value as an empty string, such as "location": "". Also, only pass through values that you need. For example, instead of sending {"location": "", "work\_notes": "stuff"}, simply send {"work\_notes": "stuff"}.
  - A long snippet might delay the ticket being created
4. Click **[Save]**.
  5. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation) and open the "ServiceNow: Add/Update Incident" Run Book Automation Policy.

- In the **Available Actions** section, add the new Run Book Action *before* the "ServiceNow: Create, Update, Clear Incident" Run Book Action:



**NOTE:** The output of this new Run Book Action will be consumed by the "ServiceNow: Create, Update, Clear Incident" Run Book Action, ensuring that the EM7\_RESULT dictionary is passed through to ServiceNow. The "ServiceNow: Create, Update, Clear Incident" Run Book Action automatically populates the passthrough values with any values from EM7\_LAST\_RESULT. The passthrough overwrites any other previously defined fields, such as assignment group.

- You can add additional Run Book Actions to the Run Book Automation Policy for any additional workflows that you might want to run. The Automation Policy execute these Actions in a sequential, top-down order. However, the "ServiceNow: Create, Update, Clear Incident" Run Book Action only consumes the EM7\_RESULT dictionary from the Run Book Action directly above it.

## Snippet Code Example

The following snippet code example shows how to pull additional information and make it available for passthrough. All of the additional information that is going to be sent is contained in a dictionary variable called EM7\_RESULT. You can pass through multiple items through in a single Run Book Action by adding additional keys to the EM7\_RESULT dictionary.

This example lets you assign assignment groups to an Incident based on certain criteria, such as event policy IDs:

```
from future.utils import iteritems

def invert_mappings(mappings):
    """
    Invert received one-to-many mappings and converts it into a one-to-one
    mapping.

    Args:
        mappings (dict): Dictionary of mapped values

    Returns:
        dict: inverted dictionary.

    """
    inverted_mappings = dict()
    for key, values in iteritems(mappings):
        for sub_value in values:
            invert_mappings[sub_value] = key
    return inverted_mappings

# Example of assignment group to list of event policy ids mapping.
assignment_groups_to_event_policies = {
    "sys_id_1": [1, 2, 3, 4, 5],
    "sys_id_2": [6, 7, 8, 9, 10],
}
# which sys_id to use if the current event_policy_id isn't mapped
default_sys_id = "sys_id_3"

# invert the mappings
event_policy_to_assignment_group = invert_mappings(assignment_groups_to_event_
policies)

# Send assignment group sys_id to IS RBA
EM7_RESULT = {
    "assignment_group": event_policy_to_assignment_group.get(
        EM7_VALUES["%3"], default_sys_id
    )
}
```

---

## Configuring the "ServiceNow: Click to Create Incident" Automation Policy

The "ServiceNow: [Cases, Events, or Incident] Click to Create" Run Book Automation policy lets you manually create a case, event, or incident in ServiceNow by clicking the **Actions** button (☰) in SL1 for an event and selecting "Create External Ticket" (or by clicking the life-preserver icon (🛑) for an event in the classic user interface).

This Run Book Automation policy is available in the *ServiceNow Base Pack PowerPack*.

**NOTE:** Versions 104 and 105 of the *ServiceNow Base Pack PowerPack* separated this Run Book Automation policy into the following: "ServiceNow: [Cases] Click to Create", "ServiceNow: [Events] Click to Create", and "ServiceNow: [Incident] Click to Create".

To configure the "ServiceNow: Click to Create" Run Book Automation policy:

1. In SL1, go to the Behavior Settings page (System > Settings > Behavior) and set the **Event Console Ticket Life Ring Button Behavior** option to *Create/View External Ticket*.
2. Click **[Save]** to save your changes. You might need to log out of SL1 and log back into SL1 for the changes to update.
3. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
4. Locate the **ServiceNow: [(Cases, Events, or Incident)] Click to Create** policy and click its wrench icon (🔧). The **Automation Policy Editor** page appears:

The screenshot shows the "Automation Policy Editor" interface for editing a policy named "ServiceNow: [Cases] - Click to Create". The policy is currently set to "Active Events" type, "Enabled" state, "Default" priority, and "System" organization. The criteria logic is set to "[Severity >=] [Minor,] [and no time has elapsed] [since the first occurrence,] [and event is NOT cleared] [and all times are valid]". The match logic is "[Text search]". The repeat time is "Every 1 minute until satisfied". The policy is aligned with "[Devices]".

The "Available Actions" list includes:

- SNMP Trap [1]: SL1 Event Trap
- Snippet [5]: AWS: Disable Instance By Tag
- Snippet [5]: AWS: Discover from EC2 IP
- Snippet [5]: AWS: Get EC2 Instance Configuration
- Snippet [5]: AWS: Merge Physical with Component

The "Aligned Actions" list contains one action: "1. ServiceNow: Create, Update, Clear Ticket [101]: Ser".

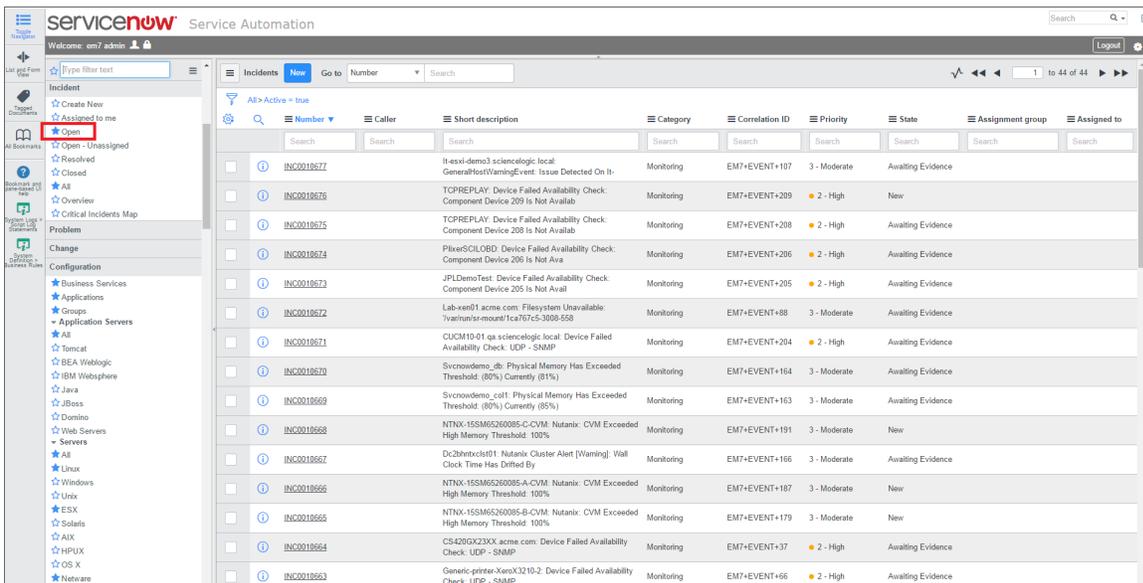
5. Update the following fields:

- **Policy State.** Select *Enabled*.
- **Repeat Time.** Specify the frequency at which SL1 should execute the automation policy while the conditions are still met. The choices range from "every 30 seconds until satisfied" to "every 2 hours until satisfied", or "only once". By default, the policy only runs once.
- **Available Actions.** If it is not already selected, select the corresponding ServiceNow Run Book Action policy to add it to the **Aligned Actions** field.

