SL1 PowerFlow Service: ServiceNow Events Synchronization PowerPack

Version 1.0.0

Table of Contents

Chapter 1

Introduction to the ServiceNow Events Synchronization PowerPack

Overview

This chapter describes the SL1 *PowerFlow Service: ServiceNow Events* Synchronization PowerPack, which is the ScienceLogic integration with the ServiceNow Events Module. This Synchronization PowerPack provides a bi-directional sync of SL1 Events with ServiceNow Alerts, including event creation, acknowledgment, and resolution.

This Synchronization PowerPack requires a ServiceNow MID Server.

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

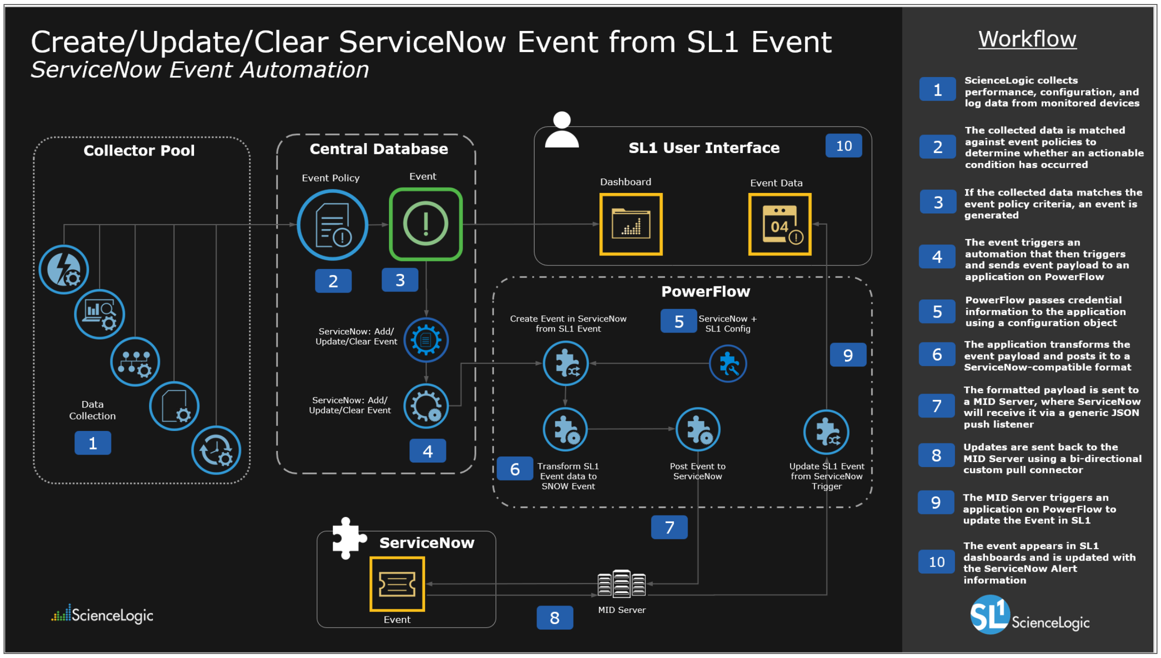
NOTE: After the 2.1.0 release of the Integration Service platform, the *Integration Service* was rebranded as the SL1 *PowerFlow service*, which will be available in SL1 Standard solutions. Also, the *Automation Builder* was rebranded as the SL1 *PowerFlow builder*, which will be available in SL1 Premium solutions.

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

This chapter covers the following topics:

Event Integration Architecture and Workflow

The following diagram details the workflow and architecture for the bi-directional sync of SL1 Events with ServiceNow Alerts, including event creation, acknowledgment, and resolution:



Create/Update/Clear ServiceNow Event from SL1 Event

ServiceNow Event Automation

Workflow

1. ScienceLogic collects performance, configuration, and log data from monitored devices
2. The collected data is matched against event policies to determine whether an actionable condition has occurred
3. If the collected data matches the event policy criteria, an event is generated
4. The event triggers an automation that then triggers and sends event payload to an application on PowerFlow
5. PowerFlow passes credential information to the application using a configuration object
6. The application transforms the event payload and posts it to a ServiceNow-compatible format
7. The formatted payload is sent to a MID Server, where ServiceNow will receive it via a generic JSON push listener
8. Updates are sent back to the MID Server using a bi-directional custom pull connector
9. The MID Server triggers an application on PowerFlow to update the Event in SL1
10. The event appears in SL1 dashboards and is updated with the ServiceNow Alert information

