



---

# ServiceNow Cases SyncPack

Version 1.1.0

---

# Table of Contents

<b>Introduction to the ServiceNow Cases SyncPack</b>	<b>4</b>
What Can I Do with this SyncPack?	5
Contents of the SyncPack	6
PowerFlow Applications	6
Steps	6
<b>Installing the ServiceNow Cases SyncPack</b>	<b>7</b>
Architecture Overview for ServiceNow SyncPacks	8
Prerequisites for ServiceNow SyncPacks	8
SL1 and ServiceNow Terminology	9
Dependency Map for ServiceNow SyncPacks	9
Downloading the SyncPack	10
Importing the SyncPack	10
Installing the SyncPack	11
Installing the "ScienceLogic SL1: Customer Service Management Integration" Application in ServiceNow	12
For Case Integration ServiceNow Environments Only	13
Installing the ServiceNow Base Pack PowerPack in SL1	13
<b>Configuring Applications for the Cases SyncPack</b>	<b>15</b>
Workflow for Configuring the SyncPack	16
Configuring SL1	16
Configuring PowerFlow	16
Configuring SL1	17
Creating a ServiceNow Credential in SL1	17
Enabling the Run Book Automation Policies	18
Enabling and Customizing the Run Book Action Policy	18
Customizing the Snippet Code in the Input Parameters Pane	20
Customizing Logging in the Run Book Action	21
Sending Custom Data to ServiceNow Using the Passthrough Option	21
Passing Custom Data to ServiceNow	22
Passthrough Example	23
Snippet Code Example	24
Configuring the "ServiceNow: [Cases] Click to Create" Automation Policy	25

Enabling Run Book Automation Queue Retries .....	26
Configuring PowerFlow .....	26
Creating and Aligning a Configuration Object in PowerFlow .....	26
Creating a Configuration Object .....	27
Creating an OAuth2 Credential Record in ServiceNow .....	29
Aligning a Configuration Object and Configuring PowerFlow Applications .....	30
Scheduling PowerFlow Applications .....	32
<b>Troubleshooting the Cases SyncPack .....</b>	<b>33</b>
Initial Troubleshooting Steps .....	34
SL1 PowerFlow .....	34
ServiceNow .....	34
Resources for Troubleshooting .....	34
Useful PowerFlow Ports .....	34
Helpful Docker Commands .....	35
Viewing Container Versions and Status .....	35
Restarting a Service .....	35
Stopping all PowerFlow Services .....	35
Restarting Docker .....	35
Diagnosis Tools .....	36
Retrieving Additional Debug Information (Debug Mode) .....	36

---

# Chapter

# 1

## Introduction to the ServiceNow Cases SyncPack

---

### Overview

This chapter describes the "ServiceNow Cases" SyncPack, which lets you sync SL1 events with ServiceNow cases. This SyncPack uses the ServiceNow Customer Service Management (CSM) module.

This SyncPack uses the "ScienceLogic SL1: Customer Service Management Integration certified application in ServiceNow and the latest "ServiceNow Base Pack" PowerPack in SL1.

Do not use the "ServiceNow Cases" SyncPack and the "ServiceNow Events" or the "ServiceNow Incidents" SyncPacks on the same PowerFlow system.

This chapter covers the following topics:

<i>What Can I Do with this SyncPack?</i> .....	5
<i>Contents of the SyncPack</i> .....	6

---

## What Can I Do with this SyncPack?

The "ServiceNow Cases" SyncPack is the ScienceLogic integration with the ServiceNow Customer Service Management (CSM) module.

This SyncPack automatically logs, de-duplicates, correlates, updates, and appends ServiceNow incidents and cases, reducing the amount of time to resolve critical service issues. This SyncPack covers the entire incident and case life cycle, providing a bi-directional integration between SL1 events and ServiceNow incidents and cases, while providing a granular view into both the event and the associated record in ServiceNow.

With this SyncPack, any changes (except for acknowledgments) made to an SL1 event is sent to ServiceNow to update the corresponding case. Any status changes to the ServiceNow case are synced back to the corresponding SL1 event.

You can also use this SyncPack in conjunction with the "ServiceNow Configuration Management Database (CMDB)" SyncPack to create cases with a linked ServiceNow Configuration Item (CI).

For this SyncPack, you can configure a run book action policy to ensure that whenever SL1 detects a new, acknowledged, or cleared event, a corresponding case is created or updated in ServiceNow. These automations are included in the latest "ServiceNow Base Pack" PowerPack. For more information, see [Enabling the Run Book Automation Policies](#).

**WARNING:** Do not use the "ServiceNow Cases" SyncPack and the "ServiceNow Events" SyncPack on the same PowerFlow system. Also, do not use the Cases and Incident SyncPacks on the same PowerFlow system.

This SyncPack includes the following applications:

- **Create or Update ServiceNow Case from SL1 Event.** From a new or existing SL1 Event, transforms SL1 Event data into a new or existing ServiceNow Case and then posts the Event details to ServiceNow.
- **Resolve ServiceNow Case from SL1 Event.** Resolves a ServiceNow Case when an SL1 Event is cleared.
- **Sync Case or Incident State from ServiceNow to SL1 Event.** Clears or updates SL1 Events when the related ServiceNow Case or Incident is updated.

For more information, see [Configuring Applications for the Cases SyncPack](#).

---

# Contents of the SyncPack

This section lists the contents of the "ServiceNow Cases" SyncPack.

## PowerFlow Applications

The following PowerFlow applications are included with the "ServiceNow Cases" SyncPack:

- **Create or Update ServiceNow Case from SL1 Event.** From a new or existing SL1 Event, transforms SL1 Event data into a new or existing ServiceNow Case and then posts the Event details to ServiceNow.
- **Resolve ServiceNow Case from SL1 Event.** Resolves a ServiceNow Case when an SL1 Event is cleared.
- **Sync Case or Incident State from ServiceNow to SL1 Event.** Clears or updates SL1 Events when the related ServiceNow Case or Incident is updated.