6. Click [Save].

## Viewing Incidents and Events

To view SL1-created incidents in ServiceNow, go to the **Incidents** page (Incident > Open) in ServiceNow:



	Number	Caller	Short description	Category	Correlation ID	Priority	State	Assignment group	Assigned to
	INC0019677		tnaxidemo3.sciencetologic.local: GeneralHostWarningEvent: Issue Detected On It-	Monitoring	EM7+EVENT+107	3 - Moderate	Awaiting Evidence		
	INC0019676		TCPREPLAY: Device Failed Availability Check: Component Device 209 Is Not Availab-	Monitoring	EM7+EVENT+209	2 - High	New		
	INC0019675		TCPREPLAY: Device Failed Availability Check: Component Device 208 Is Not Availab-	Monitoring	EM7+EVENT+208	2 - High	Awaiting Evidence		
	INC0019674		PilverSIOLOB: Device Failed Availability Check: Component Device 206 Is Not Ava-	Monitoring	EM7+EVENT+206	2 - High	Awaiting Evidence		
	INC0019673		JPLDemoTest: Device Failed Availability Check: Component Device 205 Is Not Avail	Monitoring	EM7+EVENT+205	2 - High	Awaiting Evidence		
	INC0019672		Lab-ven01.acme.com: Filesystem Unavailable: \waintr-srv\mount\tsa707c5-3008-558	Monitoring	EM7+EVENT+88	3 - Moderate	Awaiting Evidence		
	INC0019671		CLCM10-01.qa.sciencetologic.local: Device Failed Availability Check: LDAP - SNMP	Monitoring	EM7+EVENT+204	2 - High	Awaiting Evidence		
	INC0019670		Srvcsnodedemo_db: Physical Memory Has Exceeded Threshold: (80%) Currently (81%)	Monitoring	EM7+EVENT+164	3 - Moderate	Awaiting Evidence		
	INC0019669		Srvcsnodedemo_c01: Physical Memory Has Exceeded Threshold: (80%) Currently (85%)	Monitoring	EM7+EVENT+163	3 - Moderate	Awaiting Evidence		
	INC0019668		NTNX-155M65260085-C-CVM: Nutanix: CVM Exceeded High Memory Threshold: 100%	Monitoring	EM7+EVENT+191	3 - Moderate	New		
	INC0019667		Dc2hntxcls01: Nutanix Cluster Alert [Warning]: Wall Clock Time Has Drifted By	Monitoring	EM7+EVENT+166	3 - Moderate	Awaiting Evidence		
	INC0019666		NTNX-155M65260085-A-CVM: Nutanix: CVM Exceeded High Memory Threshold: 100%	Monitoring	EM7+EVENT+187	3 - Moderate	New		
	INC0019665		NTNX-155M65260085-B-CVM: Nutanix: CVM Exceeded High Memory Threshold: 100%	Monitoring	EM7+EVENT+179	3 - Moderate	New		
	INC0019664		CS-0200X230X.acme.com: Device Failed Availability Check: UDP - SNMP	Monitoring	EM7+EVENT+37	2 - High	Awaiting Evidence		
	INC0019663		Generic-printer-Xerox1310-2: Device Failed Availability Check: UDP - SNMP	Monitoring	EM7+EVENT+66	2 - High	Awaiting Evidence		

All SL1 incidents use the event message from the SL1 **Events** page as the incident description in ServiceNow. The SL1 **Events** page also links to the Incident record if you click the life-preserver icon (🔗) under the **External Ticket** column. Note that you might need to enable the **External Ticket** column by modifying the Console preferences.

SL1 and ServiceNow use slightly different methods for designating the severity or priority of an event or incident. A custom data lookup table handles the conversion by translating the SL1 event severity into ServiceNow **Impact**, **Urgency**, and **Priority** fields. This severity data lookup table for automatically deploys with the "ScienceLogic SL1: CMDB & Incident Automation Application", also called the Scoped or Certified Application:

Severity	Impact	Urgency
1	1	2
2	2	2
3	2	3
4	3	3

If a second event for the same incident occurs, that event will update the **Impact** and **Urgency** values, based on the severity lookup table.

When the SL1 Run Book Automation creates a ServiceNow incident, the action taken depends on the **Correlation Type** you specified in the "ServiceNow Add/Update/Clear Incident" Run Book Action. If one or more events are aligned to the same incident, those events will show up in the **[Events]** tabs at the bottom of the Incident record in ServiceNow. All event-specific information for an incident is listed on this page:

Incident = INC0013377	Action policy	Active	Automation policy	Correlation	Device	Event count	Event created	Event ID	Event policy	Hyperlink
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 09:23:16	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117652264g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117652264g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 08:28:17	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117655784g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117655784g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 08:18:16	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117654964g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117654964g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 07:33:17	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117652064g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117652064g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 07:13:16	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117650564g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117650564g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	
ServiceNow: Add/Update /Clear Incident	false	ServiceNow: Add/Update Incident	fsundemo89+DEV+2197+EVENT+1705	(empty)	1	2019-09-19 07:03:18	●	Poller: Network Latency Exceeded Threshold	<a href="http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117649864g_sev=1&amp;g_sort=0&amp;g_oper=0">http://em7.mydomain.com/em7/index.em?nwec-event&amp;g_type=aid&amp;g_arg=117649864g_sev=1&amp;g_sort=0&amp;g_oper=0</a>	

**TIP:** If the **[Events]** tab does not display at the bottom of the Incident record, you can add it by opening the Incident record, clicking the Additional actions icon (☰), and selecting **Configure > Related Lists**. Add **Event > Incident** to the **Selected** column and click **[Save]**.

For more information about ServiceNow incident management, see [https://docs.servicenow.com/bundle/newyork-it-service-management/page/product/incident-management/concept/c\\_IncidentManagement.html](https://docs.servicenow.com/bundle/newyork-it-service-management/page/product/incident-management/concept/c_IncidentManagement.html).

# Incident Topology Suppression

**Incident topology suppression** is used when ServiceNow incidents that have been synced with SL1 devices occur on devices that have a parent/child relationship. If you choose to enable incident topology suppression in SL1, child events synced with ServiceNow incidents do not appear in the SL1 **Event Console** as separate events. Instead, the child events are nested under the parent event.

**NOTE:** The steps in this process use the Classic user interface for SL1.

To enable incident topology suppression:

1. In SL1, navigate to the **Event Policy Manager** page (Registry > Events > Event Manager) and click the **[Create]** button. The **Event Policy Editor** modal appears:

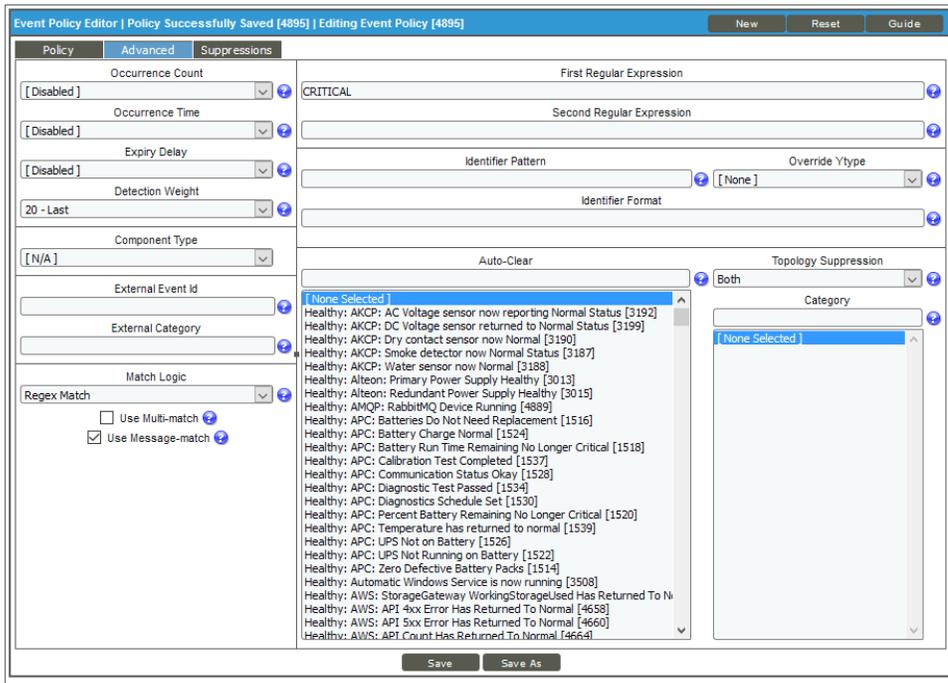
The screenshot shows the 'Event Policy Editor' modal with the following configuration:

- Event Source:** API
- Operational State:** Enabled
- Event Severity:** Critical
- Policy Name:** Topology Suppress
- Event Message:** Child events were suppressed for this device.