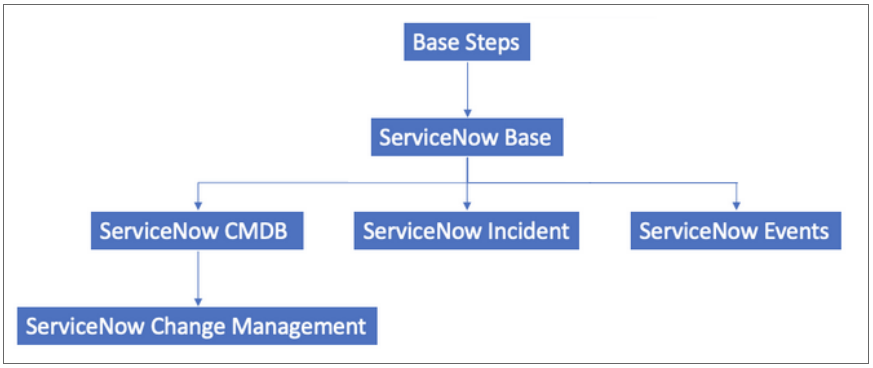
SL1 and ServiceNow Terminology

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

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

Dependency Map for ServiceNow Synchronization PowerPacks

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



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

Prerequisites for ServiceNow Synchronization PowerPacks

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

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

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

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

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

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

Contents of the Synchronization PowerPack

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

Integration Applications

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

- ***Create Event in ServiceNow from SL1 Event***. Transforms SL1 event data into a ServiceNow event, and then posts that event to ServiceNow.
- ***Update SL1 Event from ServiceNow Trigger***. Updates or clears an SL1 Event based on a ServiceNow action.
- ***Update SL1 Events***. Updates SL1 events based on the method selected in the configuration

Steps

- Convert ScienceLogic Event Data to SNOW Event
- Process Event Trigger from ServiceNow

Chapter

2

Installing the Events Synchronization PowerPack

Overview

This chapter describes how to install the *ServiceNow Events* Synchronization PowerPack in the PowerFlow service, and how to install the *ServiceNow Base Pack* PowerPack in SL1.

This chapter covers the following topics:

Downloading and Installing the Synchronization PowerPack

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

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

To locate and download the Synchronization PowerPack:

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

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

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

Importing the Synchronization PowerPack

To import a Synchronization PowerPack in the PowerFlow service user interface:

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

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

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

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

Installing the Synchronization PowerPack

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 service user interface:

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

   TIP: By default, the **SyncPacks** page displays only activated and installed PowerPacks. If you do not see the PowerPack that you want to install, click the 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" integration application appears, and you can view the Step Log for the steps. For a failed installation, the **Error Logs** window appears.

Installing the ServiceNow Base Pack PowerPack in SL1

The *ServiceNow Base Pack PowerPack* version 103 or later contains Run Book Automations that you use to integrate with the ServiceNow Events 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. Download the latest version of the PowerPack from the Customer Portal to a local computer.
2. In SL1, log in and go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. Click **[Actions]** and select *Import PowerPack*.
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.

Chapter

3

Configuring Integrations for the Events Synchronization PowerPack

Overview

This chapter describes how to configure the bi-directional Event integration between ServiceNow and SL1.

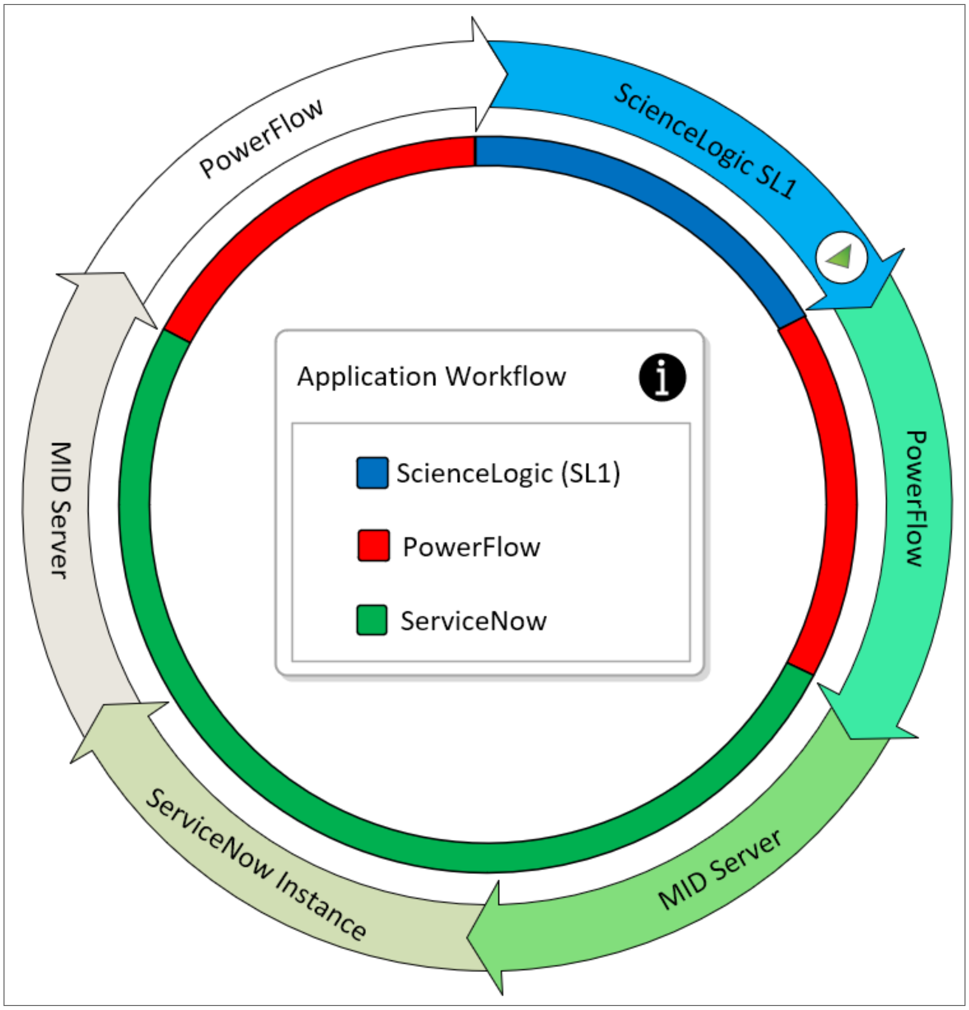
This chapter covers the following topics:

An Overview of Configuring Event Integration

The bi-directional Event integration provided by the *ServiceNow Events* Synchronization PowerPack requires the following components:

- An SL1 instance (the ScienceLogic platform)
- The SL1 PowerFlow service
- A ServiceNow Management, Instrumentation, and Discovery (MID) Server, which includes an Event Listener
- A ServiceNow instance

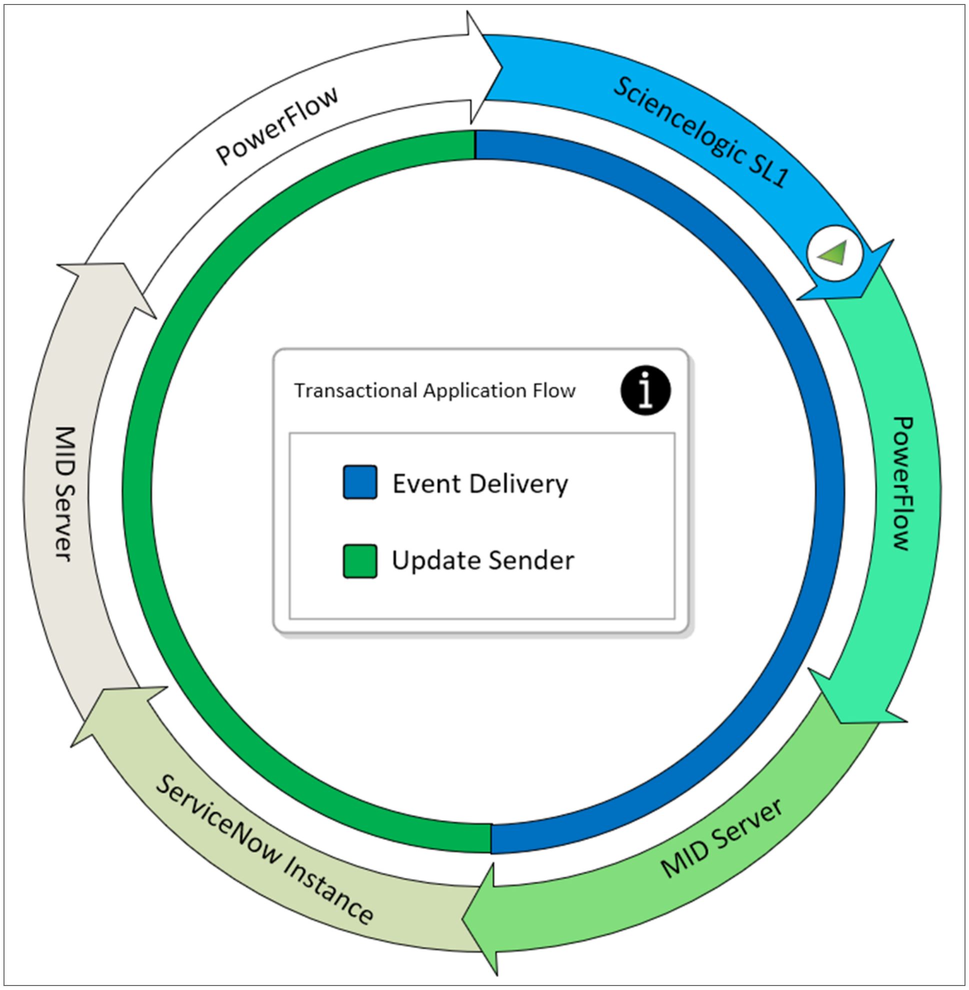
The following diagram shows the flow of Event data through these different components, starting with SL1:



The Event integration includes two main transaction types:

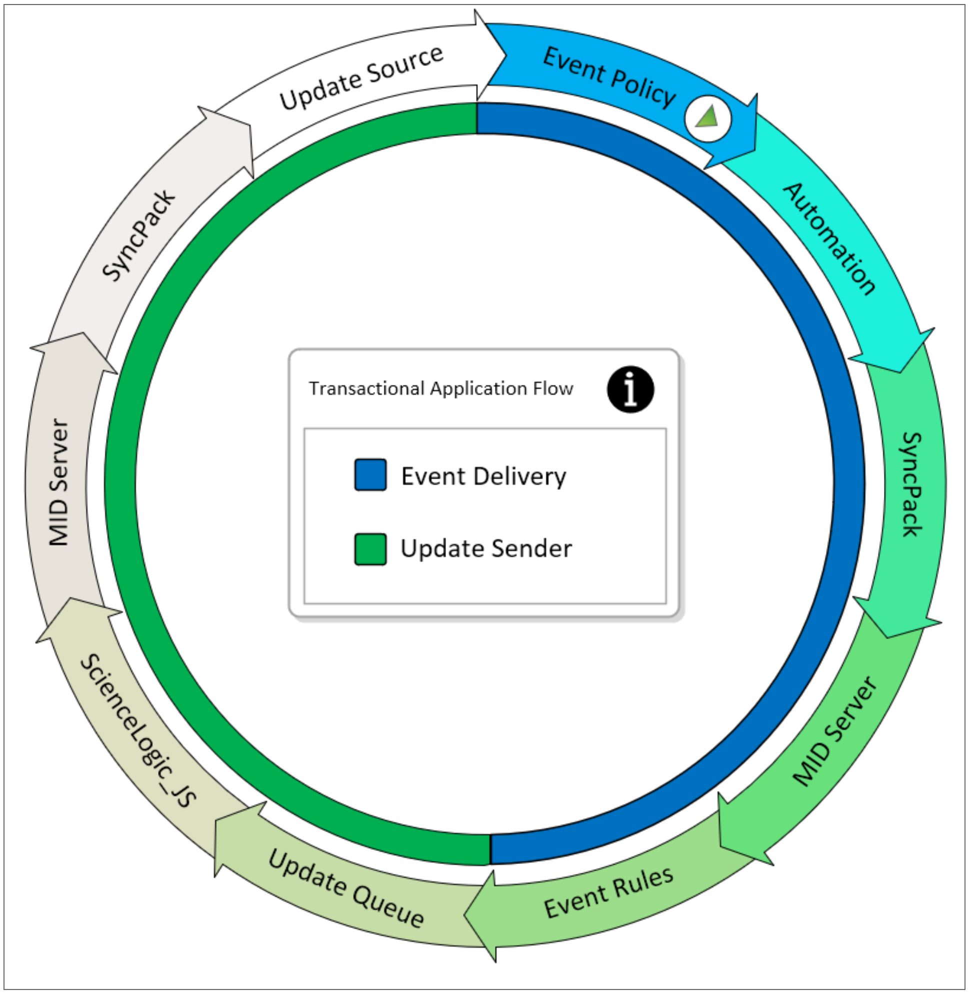
- ***Event Delivery***. Sends SL1 Events to ServiceNow using a push methodology. This method incorporates three technologies starting with SL1, followed by the PowerFlow service, and ending with the ServiceNow MID Server Event Listener. In the following diagram, Event Delivery is represented by the blue half-circle inside the arrows on the right.
- ***Update Sender***. Sends updates or changes to Alerts from the ServiceNow instance to the MID Server, which determines which Alert changes should be bundled and sent back to the PowerFlow service. Next, the PowerFlow service syncs the Alert changes with SL1 Events. In the following diagram, Update Sender is represented by the green half-circle inside the arrows on the left.

The following diagram shows the flow of the **Event Delivery** and **Update Sender** transactions through the different components, starting with SL1:



The major transaction workflow is made up of many steps that end with the delivery of an Alert triggered in SL1, and the process ends with updating Events back to the PowerFlow service. Each step is responsible for either evaluating, transforming, or moving data between different components or within a component.

The following diagram displays the different steps and components used by the **Event Delivery** and **Update Sender** transactions:



The **Event Delivery** transaction uses the following steps and components:

- ***Event Policy***. A system-defined event trigger in SL1 starts the **Event Delivery** transaction.
- ***Automation*** (also called a Run Book Automation). Lets you specify the actions you want SL1 to execute automatically when specific event conditions are met. Be sure to configure the Automation correctly, as it is the first step in passing Alert and resulting Event data to another application.
- ***SyncPack*** (*ServiceNow Events* Synchronization PowerPack). The SyncPack sends the Event to ServiceNow and handles the data received from SL1. Based on the configuration, the SyncPack transforms or formats the data to ServiceNow JSON-defined format and forwards the message on to MID Server. This is a triggered integration application and is not scheduled-based.