## Steps

The following steps are included with the SyncPack:

- Process Cases and Incidents
- Transform ServiceNow Result Details to SL1 Event Data
- Process Event Data for Clear
- Process SL1 RBA for Incident of Case
- Pull Tickets from ServiceNow
- Post ServiceNow Ticket Details to SL1

---

# Chapter

# 2

## Installing the ServiceNow Cases SyncPack

---

### Overview

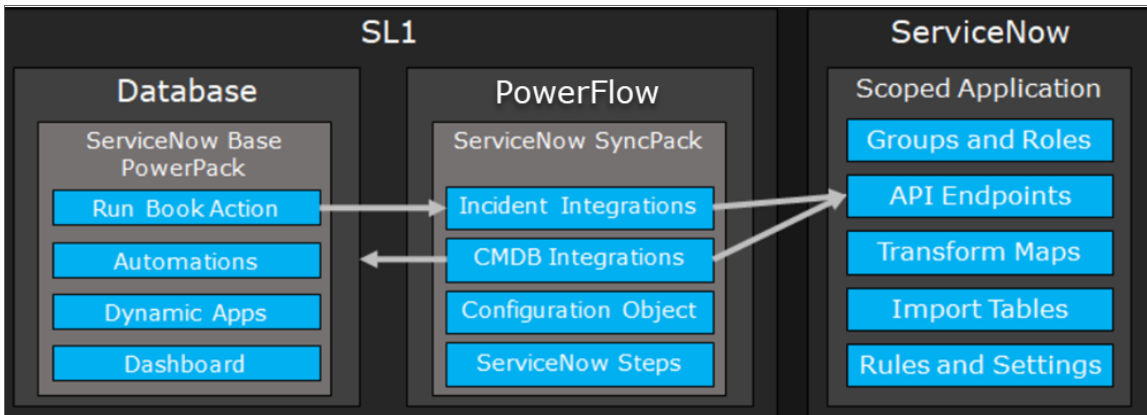
This chapter describes how to install the "ServiceNow Cases" SyncPack and the other applications needed to use the SyncPack, including the "ScienceLogic SL1: Customer Service Management Integration" application and the "ServiceNow Base Pack" PowerPack.

This chapter covers the following topics:

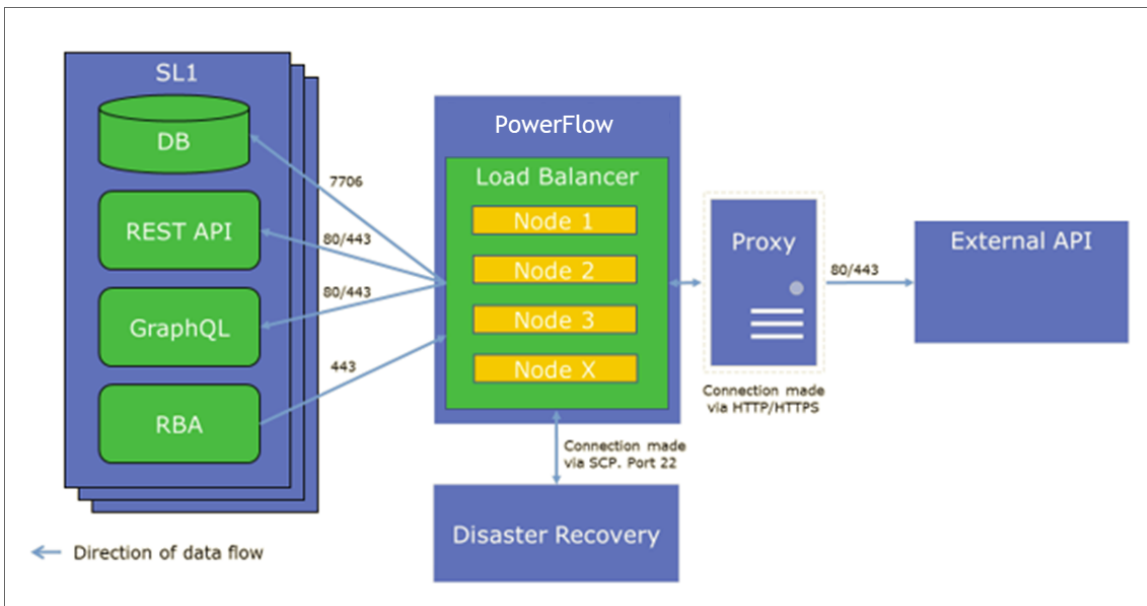
<i>Architecture Overview for ServiceNow SyncPacks</i> .....	8
<i>Prerequisites for ServiceNow SyncPacks</i> .....	8
<i>SL1 and ServiceNow Terminology</i> .....	9
<i>Dependency Map for ServiceNow SyncPacks</i> .....	9
<i>Downloading the SyncPack</i> .....	10
<i>Importing the SyncPack</i> .....	10
<i>Installing the SyncPack</i> .....	11
<i>Installing the "ScienceLogic SL1: Customer Service Management Integration" Application in ServiceNow</i> .....	12
<i>Installing the ServiceNow Base Pack PowerPack in SL1</i> .....	13

## Architecture Overview for ServiceNow SyncPacks

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:



## Prerequisites for ServiceNow SyncPacks

This section describes the prerequisites for the ServiceNow SyncPacks. For more information about the specific software versions required by a ServiceNow SyncPack, see the latest release notes for this SyncPack on the [SL 1 PowerFlow SyncPack Release Notes](#) page.