2. On the **[Policy]** tab, update the following fields:

- **Event Source:** Select *API*.
- **Operational State:** Select *Enabled*.
- **Event Severity:** Select *Critical* as the severity of the event.
- **Policy Name.** Type the name of the event. Can be any combination of alphanumeric characters, up to 48 characters in length
- **Event Message.** Type the message that will appear when this event occurs.

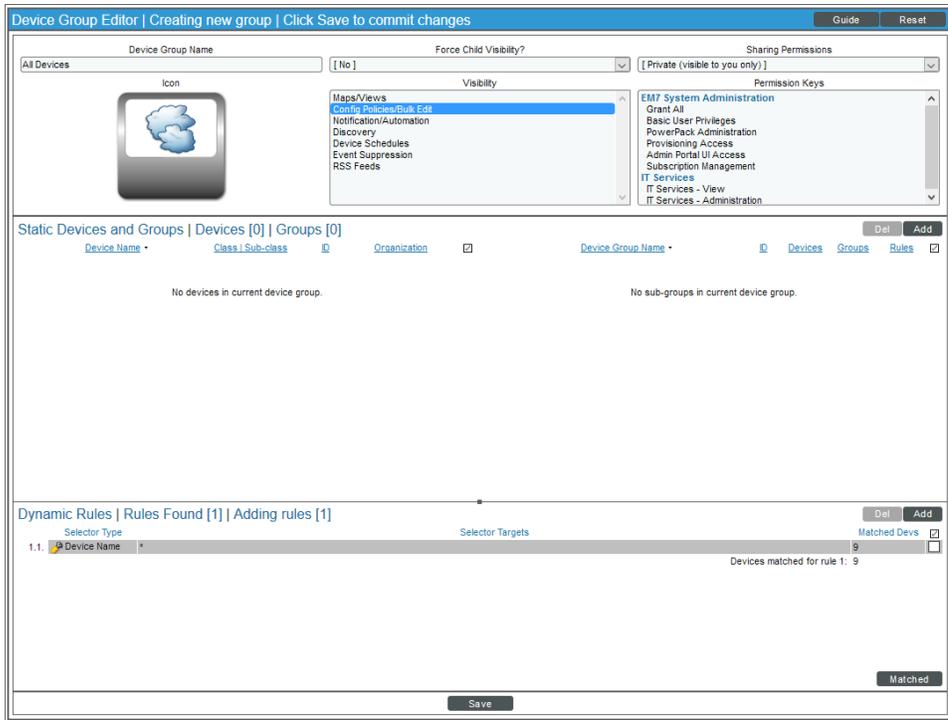
3. Click the **[Advanced]** tab.



4. On the **[Advanced]** tab, update the following fields:

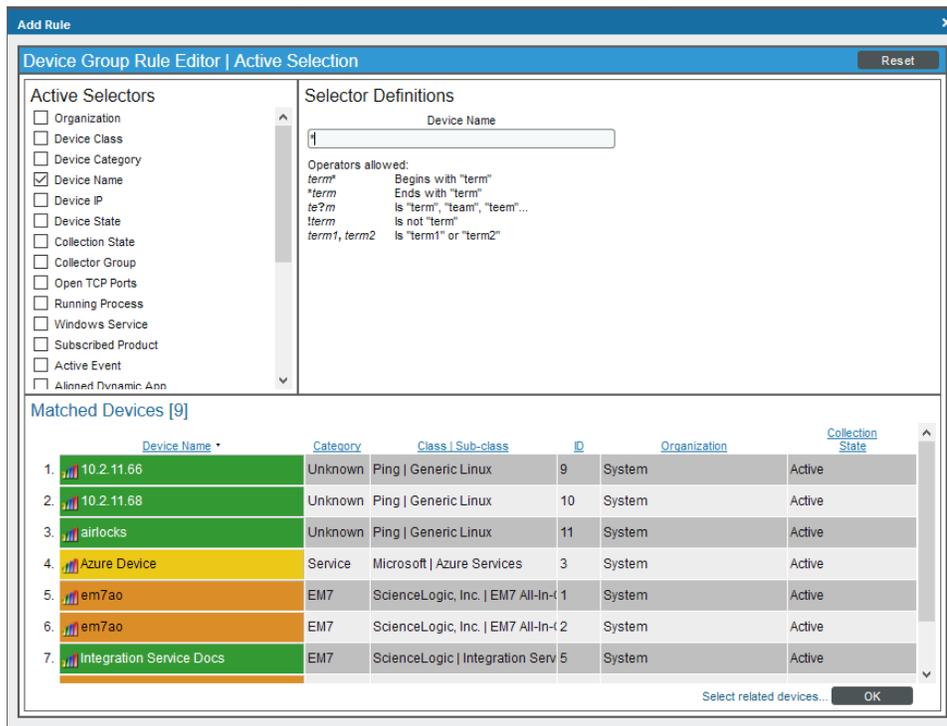
- **Detection Weight.** Select *20 - Last*. If two event definitions are very similar, the weight field specifies the order in which SL1 should match messages against the similar event definitions. The event definition with the lowest weight will be matched first. This field is most useful for events that use expression matching. Options range from 0 (first) - 20 (last).
- **Match Logic.** Select *Regex Match*. Specifies whether SL1 should process the First Match String field and Second Match String as regular expressions or as simple text matches. Because you selected *Regex Match*, you cannot define a "match all" expression by leaving the First Match String and Second Match String fields empty.
- **Use Message-match.** Select this option. If SL1 has generated an event and then a second log message or alert matches the same event policy for the same entity, SL1 will not generate a second event, but will increase the count value for the original event. This behavior will occur only if the log messages or alerts contain the same message.
- **First Regular Expression.** Type "CRITICAL" as the string used to correlate the event with a log message.
- **Topology Suppression.** Select *Both*. If this event occurs on a parent device, it behaves as a suppressing event. If this event occurs on a child device, it behaves as a suppressible event.

5. Click **[Save]** and close the **Event Policy Editor** modal.
6. Next, go to the **Device Groups** page (Registry > Device Groups) and click the **[Create]** button. A **Device Group Editor** page appears:



7. Complete the following fields, and leave the default settings for the remaining fields:
  - **Template Name.** Specify the name of the new device group.
  - **Force Child Visibility.** Select "No".
  - **Visibility.** Select *Config Policies/Bulk Edit* to let you configure all the devices in the new device group using a device template.

- Click the **[Save]** button and then click the **[Add]** button in the **Dynamic Rules** pane to add dynamic rules to the new device group. The Device Group Rule Editor modal page appears:



- In the **Active Selectors** pane, select *Device Name*.
- Optionally, in the **Selector Definitions** pane, type an asterisk (\*) in the **Device Name** field. Using the \* includes all devices by Device Name. In the **Matched Devices** pane, a list of all devices appears.
- Click **[OK]** to close the modal page.
- On the **Device Group Editor** modal page, click **[Save]** and close the page.