- ***MID Server***. The Event Listener on the MID Server listens on a user-defined port, and it takes the JSON objects it receives from the SyncPack and applies them to the Event table (**em_event**). While there are multiple ways to handle events from an outside source, SL1 pushes Events to the MID WebService Event Collector using a JSONv2-formatted message defined by ServiceNow.
- ***Event Rules***. These rules are not defined within the "ScienceLogic SL1: Event Automation" application (also called the Certified Application), but they have a direct impact on the Event raising an Alert.

The **Update Sender** transaction, which updates Alerts that are being sent back to their original sources, uses the following steps and components:

- ***Update Queue***. A ServiceNow-developed business rule that sits on the Alert table (**em_alert**). This rule tracks any changes to a record, including all the delta changes when records are inserted or updated. The rule posts the changes to the **connector update queue** table for bulk export back to the source.
- ***ScienceLogic_JS***. MID Server Script Include delivered with the ScienceLogic Application. This handles what is updated back to the Sender.
- ***MID Server***. Sends the payload compiled by the MID Server script to the PowerFlow service.
- ***SyncPack*** (*ServiceNow Events* Synchronization PowerPack). Handles the payload received from the MID Server script about updates that need to be made to specific SL1 Events.
- ***Update Source***. Makes the changes directly to the required SL1 Events.

TIP: For more information about troubleshooting the configuration, see ***Troubleshooting Event Delivery and Update Sender Transactions***.

Workflow for Configuring Event Integration

The following topics cover how to configure the various components that make up the Event integration:

Configuring the ServiceNow MID Server

1. *Install the MID Server*.
2. *Configure the MID Web Server*.
3. *Configure the MID WebService Event Collector*.
4. *Start the MID Web Server and the Event Collector*.
5. *Run an external test of the MID Server configuration*.

Configuring the ServiceNow Instance

1. Install the ServiceNow Event Management plugin (**com.glideapp.itom.snac**). You can request this plugin directly from ServiceNow.
2. *Install the "ScienceLogic SL1: Event Integration" (Scoped) application* (also called the Scoped or Certified application).
3. *Review the MID Server Script Include (ScienceLogic_JS)*.
4. *Configure the Connector Definitions*.
5. *Create a ServiceNow credential to use with the Connector Instance*.
6. *Configure the Connector Instance*.

Configuring the SL1 Instance

1. *Install the ServiceNow Base Pack PowerPack* version 103 or later.
2. *Create a ServiceNow credential in SL1*.
3. *Enable the following Run Book Automation Policies*:
   - ServiceNow: [Events] - Add/Update
   - ServiceNow: [Events] - Event Acknowledged
   - ServiceNow: [Events] - Event Cleared
4. *Enable the "ServiceNow: Add/Update/Clear Event" Run Book Action*.
5. *Optionally, send custom data to ServiceNow using the Pass-through Option*.
6. Optionally, *enable the "ServiceNow: [Events] - Click to Create" Automation Policy*.

Configuring the PowerFlow Service

1. *Install the ServiceNow Events Synchronization PowerPack*.
2. *Create a configuration object in the PowerFlow service user interface.*

3. *Align the new configuration file with the following integration applications*:
   - "Create Event in ServiceNow from SL1 Event"
   - "Update SL1 Event from ServiceNow Trigger"
   - "Update SL1 Events"

Configuring the ServiceNow MID Server

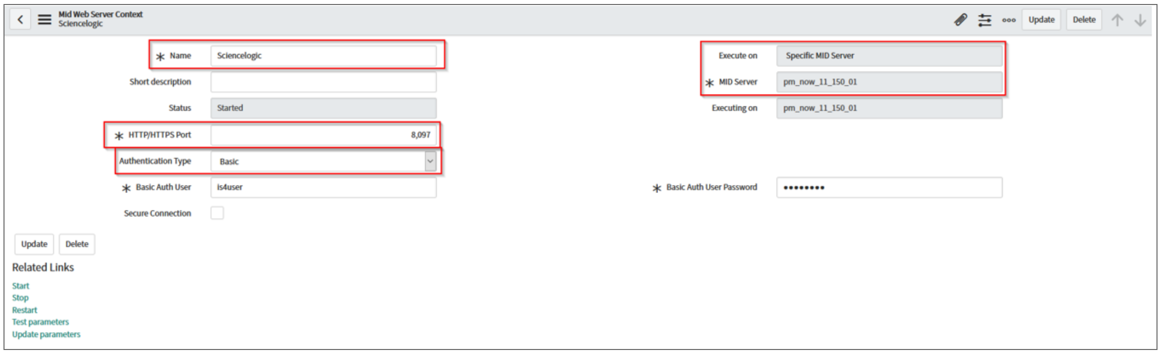
See the following link for instructions on installing and configuring the MID Server for your ServiceNow instance: MID Server installation.

After you set up the MID Server, you can use the following procedures to configure the ServiceNow MID Server to work with the *ServiceNow Events* Synchronization PowerPack.