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

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

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

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

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

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

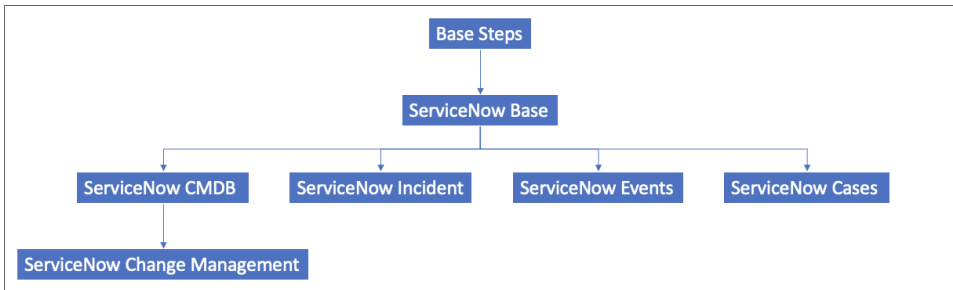
## SL1 and ServiceNow Terminology

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 SyncPack you are using)
Alert	Event
Organization	Company, Domain
Schedule, Maintenance Schedule	Change Request, Change Schedule
Topology, Relationships, Dynamic Component Mapping and Relationships (DCM+R)	Dependency View, Affected CIs

## Dependency Map for ServiceNow SyncPacks

The following graphic describes which SyncPacks depend on other SyncPacks:



**TIP:** For more information about the Standard *Base Steps* SyncPack, see the *SL1 PowerFlow Platform* manual.

## Downloading the SyncPack

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

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

To locate and download the SyncPack:

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

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

## Importing the SyncPack

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

To import a SyncPack in the 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 SyncPack you want to install. You can also drag and drop a **.whl** file to the **Import SyncPack** page.
3. Click **[Import]**. PowerFlow registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in PowerFlow. For more information, see the following topic.

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

---

## Installing the SyncPack

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


To activate and install a SyncPack in the PowerFlow user interface:

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

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

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

2. Click **[Yes]** to confirm the activation and installation. When the SyncPack is activated, the **SyncPacks** page displays a green check mark icon (✓) for that SyncPack. If the activation or installation failed, then a red exclamation mark icon (❗) appears.
3. For more information about the activation and installation process, click the check mark icon (✓) or the exclamation mark icon (❗) in the **Activated** column for that SyncPack. For a successful installation, the "Activate & Install SyncPack" application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

4. If you have other versions of the same SyncPack on your PowerFlow system, you can click the **[Actions]** button (  ) for that SyncPack and select *Change active version* to activate a different version other than the version that is currently running.

---

## Installing the "ScienceLogic SL1: Customer Service Management Integration" Application in ServiceNow

You must install the "ScienceLogic SL1: Customer Service Management Integration" application on the ServiceNow instance to enable this SyncPack. You must first request the "ScienceLogic SL1: Customer Service Management Integration" application from the ServiceNow Store, and then you can install it.

**NOTE:** If you want to sync CMDB data, such as Company information, you should also install the "ScienceLogic SL1: CMDB & Incident Automation" application version 1.0.45 or later.

**WARNING:** The "ScienceLogic SL1: Case Integration" application requires that you have already installed the ServiceNow "Customer Service Management" module on the target ServiceNow instance.

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

To request and install the "ScienceLogic SL1: Customer Service Management Integration" application:

1. Go to the ServiceNow Store at <https://store.servicenow.com> and search for "ScienceLogic SL1".
2. Select the "ScienceLogic SL1: Customer Service Management Integration" application. The detail page for the application appears.
3. Click **Get** and log in with your HI credentials.
4. After the request is approved, log in to ServiceNow as an administrator and navigate to **Application Manager** (System Applications > Applications).
5. Click **Downloads** in the menu header or search for "ScienceLogic".
6. Click the version drop-down for the "ScienceLogic SL1: Customer Service Management Integration" application listing to make sure you are using the correct version of the application that is compatible with your version of the ServiceNow Events SyncPack.
7. Click **Install** for the "ScienceLogic SL1: Customer Service Management 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 navigation menu in the left pane.

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

## For Case Integration ServiceNow Environments Only

Add the following roles to the Integration user so that user can interact with the "ScienceLogic SL1: Customer Service Management Integration" Application in ServiceNow.

- **x\_sclo\_case\_mgmt.admin.** Provides user rights to interact with the scoped application tables and modules in ServiceNow.
- **import\_transformer.** Provides user rights to manage import set transform maps, run transforms, and access responses.

If your ServiceNow environment is using the Case Integration module and you intend to use **customer\_account** records, you will need to add additional rights to the Integration user. These rights allow the Integration user to read the table fields:

- **sn\_customerservice.customer\_data\_viewer**

You will need to add cross-scoped access for read-only to the **customer\_account** table as well. ScienceLogic recommends that you use ServiceNow as the source of truth for Organizations (Companies). For more information, see the "Allowing Cross-Scoped Access in ServiceNow" topic in the **ServiceNow CMDB SyncPack** manual.

---

## Installing the ServiceNow Base Pack PowerPack in SL1

The *ServiceNow Base Pack PowerPack* version 105 or later contains Run Book Automations that you use to integrate with the *ServiceNow Cases SyncPack*.

**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 (Product Downloads > PowerPacks & SyncPacks) at <https://support.sciencelogic.com/s/>.
2. In SL1, go to the **PowerPacks** 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 **PowerPacks** page.

---

# Chapter

# 3

## Configuring Applications for the Cases SyncPack

---

### Overview

This chapter describes how to configure PowerFlow applications for the "ServiceNow Cases" SyncPack.

With this SyncPack, any changes (except for acknowledgments) made to an SL1 event is sent to ServiceNow to update the corresponding case. Any status changes to the ServiceNow case are synced back to the corresponding SL1 event.