13. Next, create a Device Group Template that will disable Event Masking for all devices in the new Device Group. Click the building blocks icon (🧱) for the new device group. A **Device Template Editor** page appears:

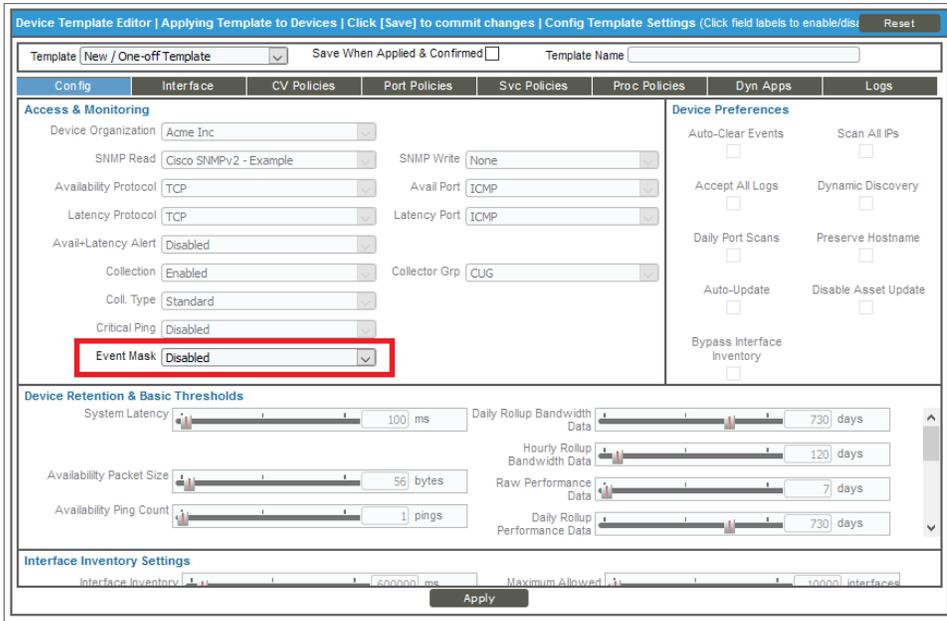
The screenshot displays the 'Device Template Editor' interface. At the top, there is a header bar with the text 'Device Template Editor | Applying Template to Devices | Click [Save] to commit changes | Config Template Settings (Click field labels to enable/dis) | Reset'. Below the header, there is a navigation bar with tabs for 'Config', 'Interface', 'CV Policies', 'Port Policies', 'Svc Policies', 'Proc Policies', 'Dyn Apps', and 'Logs'. The 'Config' tab is selected.

The main content area is divided into several sections:

- Access & Monitoring:** This section contains various configuration options. The 'Event Mask' dropdown menu is highlighted with a red box and is set to 'Disabled'. Other options include Device Organization (Acme Inc), SNMP Read (Cisco SNMPv2 - Example), SNMP Write (None), Availability Protocol (TCP), Avail Port (ICMP), Latency Protocol (TCP), Latency Port (ICMP), Avail+Latency Alert (Disabled), Collection (Enabled), Collector Grp (CUG), Coll. Type (Standard), and Critical Ping (Disabled).
- Device Preferences:** This section contains several checkboxes: Auto-Clear Events, Scan All IPs, Accept All Logs, Dynamic Discovery, Daily Port Scans, Preserve Hostname, Auto-Update, Disable Asset Update, and Bypass Interface Inventory.
- Device Retention & Basic Thresholds:** This section contains several sliders and input fields for data retention and thresholds. For example, System Latency is set to 100 ms, Daily Rollup Bandwidth Data is set to 730 days, Hourly Rollup Bandwidth Data is set to 120 days, Raw Performance Data is set to 7 days, and Daily Rollup Performance Data is set to 730 days.
- Interface Inventory Settings:** This section contains input fields for Interface Inventory (set to 60000) and Maximum Allowed (set to 10000).

An 'Apply' button is located at the bottom center of the interface.

14. Because all of the fields are disabled (grayed-out) by default, click the **Event Mask** field name to enable the field. Use the default setting of *Disabled*.
15. Click **[Apply]** and click **[Confirm]** on the **Device Template Editor** page.
16. Next, turn off the *Trigger on Child Rollup* option on the "ServiceNow: Add/Update Incident" Run Book Automation. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation) and click the wrench icon (🔧) for the "ServiceNow: Add/Update Incident" Run Book Automation. The **Automation Policy Editor** page appears:



17. Make sure the *Trigger on Child Rollup* option is not selected and click **[Save]**. Close the **Automation Policy Editor** page.

## Hyperlinking Events

Both ServiceNow and SL1 provide mechanisms for hyperlinking to multiple active events and incidents. This section describes those processes.

## ServiceNow Hyperlinking

Each incident in ServiceNow will have one or more events aligned with it through the Related List of events, which you can view in the **[Events]** tab at the bottom of the Incident record.

By default the **Hyperlink** field appears on the custom event table provided with the Certified application (`x_sclo_scilogic_event`). Each event shown in the **[Events]** tab at the bottom of the Incident record will have its own URL associated. The following image shows the Event record for an event aligned with an Incident:

The screenshot shows the ServiceNow Event record form for Event ID 315. The form is titled "Event 315" and includes a navigation bar with "Update" and "Delete" buttons. The form fields are organized into two columns. The left column includes Incident (INC0010020), Region (region6341), Event ID (315), Event policy (Dynamic App Snippet Exception), Automation policy (ServiceNow: Click to Create Incident), Action policy (ServiceNow: Add/Update/Clear Inciden), and Message (em7-ao-47: App: 90, Snippet: 110 reported a collection problem (Explanation: SNMP error returned: Timeout. Location: Using walkbulk to query .1.3.6.1.2.1.25.2.3.1.2)). The right column includes Device (em7-ao-47), Organization (System), Severity (Minor), Event created (2019-10-30 13:00:19), Last detected (2019-10-30 13:00:19), and Event count (1). The Hyperlink field is highlighted with a red box and contains the URL: [http://em7.mydomain.com/em7/index.em7?exec=events&q\\_type=aid&q\\_arg=315&q\\_sev=1&q\\_sort=0&q\\_oper=0](http://em7.mydomain.com/em7/index.em7?exec=events&q_type=aid&q_arg=315&q_sev=1&q_sort=0&q_oper=0). The form also includes "Update" and "Delete" buttons at the bottom left and a refresh icon at the bottom right.

**TIP:** If the **[Events]** tab does not display at the bottom of the Incident record, you can add it by opening the Incident record, clicking the Additional actions icon (☰), and selecting *Configure > Related Lists*. Add *Event->Incident* to the **Selected** column and click **[Save]**.

## Viewing the Incident Import Table in ServiceNow

Each time SL1 creates or changes an incident in ServiceNow, data is inserted into a temporary import table on the ServiceNow system. This table displays all inbound data from SL1 and is a useful tool to determine what data is being sent and imported. The incident import table is created automatically when you install the ScienceLogic Certified (Scoped) Application.

To view the data and the status of the import process, go to the **Import Incidents** page (ScienceLogic > Event > Events) in ServiceNow:

Event ID	Created	Correlation ID	Incident	CMDB CI	Updated	Target record	State	Incident state	Created by	Import set run
1175595	2019-09-19 09:33:21	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:33:21	Event_1175595	Updated	Resolved	fun	(empty)
1175595	2019-09-19 09:33:21	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:33:21	Incident_INC0013315	Updated	Resolved	fun	(empty)
1175595	2019-09-19 09:28:27	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:28:27	Event_1175595	Inserted	In Progress	fun	(empty)
1175595	2019-09-19 09:28:27	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:28:27	Incident_INC0013315	Updated	In Progress	fun	(empty)
1175592	2019-09-19 09:28:26	fuadem08@DEV-2197-EVENT+1705	INC0013317	(empty)	2019-09-19 09:28:26	Incident_INC0013317	Updated	Resolved	fun	(empty)
1175592	2019-09-19 09:28:26	fuadem08@DEV-2197-EVENT+1705	INC0013317	(empty)	2019-09-19 09:28:26	Event_1175592	Updated	Resolved	fun	(empty)
1175592	2019-09-19 09:23:22	fuadem08@DEV-2197-EVENT+1705	INC0013317	(empty)	2019-09-19 09:23:22	Incident_INC0013317	Updated	In Progress	fun	(empty)
1175592	2019-09-19 09:23:22	fuadem08@DEV-2197-EVENT+1705	INC0013317	(empty)	2019-09-19 09:23:22	Event_1175592	Updated	In Progress	fun	(empty)
1175577	2019-09-19 09:03:34	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:03:34	Event_1175577	Updated	Resolved	fun	(empty)
1175577	2019-09-19 09:03:34	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 09:03:34	Incident_INC0013315	Updated	Resolved	fun	(empty)
1175577	2019-09-19 08:58:30	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:58:30	Incident_INC0013315	Updated	In Progress	fun	(empty)
1175577	2019-09-19 08:58:30	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:58:30	Event_1175577	Inserted	In Progress	fun	(empty)
1175579	2019-09-19 08:53:35	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:53:35	Incident_INC0013315	Updated	Resolved	fun	(empty)
1175579	2019-09-19 08:53:35	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:53:35	Event_1175579	Updated	Resolved	fun	(empty)
1175579	2019-09-19 08:48:31	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:48:31	Incident_INC0013315	Updated	In Progress	fun	(empty)
1175579	2019-09-19 08:48:31	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:48:31	Event_1175579	Inserted	In Progress	fun	(empty)
1175556	2019-09-19 08:38:32	fuadem08@DEV-2196-EVENT+1705	INC0013315	(empty)	2019-09-19 08:38:32	Incident_INC0013315	Updated	Resolved	fun	(empty)