Configuring the MID Web Server

To configure the MID Web Server (a MID Server is required for this step):

1. In ServiceNow, install the Event Management plugin. You can request this plugin directly from ServiceNow.
2. Navigate to **Mid Server > Extensions > MID Web Server**. The **Mid Web Server Contexts** page appears.
3. Click **[New]** and complete the following fields on the new record:
   - ***Name***. Type a unique name. Required.
   - ***HTTP/HTTPS Port***: Specify a port number to listen on. The port number must be unique.
   - ***Authentication Type***: Select *Basic*. Currently Basic is the only supported Authentication method. Complete the ***Basic Auth User*** and ***Basic Auth User Password*** fields with the relevant credentials.
   - ***Execute on***: Select a specific MID Server or Cluster of MID Servers.
   - ***MID Server***: Specify the MID Server or Mid Server Cluster you will use.



Mid Web Server Context
Sciencelogic

∗ Name: Sciencelogic
Short description:
Status: Started
∗ HTTP/HTTPS Port: 8,097
Authentication Type: Basic
∗ Basic Auth User: is4user
Secure Connection: □

Execute on: Specific MID Server
∗ MID Server: pm_now_11_150_01
Executing on: pm_now_11_150_01
∗ Basic Auth User Password: ••••••••

4. Click **[Submit]** to save the new record.

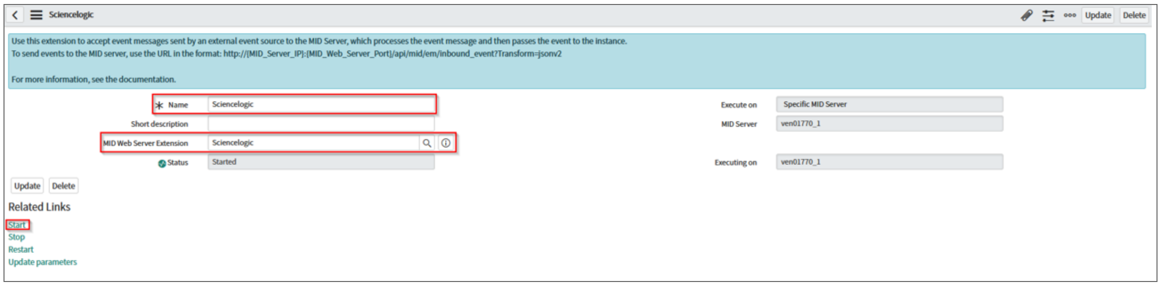
> **NOTE:** In the **Related Links** section, the **Test parameters** option is not a valid test to confirm your setup.

Configuring the MID WebService Event Collector

To configure the MID WebService Event Collector:

1. In ServiceNow, navigate to **Mid Server > Extensions > MID WebService Event Listener**.
2. Click **[New]** and complete the following fields on the new record:

   - ***Name***. Type a unique name. Required.
   - ***MID Web Server Extension*** : Select the MID Web Server you created in *Configuring the MID Web Server*.



Use this extension to accept event messages sent by an external event source to the MID Server, which processes the event message and then passes the event to the instance.
To send events to the MID server, use the URL in the format: http://[MID_Server_IP]:[MID_Web_Server_Port]/api/mid/em/inbound_event?Transform=jsonv2

For more information, see the documentation.

* Name: Sciencelogic
Short description:
MID Web Server Extension: Sciencelogic
Status: Started
Execute on: Specific MID Server
MID Server: ven01770_1
Executing on: ven01770_1

3. Click **[Submit]** to save the new record.

Starting the MID Web Server and the Event Collector

To start the MID Web Server and the Event Listener:

1. Navigate the **MID Server > Extensions > Mid Web Server** and select the MID Web Server you set up in *Configuring the MID Web Server*.
2. In the **Related Links** section, click the **Start** link.
3. Navigate to the **MID Server > Extensions > MID WebService Event Listener** and select the MID WebService Event Listener you set up in *Configuring the MID WebService Event Collector*.
4. In the **Related Links** section, click the **Start** link.
5. Confirm that the MID Web Server and the MID WebService Event Listener has a Status of "Started".

External Testing of the MID Server Configuration

ScienceLogic recommends that you test the MID Server connection by using an external testing source outside of ServiceNow, the PowerFlow service, and SL1.

1. Post a message to **http://{MID_SERVER_IP}:{MID_Web_Server_port}/api/mid/em/jsonv2**
2. Example body:

```
{
    "records":
    [
       {
        "source" : "Simulated",
        "event_class": "ScienceLogic",
        "node" : "Postman",
        "type" : "High Virtual Memory",
        "resource" : "C:",
        "severity" : "2",
        "description" : "Test Event",
        "ci_type":"cmdb_ci_app_server_tomcat",
        "message_key": "Postman_109843242",
        "additional_info":{"name":"Create an event in snow"}
       }
   ]
}
```

IMPORTANT: The value of the ***Source Instance*** (event_class) field must be the *same* case-sensitive value used in the Connector Instance ***Name***. If the values do not match exactly, the bi-directional connection will not work. In the example above, "ScienceLogic" is the ***Source Instance*** (event_class) name. For more information, see *Configuring the Connector Instances.*

3. Verify that the Send resulted in an **HTTPS Status 200**.

Using the MID Server Script Include (ScienceLogic_JS) in ServiceNow

The MID Server Script Include is provided with the "ScienceLogic SL1: Event Automation" application (also called the Certified Application), and it requires the ServiceNow Event Management plugin (**com.glideapp.itom.snac**). You can request the ServiceNow Event Management plugin directly from ServiceNow.