You can also use this SyncPack in conjunction with the "ServiceNow Configuration Management Database (CMDB)" SyncPack to create cases with a linked ServiceNow Configuration Item (CI).

This chapter covers the following topics:

<i>Workflow for Configuring the SyncPack</i> .....	16
<i>Configuring SL1</i> .....	17
<i>Configuring PowerFlow</i> .....	26

---

# Workflow for Configuring the SyncPack

The following workflows describe how to configure SL1 and PowerFlow to work with the "ServiceNow Cases" SyncPack.

## Configuring SL1

1. [Create a ServiceNow credential](#)
2. [Enable the following run book automation policies:](#)
  - "ServiceNow: [Cases] Add/Update"
  - "ServiceNow: [Cases] Event Cleared"
3. [Enable and customize the "ServiceNow: Add/Update/Clear Case" run book action policy](#)
4. Optionally, [send custom data to ServiceNow using the passthrough option](#)
5. Optionally, [enable and configure the "ServiceNow: \[Cases\] Click to Create" policy](#)
6. Optionally, [enable run book automation queue retries](#)

## Configuring PowerFlow

1. [Create a configuration object in the PowerFlow user interface](#)
2. [Align the new configuration file with the following PowerFlow applications:](#)
  - "Create or Update ServiceNow Case from SL1 Event"
  - "Resolve ServiceNow Case from SL1 Event"
  - "Sync Case or Incident State from ServiceNow to SL1 Event"
3. [Schedule the PowerFlow applications as needed](#)



---

# Configuring SL1


The following topics cover how to set up your SL1 instance to work with the "ServiceNow Cases" SyncPack.

## 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 automations 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 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.
3. Complete the following fields:
  - **Profile Name**. Type a new name for the ServiceNow credential.
  - **Timeout**. Leave as the default of "5000" ms.
  - **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.
4. Click **[Save & Close]**. The credential is added to the **Credentials** page.
5. On the **Credentials** page, make a note of the value in the **ID** column for the credential you just created. You will use this value with the `sl1_credential_id` parameter when you [enable the snippet code of the "ServiceNow: Add/Update/Clear \(Case/Event/Incident\)" run book action policy](#).


## Enabling the Run Book Automation Policies

**NOTE:** Versions 104 and later of the "ServiceNow Base Pack" PowerPack separated these run book action policies by Cases, Events, and Incident, such as "ServiceNow: [Cases] - Add/Update".

Before you can run the "ServiceNow: Add/Update/Clear Case" run book action, you must enable the two case-specific run book automation policies in SL1:

- ServiceNow: [Cases] - Add/Update
- ServiceNow: [Cases] - Event Cleared

To enable the run book automation policies:

1. In SL1, go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
2. Locate the "ServiceNow: [Cases] - 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 "ServiceNow: Create, Update, Clear Incident or Event".

**TIP:** By default, the "ServiceNow: [Cases] Add/Update" automation policy will create ServiceNow Cases 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: [Cases] - Event Cleared" run book automation policy.

## Enabling and Customizing the Run Book Action Policy


The "ServiceNow: Add/Update/Clear Case" run book action policy contains snippet code that you can customize to use with the "ServiceNow Cases" SyncPack. You edit these values in the **Input Parameters** pane of the **Actions** page for this policy.

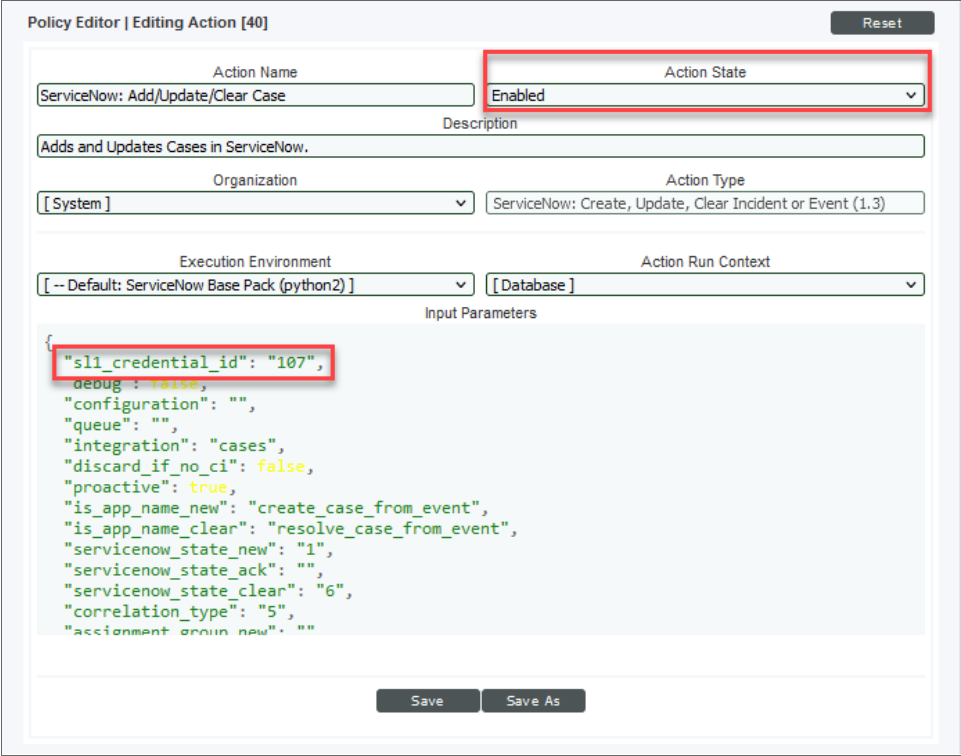
For example, you can choose to let PowerFlow create both an incident and a case in ServiceNow when an SL1 event is created, or you can choose to have PowerFlow create just an Incident when an SL1 event is created. You can edit this setting with the **proactive** parameter, which is described in the "Customizing the Snippet Code in the Input Parameters Pane" section, below.