You can view a complete audit of all import data and transforms by going to the **Transform Histories** page (System Import Sets > Advanced > Transform History):

Started	State	Completed	Run time	Set	Import set table	Total	Inserts	Updates	Ignored	Skipped	Errors	Transform Map
2016-09-04 04:00:50	Complete	2016-09-04 04:00:50	0 Seconds	ISET0013291	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-09-01 12:00:27	Complete	2016-09-01 12:00:27	0 Seconds	ISET0013275	ScienceLogic Network Interfaces [u_sciencelogic_adapters]	1	0	0	1	0	0	ScienceLogic Adapter T-Map
2016-09-02 14:00:12	Complete	2016-09-02 14:00:12	0 Seconds	ISET0013278	ScienceLogic Hardware Models [u_sciencelogic_hardware_models]	1	0	0	1	0	0	ScienceLogic Hardware Model T-Map
2016-09-01 16:00:44	Complete	2016-09-01 16:00:44	0 Seconds	ISET0013276	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-08-31 18:01:16	Complete	2016-08-31 18:01:16	0 Seconds	ISET0013271	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-09-02 03:00:28	Complete	2016-09-02 03:00:28	0 Seconds	ISET0013280	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-09-01 20:01:03	Complete	2016-09-01 20:01:03	0 Seconds	ISET0013276	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-09-03 02:01:22	Complete	2016-09-03 02:01:22	0 Seconds	ISET0013286	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map
2016-09-01 12:00:59	Complete	2016-09-01 12:00:59	0 Seconds	ISET0013275	ScienceLogic Network Interfaces [u_sciencelogic_adapters]	1	0	0	1	0	0	ScienceLogic Adapter T-Map
2016-09-03 12:00:12	Complete	2016-09-03 12:00:12	0 Seconds	ISET0013283	ScienceLogic Hardware Models [u_sciencelogic_hardware_models]	1	0	0	1	0	0	ScienceLogic Hardware Model T-Map
2016-09-04 22:00:33	Complete	2016-09-04 22:00:33	0 Seconds	ISET0013290	ScienceLogic Network Interfaces [u_sciencelogic_adapters]	1	0	0	1	0	0	ScienceLogic Adapter T-Map
2016-09-01 16:01:17	Complete	2016-09-01 16:01:17	0 Seconds	ISET0013276	ScienceLogic File System [u_sciencelogic_file_system]	1	0	0	1	0	0	ScienceLogic File System T-Map

## ServiceNow, ScienceLogic Event, and Incident Priority Matrix

By default, when SL1 triggers an event, it is sent to ServiceNow through PowerFlow. The following mappings are currently in place for mapping an SL1 Event to a ServiceNow Incident (ScienceLogic > Event > Severity Lookup Rules):

Severity	Impact	Urgency
1	1	2
2	2	2
3	2	3
4	3	3

**NOTE:** This severity lookup table handles all Severity conversions.

A **transformation script** that translates the SL1 event severity into the ServiceNow **Impact**, **Urgency**, and **Priority** fields automatically deploys with the ScienceLogic Certified (Scoped) Application.

By default, the **Priority** field is read-only and must be set by selecting the **Impact** and **Urgency** values.

Priority is calculated according to the following data lookup rules:

Impact	Urgency	Priority
1 - High	1 - High	1 - Critical
1 - High	2 - Medium	2 - High
1 - High	3 - Low	3 - Moderate
2 - Medium	1 - High	2 - High
2 - Medium	2 - Medium	3 - Moderate
2 - Medium	3 - Low	4 - Low
3 - Low	1 - High	3 - Moderate
3 - Low	2 - Medium	4 - Low
3 - Low	3 - Low	5 - Planning

# Adding Additional Fields to the Transform Map

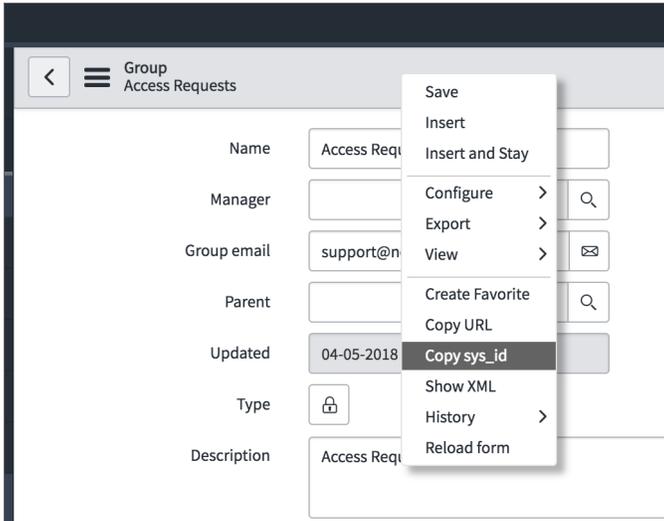
If you require additional mandatory fields to be filled out to resolve an incident, you can add those fields to the **transform map** in ServiceNow.

For example, if you require four mandatory fields in the ServiceNow Incident—**Assignment Group**, **IT Service**, **Service Component**, and **Description**—to be filled out before that incident can be resolved in SL1, you would perform the following steps.

The screenshot displays the 'Main Details' section of a ServiceNow Incident form. The form is organized into two columns of fields. The left column includes fields for Caller (Science Logic), Behalf of user, Company (Motorpoint Limited), Location, IT service, Service component, Symptom, Configuration item, and Short description. The right column includes Contact type (web), Impact (Some Users), Urgency (3 - Low), Priority (4 - Low), Major incident state, Owing group, Assignment group, and Assigned to. A 'Related Search Results' section is located below the main form fields. The fields for IT service, Service component, Assignment group, and Description are highlighted with red rectangular boxes, indicating they are mandatory for resolution.

To add an assignment group:

1. Navigate to **User Administration > Groups** and select the assignment group you want to add. The Group record appears.
2. Right-click the gray task bar at the top and select **Copy sys\_id**.



3. In SL1, open to the "ServiceNow: Add/Update/Clear Incident" Run Book Action (Registry > Run Book > Actions).
4. Edit the **Input Parameters** of the Run Book Action to add the **sys\_id** to the relevant parameter or parameters to assign the assignment group to one of the new, acknowledged, or cleared incidents that are mapped. After an incident is created, the assignment group value will not be changed by the Run Book Action.