NOTE: For information about installing the Certified application, see *Installing the "ScienceLogic SL1: Event Integration Application" in ServiceNow*.

The MID Server Script Include contains the following main parts:

- Test Connection
- Execute
- Update Source

Test Connection is for validating that setup is complete for returning delta changes back to the source instance.

The Test Connection process includes the following steps:

1. The Initialize step pulls in the required functions and defines default variables.
2. The Test Connection is a two-part requirement that checks for the application and configuration availability:
   - Validates Application is available via GET HTTP request to the PowerFlow service.
   - Validates Configuration is available via Get HTTP request to the PowerFlow service.
3. For the Logs step, because the script is run on the MID Server, all logs are submitted to **ms.log()**.

Execute: The part of the Script is currently is not in use, but it runs the same process as test Connection step. At this time the execute step does not pull down events from ScienceLogic with the MID Server Script Include. Events are delivered using a push methodology to the MID Server Listener.

Update Source: Picks up pending updates on the Connector Update Queues (**em_connector_update_queue**) table and sends them back to the configured source instance.

The Update Source process includes the following steps:

1. The Initialize step pulls in the required functions and defines default variables.
2. Builds the URL.
3. Validates the URL.
4. Pulls all Records matching even_class, source instance and connector Instance name sitting in pending.
5. For each record, checks to see if the record matches the following criteria:
   - *New*: Checks the ***Number*** field being null and a value being assigned.
   - *Open*: Checks for field change on the ***State*** field from Closed to a new value of Reopen or Open.
   - *Closed*: Checks for field change on the ***State*** field from Open or Reopen to a new value of Closed.
   - *Acknowledge*: Due the difference in how acknowledgments are handled in the two platforms, the Acknowledged criteria tracks the boolean value flipping to true and will return either the default ScienceLogic user_id of 1 or a defined user id.
6. Matching values are written to new JSON code.
7. Matching JSON code is validated to ensure that it has information that needs to be sent back to source.
8. Posts the Request back to PowerFlow service.
9. For the Logs step, because the script is run on the MID Server, all logs are submitted to **ms.log()**.

Installing the "ScienceLogic SL1: Event Integration" (Scoped) Application in ServiceNow

The *ServiceNow Events* Synchronization PowerPack uses the "ScienceLogic SL1: Event Integration" application (also called the Scoped or Certified application) to sync event status updates from ServiceNow back to SL1. You must install the "ScienceLogic SL1: Event Integration" application on your ServiceNow instance to enable this Synchronization PowerPack.

WARNING: The "ScienceLogic SL1: Event Integration" application requires that you have already installed the ServiceNow Event Management plugin on the target ServiceNow instance. You can request this plugin directly from ServiceNow. Also, the MID Server must be configured.

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

You first request the "ScienceLogic SL1: Event Integration" application from the ServiceNow Store, and then you can install it.

To request and install the "ScienceLogic SL1: Event Integration" application:

1. Go to the ServiceNow Store at https://store.servicenow.com and search for "ScienceLogic SL1".
2. Select the "ScienceLogic SL1: Event Integration" 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 SL1: Event 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 SL1: Event 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.