**NOTE:** Make sure you are using the most recent version of the run book action policy. If there are two policies with the same name, always use the policy with the higher number in the **ID** column of the **Actions** page.

**NOTE:** Be sure to schedule the "Cache SL1 Users" PowerFlow application to run at least once a week to ensure that the user cache has the most recent data.

To enable and customize the Case run book action policy:

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



Policy Editor | Editing Action [40] Reset

Action Name: ServiceNow: Add/Update/Clear Case Action State: Enabled

Description: Adds and Updates Cases in ServiceNow.

Organization: [System] Action Type: ServiceNow: Create, Update, Clear Incident or Event (1.3)

Execution Environment: [-- Default: ServiceNow Base Pack (python2)] Action Run Context: [Database]

Input Parameters

```
{
  "sli_credential_id": "107",
  "debug": false,
  "configuration": "",
  "queue": "",
  "integration": "cases",
  "discard_if_no_ci": false,
  "proactive": true,
  "is_app_name_new": "create_case_from_event",
  "is_app_name_clear": "resolve_case_from_event",
  "servicenow_state_new": "1",
  "servicenow_state_ack": "",
  "servicenow_state_clear": "6",
  "correlation_type": "5",
  "assignment_group_new": ""
}
```

Save Save As

3. For the **Action State** field select *Enabled*.
4. For the **sli\_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: `"sli_credential_id": "101"`

5. Edit the snippet code as necessary, using the information in the **Customizing the Snippet Code in the Input Parameters Pane** section, below.
6. 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.

You can customize the following values in the "ServiceNow: Add/Update/Clear Case" 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 (Manage > Credentials of SL1). For example: `"sl1_credential_id": "41"`
- **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": "docs_config"`

**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.
- **integration**. Specifies the SyncPack you are using for this run book action. For example: `"integration": "cases"`.
- **discard\_if\_no\_ci**. Specifies whether PowerFlow should create cases in ServiceNow for devices that do not have a matching CI record. Your options include:
  - *true*. If a device is not mapped to a CI, PowerFlow will *not* create a case 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 in ServiceNow and update SL1. The default is *false*.
- **proactive**. Creates both a Case and an Incident in ServiceNow when an SL1 Event is created, or creates just a Case when an SL1 Event is created. If a Case or Incident is assigned in ServiceNow, the SL1 Event will be acknowledged. Users will be matched first on username, then email, then first and last name. The default is "true".
  - *true*. Specify "true" if you want to create both a Case and an Incident in ServiceNow. This option creates an "External Ticket" link for the SL1 Event that opens the corresponding Case in ServiceNow. If an Event has a Case in ServiceNow, the case status will be populated; if not, the Incident status will be populated. If a Case or Incident is assigned in ServiceNow, the SL1 Event will be acknowledged.

- *false*. Specify "false" if you only want to create an Incident in ServiceNow. This option creates an "External Ticket" link for the SL1 Event that opens the corresponding Incident in ServiceNow.

**WARNING:** If you change the **proactive** value from true to false after initial configuration and you have active Cases, the External Ticket link for the SL1 Event will continue to link to the Case record in ServiceNow. To avoid this situation, ensure that all synced Cases are Closed or Inactive before changing the **proactive** value. The External Ticket link will only be updated to the Incident record for new SL1 Events.

## Customizing Logging in the Run Book Action

You can customize the following logging-related items in the "ServiceNow: Add/Update/Clear Case" 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: [(Cases/Events/Incidents)] Add/Update" run book automation and the "ServiceNow: Add/Update/Clear (Case/Event/Incident)" run book action to "pass through" custom data about SL1 cases, events, or incidents to ServiceNow (depending on the SyncPack you are using with PowerFlow).

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.

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/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 **Actions** page (Registry > Run Book > Actions) and click **[Create]** to create a new run book action policy.
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, or leave them at their default settings.
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.

6. In the **Available Actions** section, add the new run book action *before* the "ServiceNow: Create, Update, Clear Incident" run book action:

The screenshot shows the 'Automation Policy Editor | Creating New Automation Policy' interface. The 'Policy Name' is 'ServiceNow: Add/Update Incident'. The 'Policy Type' is 'Active Events', 'Policy State' is 'Enabled', 'Policy Priority' is 'High', and 'Organization' is 'System'. The 'Criteria Logic' section includes filters for severity, time, and event status. The 'Match Logic' is 'Text search' and 'Match Syntax' is empty. The 'Repeat Time' is 'Only once' and 'Align With' is 'Devices'. The 'Trigger on Child Rollup' checkbox is checked. The 'Available Devices' list includes 'System' and 'ServiceNow: Instance: ven01056'. The 'Available Events' list includes various critical alerts. The 'Available Actions' list is highlighted with a red box and includes 'Snippet [5]: Example Passthrough EM7\_RESULT'. The 'Aligned Actions' list shows the sequence of actions: '1. Snippet [5]: Example Passthrough EM7\_RESULT' and '2. ServiceNow: Create, Update, Clear Incident [100]: Se'.

**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.

7. 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.

## Passthrough Example

For the Dictionary Entry of the ServiceNow field on the import table, you can reference the XML of the record. You will need to copy the `<sys_name>` value so you can use that as the key for the passthrough.

In this example, you want to bring in an additional field called **sl1 category**.

1. Create the new **sl1 category** field on the import table in ServiceNow. You can right-click on the header of the form to view the XML:

The screenshot shows the ServiceNow Dictionary Entry form for 'sl1 category'. The form fields are:

- \* Table: ScienceLogic Incident Import [x\_slo\_incide...]
- \* Type: String
- \* Column label: sl1 category
- \* Column name: u\_sl1\_category
- \* Max length: 24

Below the form fields, there are tabs for 'Choice List Specification' and 'Default Value'. The 'Choice List Specification' tab is active, showing a list of suggested values in a Choice list. Below this, there are buttons for 'Delete Column' and 'Update'.