In the following example, the assignment group is assigned to incidents that are *cleared*:

```
"assignment_group_new": "",  
"assignment_group_ack": "",  
"assignment_group_clear": "sys_id"
```

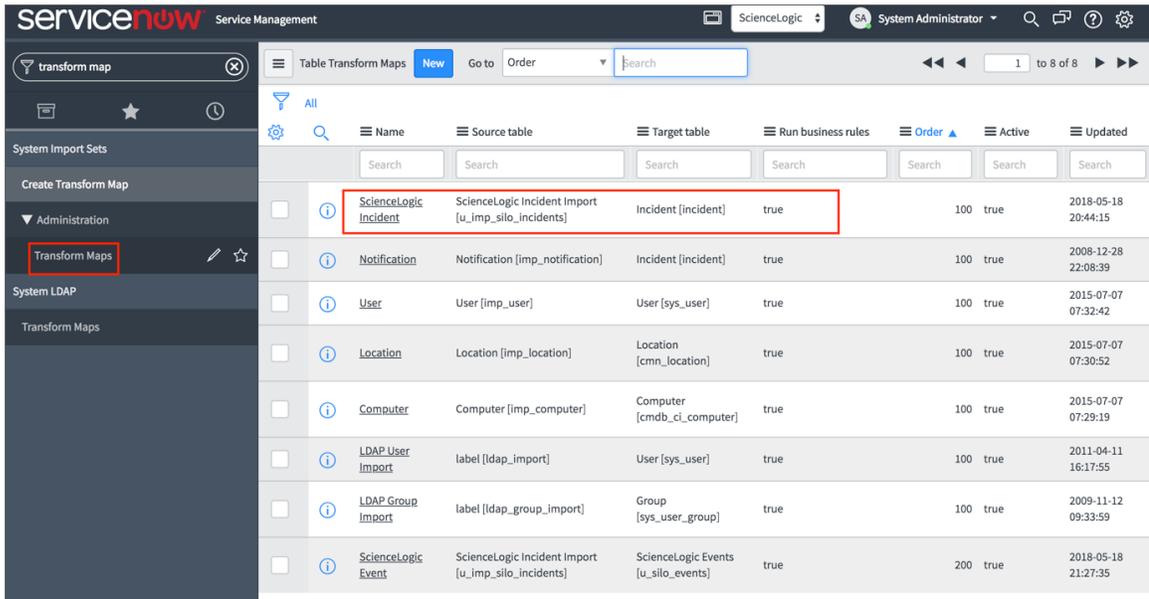
The **IT Service**, **Service Component**, and **Description** fields in our example must be filled in before an Incident can be closed. To do this, changes must be made in the transform maps that are provided in the form of update sets from ScienceLogic.

**TIP:** For more information about mapping new fields and other mappings options, see [https://docs.servicenow.com/bundle/newyork-platform-administration/page/script/server-scripting/concept/c\\_MappingOptions.html](https://docs.servicenow.com/bundle/newyork-platform-administration/page/script/server-scripting/concept/c_MappingOptions.html).

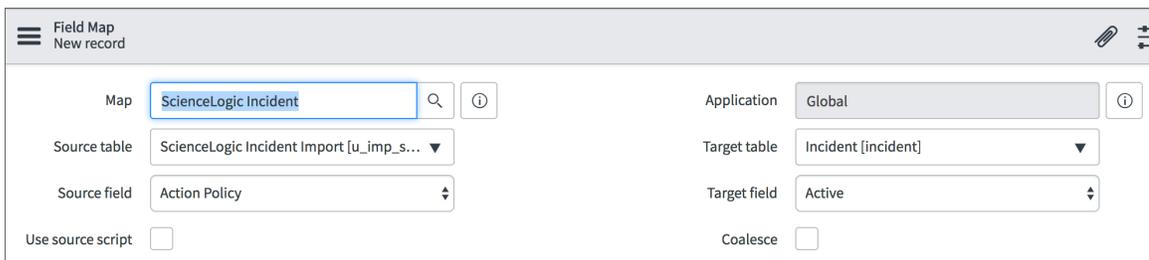
To add the **Description** field:

1. In ServiceNow, search for "transform map" in the filter navigator. Click **Transform Maps**.
2. In the list of transform maps, search for "ScienceLogic" in the field above the **Name** column.

- Open the "ScienceLogic Incident" map:



- The **Field Maps** table at the bottom of the page allows you to edit or create mappings from the ScienceLogic Incident Import table to the ServiceNow Incident table. Click **[New]** to create a new field mapping.
- The **Source table** field should contain the ScienceLogic Incident Import and the **Target table** should include the ServiceNow Incident table:



- To create a new mapping to copy the contents of the **Short description** field to the **Description** field, select **Short description** from the **Source field** drop-down menu.
- In the **Target field** drop-down menu, select **Description**.
- Click **Update** to save your changes.

The **IT Service** and **Service Component** fields in our example are set in the Transform Script in the "ScienceLogic Event" transform map. To set the fields:

- In ServiceNow, make sure you have the **sys\_id** value for the target fields. If a field contains a magnifying glass, it will require a **sys\_id**. If a field has a drop-down, type in the field you wish to apply from the drop-down. In the case of our example, the **sys\_id** values of the two fields are required.
- In your ServiceNow instance, navigate to the **Transform Maps** table and select "ScienceLogic Event".

3. In the ScienceLogic Event transform map page, click the **[Transform Script]** tab and open the "onAfter" script.

The screenshot shows the configuration page for a Table Transform Map named "ScienceLogic Event". The "Transform Scripts" tab is selected, displaying a table of scripts. The "onAfter" script is highlighted.

When	Script	Order	Active
onAfter	(function runTransformScript(source, map...	100	true
onBefore	(function runTransformScript(source, map...	100	true

4. Add the following under the "//Update target record when the Event was cleared from ScienceLogic" text:

```
sl_INT.(target field) = '[sys_id of the source field]'; //(IT service field)
sl_INT.(target field) = '[sys_id of the source field]'; //(Service component)
```

When: onAfter

Application: ScienceLogic

Active:

Order: 100

Script:

```
1 (function runTransformScript(source, map, log, target /*undefined onStart*/ ) {
2
3   event_state_management();
4
5   function event_state_management(){
6     //Check ScienceLogic Events to see, if the Incident can be closed.
7     var sl_events = new GlideRecord('u_silo_events');
8     sl_events.addEncodedQuery('u_active=true^u_silo_task=' + source.u_task);
9     sl_events.query();
10    if(sl_events.getRowCount() == 0){
11      var sl_INT = new GlideRecord('incident');
12      sl_INT.get(target.u_silo_task);
13      if(source.u_incident_state == 6 && sl_INT.close_code.nil()){
14        //Update target record when the Event was cleared from ScienceLogic.
15        sl_INT.close_code='Closed/Resolved by Caller';
16        sl_INT.close_notes=source.u_work_notes;
17        sl_INT.state= 6;
18      }else if(source.u_incident_state ==6 && !sl_INT.close_code.nil()){
19        //Update target record when Event was cleared From ServiceNow.
20        sl_INT.close_notes= sl_INT.close_notes + " (" + source.u_work_notes + ")";
21      }
22      sl_INT.update();
23    }
24    return;
25  }
26
27 }(source, map, log, target);
```

- To find the target field, make a temporary mapping to see what the target field is. This mapping can be deleted once you know the target field.

	Source field	Target field
<input type="checkbox"/>	<a href="#">u_short_description</a>	shprt_description
<input type="checkbox"/>	<a href="#">u_contact_type</a>	contact_type
<input type="checkbox"/>	<a href="#">u_active</a>	active
<input type="checkbox"/>	<a href="#">u_short_description</a>	description
<input type="checkbox"/>	<a href="#">u_assignment_group</a>	assignment_group
<input type="checkbox"/>	<a href="#">u_correlation_id</a>	correlation_id
<input type="checkbox"/>	<a href="#">sys_updated_by</a>	caller_id
<input type="checkbox"/>	<a href="#">u_urgency</a>	urgency
<input type="checkbox"/>	<a href="#">u_cmdb_ci</a>	cmdb_ci
<input type="checkbox"/>	<a href="#">u_work_notes</a>	work_notes
<input checked="" type="checkbox"/>	<a href="#">u_impact</a>	<b>u_service_component</b>
<input type="checkbox"/>	<a href="#">u_impact</a>	impact
<input type="checkbox"/>	[Script]	company
<input type="checkbox"/>	[Script]	location

- Click **[Update]** to save your changes. The selected fields will be added into an Incident on closure.

---

# Chapter

# 4

## Troubleshooting the Incident Synchronization PowerPack

---

### Overview

This chapter contains troubleshooting resources, procedures, and the answers to frequently asked questions to use with the *ServiceNow Incident Synchronization PowerPack*.

This chapter covers the following topics:

<i>Initial Troubleshooting Steps</i> .....	67
<i>Resources for Troubleshooting</i> .....	67
<i>Frequently Asked Questions</i> .....	72

---

## Initial Troubleshooting Steps

PowerFlow acts as a middle server between data platforms. For this reason, the first steps should always be to ensure that there are no issues with the data platforms with which PowerFlow is talking. There might be additional configurations or actions enabled on ServiceNow or SL1 that result in unexpected behavior. For detailed information about how to perform the steps below, see [Resources for Troubleshooting](#).

### SL1 PowerFlow

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

### ServiceNow

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

---

## Resources for Troubleshooting

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

### Useful PowerFlow Ports

- **http://<IP of PowerFlow>:8081**. Provides access to Docker Visualizer, a visualizer for Docker Swarm.
- **https://<IP of PowerFlow>:8091**. Provides access to Couchbase, a NoSQL database for storage and data retrieval.
- **https://<IP of PowerFlow>:15672**. Provides access to the RabbitMQ Dashboard, which you can use to monitor the service that distributes tasks to be executed by PowerFlow workers. Use `guest/guest` for the login.
- **https://<IP of PowerFlow>/flower**. Provides access to Flower, a tool for monitoring and administrating Celery clusters.

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

## Helpful Docker Commands

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

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

### Viewing Container Versions and Status

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

```
docker service ls
```

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

```
[root@sunislab ~]# docker service ls
ID                NAME                MODE                REPLICAS                IMAGE                PORTS
ommihu35v301     iservices_gui        replicated           1/1                     repository.auto.sciencelogic.local:5000/is-gui:1.7.0      *:80->80/tcp, *:443->443/tcp
40v99it1wvh3     iservices_redis      replicated           1/1                     redis:4.0.2
j1m6h1jyumif     iservices_flowerv    replicated           1/1                     repository.auto.sciencelogic.local:5000/is-worker:1.7.0   *:5555->5555/tcp
hh9p2l9lrsf      iservices_scheduler replicated           1/1                     repository.auto.sciencelogic.local:5000/is-worker:1.7.0   *:5000->5000/tcp
htimlcv96kxh     iservices_contentapi replicated           1/1                     repository.auto.sciencelogic.local:5000/is-api:1.7.0
cyin9qgsudmi     iservices_rabbitmq   replicated           1/1                     rabbitmq:3
klui9h8jfs6      iservices_visual     replicated           2/1                     dockersamples/visualizer:latest                          *:8081->8080/tcp
jcy38w8buauw     iservices_couchbase  replicated           1/1                     repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0 *:8091->8091/tcp, *:8092->8092/tcp
p->8093/tcp, *:8094->8094/tcp, *:11210->11210/tcp
libxwxt7uf       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 PowerFlow Services

Run the following command to stop all PowerFlow services:

```
docker stack rm iservices
```

### Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

**NOTE:** Restarting Docker does not clear the queue.

## Viewing Logs for a Specific Service

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

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

Some common examples include the following:

```
docker service logs -f iservices_couchbase
docker service logs -f iservices_steprunner
docker service logs -f iservices_contentapi
```

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

## Clearing RabbitMQ Volume

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

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

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

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

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

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

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

Re-deploy PowerFlow by running the following command:

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

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

## Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

## Deploying Services from a Defined Docker Compose File

Run the following command:

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

## Dynamically Scaling for More Workers

Run the following command:

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

## Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

## Diagnosis Tools

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

- **Docker PowerPack.** This PowerPack monitors your Linux-based PowerFlow server with SSH (the PowerFlow ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your PowerFlow server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor PowerFlow, see the "Using SL 1 to Monitor SL 1 PowerFlow" chapter in the *SL 1 PowerFlow Platform* manual.
- **Flower.** This web interface tool can be found at the /flower endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the PowerFlow platform. This tool also shows detailed information about each of the tasks that have been executed on the platform. This data includes the UUID, the state, the arguments that were passed to it, as well as the worker and the time of execution. Flower also provides a performance chart that shows the number of tasks running on each individual worker.
- **Debug Mode.** All applications can be run in "debug" mode via the PowerFlow API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see [Retrieving Additional Debug Information](#).
- **Application Logs.** All applications generate a log file specific to that application. These log files can be found at /var/log/iservices and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.
- **Step Logs.** Step logs display the log output for a specific step in the application. These step logs can be accessed via the PowerFlow user interface by clicking on a step in an application and bringing up the **Step Log** tab. These step logs display just the log output for the latest run of that step.

- **Service Logs.** Each Docker service has its own log. These can be accessed via SSH by running the following command:  
`docker service logs -f <service_name>`

## Retrieving Additional Debug Information (Debug Mode)

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

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

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

To run an application in Debug Mode, POST the following to the API endpoint:

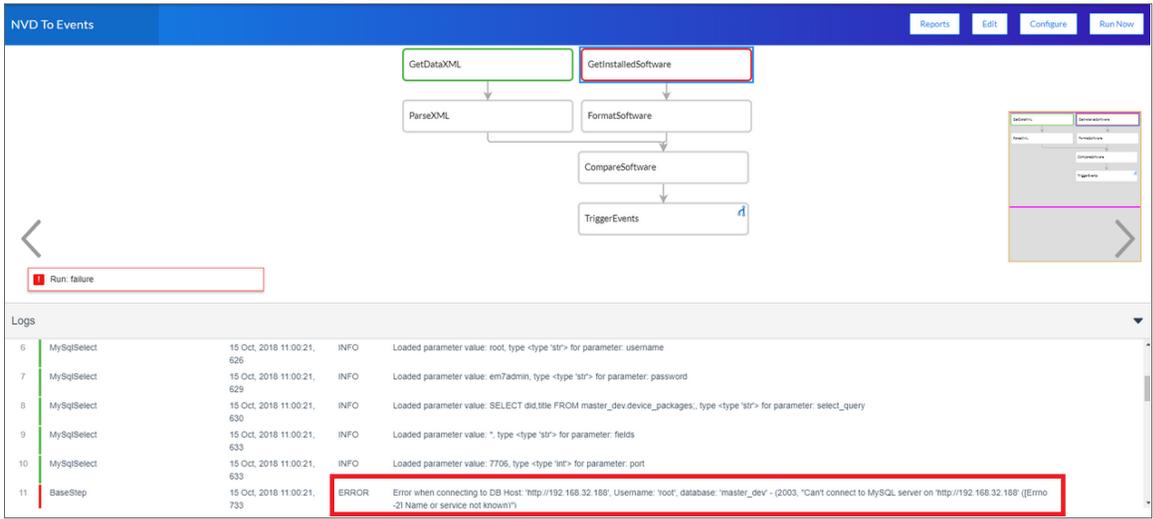
`https://<PowerFlow>/api/v1/applications/run`

Request body:

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

After running the application in Debug Mode, go back to the PowerFlow user interface and review the step logs to see detailed debug output for each step in the 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 application or step failed:



The screenshot displays the 'NVD To Events' interface. At the top, there are buttons for 'Reports', 'Edit', 'Configure', and 'Run Now'. Below this is a workflow diagram with the following steps: 'GetDataXML' (green box), 'GetInstalledSoftware' (red box), 'ParseXML', 'FormatSoftware', 'CompareSoftware', and 'TriggerEvents'. A 'Run failure' indicator is shown on the left. The 'Logs' section at the bottom contains the following data:

Step	Step Name	Timestamp	Level	Message
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 application in debug using curl via SSH:

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

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

## Frequently Asked Questions

This section contains a set of frequently asked questions (FAQs) and the answers to address those situations.



## What if my Incident does not have a CI?

For an incident with an active event:

1. In SL1, go to the **[Events Console]** (classic user interface) or the **Events** page (new user interface) and locate the event that was created.
  - In the SL1 user interface, click the **[Actions]** button ( **\*\*\*** ) for that event and select *View Automation Actions*.
  - In the Classic user interface, click the **View Notification Log** mailbox icon () for that event.

The **Event Actions Log** window appears

2. On the **Event Actions Log** window, locate the PowerFlow run ID, which is the first six or seven characters of the "isapp" integration ID associated with that run of the integration. For example: *isapp-a7805e58*.
3. In the PowerFlow user interface, go to the **Applications** page and open the application that used that run.
4. Review the Step Log and confirm that the device class was mapped in the "Sync Devices from SL1 to ServiceNow" application.
5. Confirm that the "Sync Devices from SL1 to ServiceNow" application is running at least every 24 hours, and that the "Sync Devices from SL1 to ServiceNow" application has run within 24 hours of that event sync run.

## What if the PowerFlow user interface is unresponsive and Incidents are not being generated in ServiceNow?

If the PowerFlow user interface is unresponsive, and Incidents are not being generated in ServiceNow, this might mean that during the deployment process, a change to the firewall rules for monitoring broke the ingress network for Docker.

To address this issue, run the following command to restart Docker every time you make a firewall or network configuration change:

```
systemctl restart docker
```

## Why are Incident numbers not populated in SL1 on Incident creation in ServiceNow?

If an incident exists in ServiceNow, but incident data is not getting back to SL1, and the "Sync ServiceNow Incident State to SL1 Event" application fails on the "Get Incident" step (with a 404 error) and eventually times out, the issue might be because the ServiceNow API is overloaded.

## Why am I not getting any Incidents after disabling the firewall?

If you disabled the firewall to enable SNMP monitoring on PowerFlow, but were not able to connect, you should add the additional rule you need.

---

# Appendix

# A

## Checklists for Deployment

---

### Overview

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

This appendix includes the following topics:

<i>CMDB-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow</i> .....	76
<i>CMDB-Only ServiceNow Integration with Single SL1 and Domain-Separated ServiceNow</i> .....	80
<i>CMDB-Only ServiceNow Integration with Multiple SL1 Systems, no Domain-Separated ServiceNow</i> .	84
<i>Incident-Only ServiceNow Integration with Single SL1, no Domain Separation in ServiceNow</i> .....	88

---

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

1. Deploy the PowerFlow 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 PowerFlow.
5. Create the PowerFlow 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 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 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 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 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 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 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 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 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 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.

**NOTE:** The maintenance schedule does not support changing the planned start date and planned end date after a linked Maintenance Schedule has been created in SL1.

## Sync Advanced Topology from SL1 to ServiceNow

1. Align the configuration object to the following 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 PowerFlow 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 PowerFlow.
5. Create the PowerFlow configuration object using the "ServiceNow SyncPack" configuration object as a template.

## Sync Devices from SL1 to ServiceNow

1. Align the configuration object to following 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 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 application should be run at least every 23 hours.

## Discover Devices from ServiceNow in SL1

1. Align Configuration Object to following 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 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 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 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 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 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 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 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 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 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 PowerFlow 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:

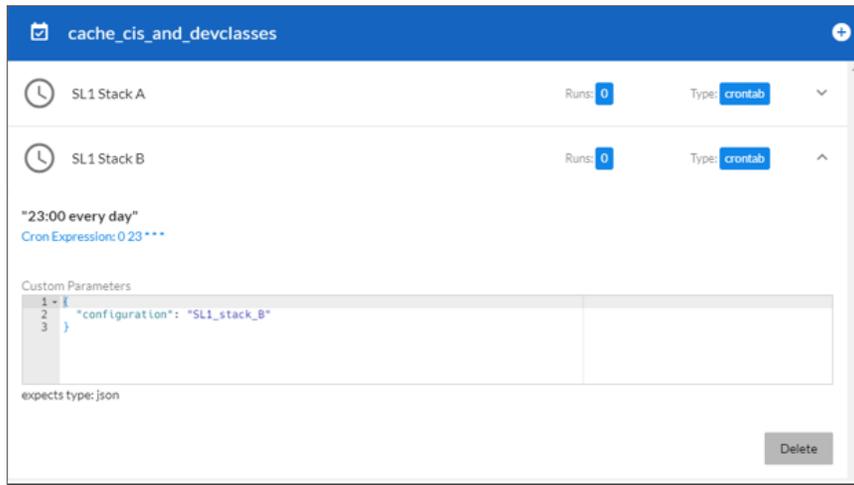
- Username
- Password
- Web Service Access Only
- GMT Time Zone
- x\_sclo\_scilogic.Admin role assigned

4. Install the ServiceNow Synchronization PowerPacks on PowerFlow.

5. Create the PowerFlow configuration object using the "ServiceNow SyncPack" configuration object as a template.

**NOTE:** The key difference between integrating a single SL1 stack on PowerFlow and integrating multiple SL1 stacks on PowerFlow is how you run the application. Running the 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:



## Sync Devices from SL1 to ServiceNow

1. Create a schedule for each SL1 stack for the following Applications 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 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 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 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 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 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 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 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 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 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 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 PowerFlow 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 PowerFlow.
4. Create the PowerFlow configuration object using the "ServiceNow SyncPack" configuration object as a template:
  - Align the configuration object to the following 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 PowerFlow 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:

<i>Roles</i> .....	90
<i>Tables</i> .....	91
<i>Table Columns (cmdb_ci)</i> .....	91
<i>Table Columns (core_company)</i> .....	92
<i>Table Columns (cmdb_group)</i> .....	92
<i>Script Includes</i> .....	92
<i>Event Registry</i> .....	92
<i>Scripted Actions</i> .....	93
<i>Data Lookup Definitions</i> .....	93
<i>System Properties</i> .....	93
<i>Catalog Item</i> .....	94
<i>Catalog UI Policies</i> .....	94
<i>Variable Sets</i> .....	94
<i>Catalog Client Scripts</i> .....	95
<i>Workflows</i> .....	95
<i>Scripted REST Resources</i> .....	96

## 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 BSMPProcessor. 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 PowerFlow
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_session_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.

Name		Comments	
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 ServiceNow Synchronization PowerPacks. These scripted endpoints reduce the amount of REST calls that PowerFlow 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:

<a href="#">Business Services</a> .....	101
<a href="#">Change Requests</a> .....	103
<a href="#">Classification version 1</a> .....	105
<a href="#">Classification version 2</a> .....	108
<a href="#">CMDB Group</a> .....	111
<a href="#">Companies</a> .....	113
<a href="#">Device Identification Engine</a> .....	116

<i>Discovery Dependents</i> .....	119
<i>File Systems</i> .....	121
<i>Import Set</i> .....	123
<i>Incidents</i> .....	125
<i>Installed Software</i> .....	128
<i>Manufacturer</i> .....	131
<i>Model</i> .....	133
<i>Network Adapters</i> .....	135
<i>Service Requests</i> .....	137

---

# 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](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 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 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": "d83dac0adb4dab00dc44f00fbf961919,2e6b7046db8dab00dc44f00fbf961929,7fb39667dba12380dc44f00fbf961901,77b39667dba12380dc44f00fbf961917,7bb39667dba12380dc44f00fbf96191c",
        "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 must 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.e17",
        "software": "671bafd8dba13700dc44f00fbf961953",
        "cldb_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/scienceologic/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"
}
]
}

```

© 2003 - 2021, ScienceLogic, Inc.

All rights reserved.

#### 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](mailto:legal@sciencelogic.com)



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

International: +1-703-354-1010