On the right side of the screenshot, the XML view of the dictionary entry is displayed. The XML structure is as follows:

```
<reference_qual/>
<reference_qual_condition/>
<reference_type/>
<incident/>
<spell_check>false/>
<staged>false/>
<sys_class_name>sys_dictionary</sys_class_name>
<sys_created_by>admin</sys_created_by>
<sys_created_on>2023-05-11 15:31:11</sys_created_on>
<sys_id>4fcfd8741bbe0102e420fa6c4bcb52</sys_id>
<sys_mod_count>2</sys_mod_count>
<sys_name>sl1 category</sys_name>
<sys_package display_value>ScienceLogic SL1: Incident Automation</sys_package display_value>
<sys_package>indc1b03b6cf090aef3d4c4bcb3d</sys_package>
<sys_policy/>
<sys_scope display_value>ScienceLogic SL1: Incident Automation</sys_scope display_value>
<sys_update_name>sys_dictionary_x_slo_incident_import_u_sl1_category</sys_update_name>
<sys_updated_by>admin</sys_updated_by>
<sys_updated_on>2023-05-11 15:49:05</sys_updated_on>
<table_reference>false/>
<table_reference/>
<test_index>false/>
<test_index/>
<unique>false/>
<unique/>
<use_dependent_field>false/>
<use_dependent_field/>
<use_dynamic_default>false/>
<use_dynamic_default/>
<use_reference_qualifier/true/>
<use_reference_qualifier/>
<virtual>false/>
<virtual/>
<virtual_type>script</virtual_type>
<widget/>
<write_roles/>
<xml_view>false/>
</sys_dictionary>
</xml>
```

2. Look for the `<sys_name>` value.
3. Copy that value directly out and use that in your EM7\_RESULT for the passthrough value (in the Snippet Code pane):

```
EM7_RESULT = {'sl1 Category': 'test'}
```

## 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: [Cases] Click to Create" Automation Policy

The "ServiceNow: [(Cases/Events/Incident)] - Click to Create" 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 action policy is available in the "ServiceNow Base Pack" PowerPack.

To configure the "ServiceNow: Click to Create" run book action 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** page (Registry > Run Book > Automation).

4. Locate the "ServiceNow: (Cases/Events/Incident) - Click to Create" policy and click its wrench icon (). The **Automation Policy Editor** page appears:
5. Update the following fields:
  - **Policy State**. Select *Enabled*.
  - In the **Criteria Logic** section, select *and external ticket IS requested* in the fifth drop-down. Leave the other values in this section at their default settings.
  - **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 *ServiceNow: Send to PowerFlow: ServiceNow: Add/Update/Clear Incident* and add it to the **Aligned Actions** field.
6. Click [Save]. The "Click to Create" feature is now available on the **Events** and **Event Investigator** pages.

## Enabling Run Book Automation Queue Retries

You can enable run book action (RBA) queue retries to keep from losing any data if PowerFlow is unavailable. Those pending PowerFlow applications are added to an RBA queue that you can access to retry the applications that failed.


For more information, see [Enabling Run Book Automation Queue Retries](#).

---

## Configuring PowerFlow

The following topics cover how to set up your PowerFlow instance to work with the "ServiceNow Cases" SyncPack.

### Creating and Aligning a Configuration Object in PowerFlow

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 SyncPack, 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 environment and the third-party environment with which you are syncing data, you might be able to use the same configuration object with more than one SyncPack.

## Creating a Configuration Object

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

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

The screenshot shows the PowerFlow Configurations page. On the left, a table lists various configuration objects. The 'ServiceNow SyncPack' object is highlighted. On the right, the configuration details for 'ServiceNow SyncPack' are shown. A 'Toggle JSON Editor' button is visible. Below it, the JSON configuration data is displayed, enclosed in a red box. At the bottom right, there are 'Copy as' and 'Save' buttons.

CONFIG NAME	VER	AUTHOR	MODIFIED (UTC -5)	DESCRIPTION
Certified Application	1.3.3.7		Oct 22, 2019 10:56:01	ServiceNow SyncPack
IS - System Backup Configuration Example	1.0.0	ScienceLogic	Oct 30, 2019 15:45:11	Example
IS - System Diagnostic Configuration Example	1.0.0	ScienceLogic	Oct 30, 2019 15:45:11	Example
IS - System Backup Configuration	1.0.0	ScienceLogic	Oct 30, 2019 11:14:23	Example
ServiceNow SyncPack	1.1.0	ScienceLogic	Oct 17, 2019 13:54:22	Sample

**ServiceNow SyncPack**

Description: Sample Configuration file for the ServiceNow SyncPack

Version: 1.1.0

Configuration Data

```
1 {
2   "encrypted": false,
3   "name": "sl1_host",
4   "value": "en7.sciencelogic.com"
5 }
6 {
7   "encrypted": true,
8   "name": "sl1_password",
9   "value": "d4fUX0j0bs+96ylhkOp8DeLnqcooh2d8fwvnt6X6ck="
10 }
11 {
12   "encrypted": false,
13   "name": "sl1_user",
14   "value": "en7admin"
15 }
16 {
17   "encrypted": false,
18   "name": "sl1_db_host",
19   "value": "${config.sl1_host}"
20 }
21 {
22   "encrypted": false,
23   "name": "sl1_db_user",
24   "value": "root"
25 }
26 {
27   "encrypted": true,
28   "name": "sl1_db_password",
29   "value": "X3Wz66e/MPwhlWbJgX086bLnpFpdFQ/SZc2rCCYNY="
30 }
31 {
32   "encrypted": false,
33   "name": "snow_host",
34   "value": "sciencelogic.service-now.com"
35 }
36 {
37   "encrypted": false,
38   "name": "snow_user",
39   "value": "root"
40 }
```

expects type: json

Copy as Save

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

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

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

4. Complete the following fields:
  - **Friendly Name.** Name of the configuration object that will display on the **Configurations** page.
  - **Description.** A brief description of the configuration object.
  - **Author.** User or organization that created the configuration object.
  - **Version.** Version of the configuration object.
5. In the **Configuration Data** field, include the required block of code to ensure that the applications aligned to this configuration object do not fail:

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

For example:

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

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

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

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

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

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

7. To create a configuration variable in the JSON Editor, define the following keys:
  - **encrypted**. Specifies whether the value will appear in plain text or encrypted in this JSON file. If you set this to "true", when the value is uploaded, PowerFlow encrypts the value of the variable. The plain text value cannot be retrieved again by an end user. The encryption key is unique to each PowerFlow system. The value is followed by a comma.
  - **name**. Specifies the name of the configuration file, without the JSON suffix. This value appears in the user interface. The value is surrounded by double-quotes and followed by a comma.
  - **value**. Specifies the value to assign to the variable. The value is surrounded by double-quotes and followed by a comma.
8. If you want to use OAuth2 for authentication with ServiceNow, complete the following Configuration Data Values fields:
  - **snow\_oauth\_client\_id**. Enter the OAuth2 Client ID from ServiceNow.
  - **snow\_oauth\_client\_secret**. Enter the OAuth2 Client secret from ServiceNow.
  - **snow\_oauth\_token\_url**. Enter the full authentication URL, including host and protocol from ServiceNow. For example, "https://<test-instance-name>.service-now.com/oauth\_token.do"
  - **snow\_auth\_method**: Enter *oauth* or *http\_basic*. If no value is provided, *http\_basic* will be used for connection.

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

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

## Creating an OAuth2 Credential Record in ServiceNow

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

1. Navigate to **System OAuth > Application Registry**. The **Application Registries** page appears.
2. Click **[New]**.

3. Select **Create an OAuth API endpoint for external clients**. A new record appears.
  4. Complete the following fields on the new record:
    - **Name**. Type a unique name for the credential. Required.
    - **Client ID**. The Client ID is automatically generated by the ServiceNow OAuth server.
    - **Client Secret**. Leave this empty. ServiceNow will auto-generate this when the record is saved.
    - **Refresh Token Lifespan**. Enter the length of time in seconds the Refresh Token will be valid.
    - **Access Token Lifespan**. Type the length of time in seconds that the Access Token will be valid. ScienceLogic recommends setting this to 3,600 seconds to avoid known issues for longer ServiceNow REST interactions.
- NOTE:** In a scenario where the **[Access Token Lifespan]** value is shorter than the duration of a PowerFlow step that makes multiple REST interactions with ServiceNow, the access token will expire and need to be refreshed. As a result, retries were added to several PowerFlow steps where this issue may occur. This issue will hopefully be addressed in future versions of the Base Steps SyncPack.
5. Under the **Auth Scope** section at the bottom of the page, double click **Insert a new row**.
  6. In the search box that appears, click the magnifying glass icon, select the *useraccount* record, and click the checkmark icon to save.
  7. Click **[Submit]** to save the new record.

The screenshot displays the 'Application Registries' page for 'PowerFlow Token'. The 'OAuth client application details' section includes the following fields:

- Name:** PowerFlow Token
- Client ID:** 85abad6bed82c2507f0168b91a1c1bcf
- Client Secret:** (masked with dots)
- Refresh Token Lifespan:** 8,640,000
- Access Token Lifespan:** 3,600
- Redirect URL:** (empty)
- Logo URL:** (empty)
- Public Client:** ☐
- Login URL:** (empty)

The 'Auth Scopes' section at the bottom shows a table with one row:

Auth Scope
useraccount

Below the table is a button labeled 'Insert a new row...'.

## Aligning a Configuration Object and Configuring PowerFlow Applications

To run Case Sync, you must "align" the configuration object to run with the following PowerFlow applications:

- Cache SL1 Users
- Create or Update ServiceNow Case from SL1 Event
- Resolve ServiceNow Case from SL1 Event
- Sync Case or Incident State from ServiceNow to SL1 Event

**NOTE:** If you want to link cases with ServiceNow Configuration Items (CIs), you will need to run the "Sync Devices from SL1 to ServiceNow" application from the *ServiceNow CMDB SyncPack*. 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 SyncPack* manual.

To align the configuration object with the relevant PowerFlow applications:

1. On the **Applications** page of the PowerFlow user interface, open one of the PowerFlow applications listed above and click **[Configure]**. The **Configurations** pane for that application appears:

Create or Update ServiceNow Incident from SL1 Event

CancelSave

Align configuration and save

Configuration

scoped-app-conf

snow\_hostname

ven01056.service-no

\$(config.snow host)

snow\_user

is4User1

\$(config.snow user)

snow\_password

••••••••••••••••

\$(config.snow password)

sl1\_hostname

10.2.11.41

\$(config.sl1 host)

sl1\_user

em7admin

\$(config.sl1 user)

sl1\_password

••••••••••••••••

\$(config.sl1 password)

eventDetails

1

expects type: json

region

QARegion10

\$(config.region)

snow\_state

2

passthrough

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.

**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.

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

**NOTE:** Do *not* update any of the other parameters on the **Configuration** pane for any of these PowerFlow applications related to Case Sync. These values are updated by the "ServiceNow: Add/Update/Clear Case" Run Book Action.

4. Click **[Save]** to align that configuration with the application.
5. Repeat this process for the other PowerFlow applications. The "Cache SL1 Users" application is in the "System Utils" SyncPack.

## Scheduling PowerFlow Applications

ScienceLogic recommends that you schedule the following PowerFlow applications:

- "Cache SL1 Users": at least once a week
- "Sync Case or Incident State from ServiceNow to SL1 Event": every 60 seconds

Do not schedule the "Create or Update ServiceNow Case from SL1 Event" application or the "Resolve ServiceNow Case from SL1 Event" application. These applications should *only* be triggered by Run Book Automations.

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



---

# Chapter

# 4

## Troubleshooting the Cases SyncPack

---

### Overview

This chapter contains troubleshooting resources and procedures to use with the "ServiceNow Cases" SyncPack.

This chapter covers the following topics:

<i>Initial Troubleshooting Steps</i> .....	34
<i>Resources for Troubleshooting</i> .....	34

---

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

### ServiceNow

1. Make a note of the ServiceNow version and SyncPack version, if applicable.
2. Make a note if you are running a ServiceNow certified application or a Service Graph SyncPack.
3. Make a note of the SyncPack application that is failing in 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

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

- <https://<IP of PowerFlow>:3141>. Provides access to the pypiserver service, which you can use to see if SyncPacks have been correctly uploaded to Devpi container.

**IMPORTANT:** Port 5556 must be open for both 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 instance and run the following command:

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

To view the individual services with their respective image versions, 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@fsunislabs ~]# docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
0mihg35v30i	iservices_gui	replicated	1/1	repository.auto.sciencelogic.local:5000/is-gui:1.7.0	*:80->80/tcp,*:443->443/tcp
0v9e1t1vnh3	iservices_redis	replicated	1/1	redis:4.0.2	
0lsh61j2um1f	iservices_flower	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	*:5555->5555/tcp
hh3pt2l01rsf	iservices_scheduler	replicated	1/1	repository.auto.sciencelogic.local:5000/is-worker:1.7.0	
ht1mltv6xhx	iservices_contentapi	replicated	1/1	repository.auto.sciencelogic.local:5000/is-api:1.7.0	*:5000->5000/tcp
py1n5qgsudm1	iservices_rabbitmq	replicated	1/1	rabbitmq:3	
x1u19h3j5d46	iservices_visual	replicated	2/1	docker.samples/visualizer:latest	*:8081->8080/tcp
yoy38w8uauw	iservices_couchbase	replicated	1/1	repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0	*:8091->8091/tcp,*:8092->8092/tcp
0->8093/tcp,*:8094->8094/tcp,*:11210->11210/tcp					
z1bxatxoz7uf	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.

## Diagnosis Tools

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

- **Docker PowerPack.** This PowerPack monitors your Linux-based PowerFlow server with SSH (the PowerFlow ISO is built on top of an Oracle Linux Operating System). This PowerPack provides key performance indicators about how your PowerFlow server is performing. For more information on the Docker PowerPack and other PowerPacks that you can use to monitor PowerFlow, see the "Using SL1 to Monitor SL1 PowerFlow" chapter in the *SL1 PowerFlow Platform* manual.
- **Flower.** This web interface tool can be found at the /flower endpoint. It provides a dashboard displaying the number of tasks in various states as well as an overview of the state of each worker. This tool shows the current number of active, processed, failed, succeeded, and retried tasks on the PowerFlow platform. This tool also shows detailed information about each of the tasks that have been executed on the platform. This data includes the UUID, the state, the arguments that were passed to it, as well as the worker and the time of execution. Flower also provides a performance chart that shows the number of tasks running on each individual worker.
- **Debug Mode.** All applications can be run in "debug" mode via the PowerFlow API. Running applications in debug mode may slow down the platform, but they will result in much more detailed logging information that is helpful for troubleshooting issues. For more information on running applications in Debug Mode, see [Retrieving Additional Debug Information](#).
- **Application Logs.** All applications generate a log file specific to that application. These log files can be found at `/var/log/iservices` and each log file will match the ID of the application. These log files combine all the log messages of all previous runs of an application up to a certain point. These log files roll over and will get auto-cleared after a certain point.
- **Step Logs.** Step logs display the log output for a specific step in the application. These step logs can be accessed via the PowerFlow user interface by clicking on a step in an application and bringing up the **Step Log** tab. These step logs display just the log output for the latest run of that step.
- **Service Logs.** Each Docker service has its own log. These can be accessed via SSH by running the following command:

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

## Retrieving Additional Debug Information (Debug Mode)

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

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

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

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

1. Select the PowerFlow application from the **Applications** page.
2. Hover over the **[Run]** button and select *Custom Run* from the pop-up menu. The **Custom Run** window appears.
3. Select the Logging Level. *Debug* is the most verbose and will take longer to run.
4. Specify the configuration object for the custom run in the **Configuration** field, and add any JSON parameters in the **Custom Parameters** field, if needed.
5. Click **[Run]**.

To run an application in Debug Mode using the API:

1. POST the following to the API endpoint:

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

2. Include the following in the request body:

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

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

The screenshot displays the ScienceLogic PowerFlow user interface. At the top, the application title is "Delete Devices From SL1". Below the title, a workflow diagram shows the following steps: "Pull Disabled Devices from VCUG in SL1" (highlighted with a red box and an error icon), "Pull Affected Device Info from SL1 (SQL)", "Pull Affected Device Info from SL1 (MySQL)", "Verify Device Delete Requests", and "Delete Devices".

Below the workflow diagram is a table titled "STEP LOG" with columns: Module, Date (UTC-4), Log Level, and Message. The table contains the following entries:

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

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

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

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

© 2003 - 2024, 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). For more information, see <https://sciencelogic.com/company/legal>.



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

International: +1-703-354-1010