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

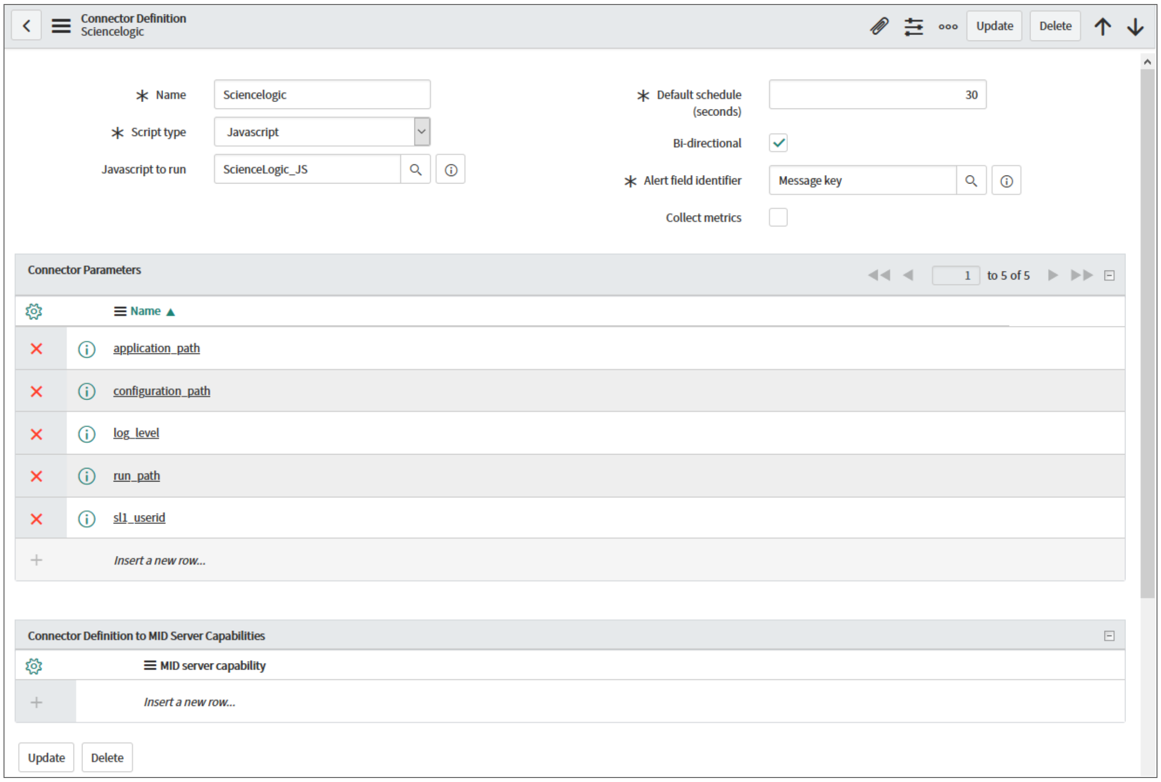
Configuring the Connector Definitions in ServiceNow

To configure the Connector Definitions used by the Update Sender transaction:

1. Navigate to **Event Management > Event Connectors (Pull) > Connector Definitions**. The **Connector Definitions** page appears.
2. Click **[New]** to create a new Connector Definition.

3. In the ***Name*** field, type a unique name and click **[Submit]** to save the new Connector Definition record. The **Connector Definitions** page appears.

> CAUTION: When you create a Connector Definition, ServiceNow creates a template MID Server Script Include with the same name and "_JS" appended at the end of the name. This item is automatically attached to the new record in the ***JavaScript to run*** field after you click **[Submit]**.



* Name	Sciencelogic
* Script type	Javascript
Javascript to run	ScienceLogic_JS
* Default schedule (seconds)	30
Bi-directional	☑
* Alert field identifier	Message key
Collect metrics	☐

Connector Parameters

1 to 5 of 5

Name ▲
application_path
configuration_path
log_level
run_path
sl1_userid
Insert a new row...

Connector Definition to MID Server Capabilities

MID server capability
Insert a new row...

4. Open the Connector Definition you just created and complete the following fields:

- ***Javascript to run***. Attach the JavaScript file from the Certified application.
- ***Default schedule (seconds)***. Specify the time in seconds between retrieval of events, such as "120". This field is not required for this integration specifically, but it is a required field. Events will not be pulled when this is triggered.
- ***Bi-Directional***. Select this option to enable ServiceNow to send Alert changes back to the source instance. Enabling this allows SL1 as an external monitoring system to receive changes made in ServiceNow that are specific to Alerts.
- ***Alert field identifier***. Set to "Message Key". The field causes the source monitoring system to be updated.

5. In the *Connector Parameters* section, double-click to add one or more rows using the following parameters to enable correct formatting and communication with the source event system:

Name	Example	Required
application_path	**/api/v1/applications/update_sl1_event_from_snow_trigger**	No
configuration_path	**/api/v1/configurations/** + Configuration Name	Yes
log_level	**30** (Default Log level)	No
run_path	**/api/v1/applications/run**	No
sl1_userid	**1** (Default user_id in SL1)	No

NOTE: The entries that are marked as not required are supplied by default.

6. Click **[Submit]** to save the updated Connector Definition

Configuring the Connector Instance in ServiceNow

Before you can set up a Connector Instance in ServiceNow, you need to create a ServiceNow credential that will be used by the Connector Instance.

Creating a ServiceNow Credential

To configure the ServiceNow credential that will be used by the Connector Instance:

1. Navigate to **Connections & Credentials > Credentials**. The **Credentials** page appears.
2. Click **[New]**.
3. Select **Basic Auth Credentials**. A new **Basic Auth Credentials** record appears.
4. Complete the following fields on the new record:

    - ***Name***. Type a unique name for the credential. Required.
    - ***User name***. Type the ServiceNow user name.
    - ***Password***: Type the ServiceNow password.

5. Click **[Submit]** to save the new record.
6. After the record is saved, open it again and check the ***Active*** field on the record.
7. Click **[Update]**.

Configuring the Connector Instance

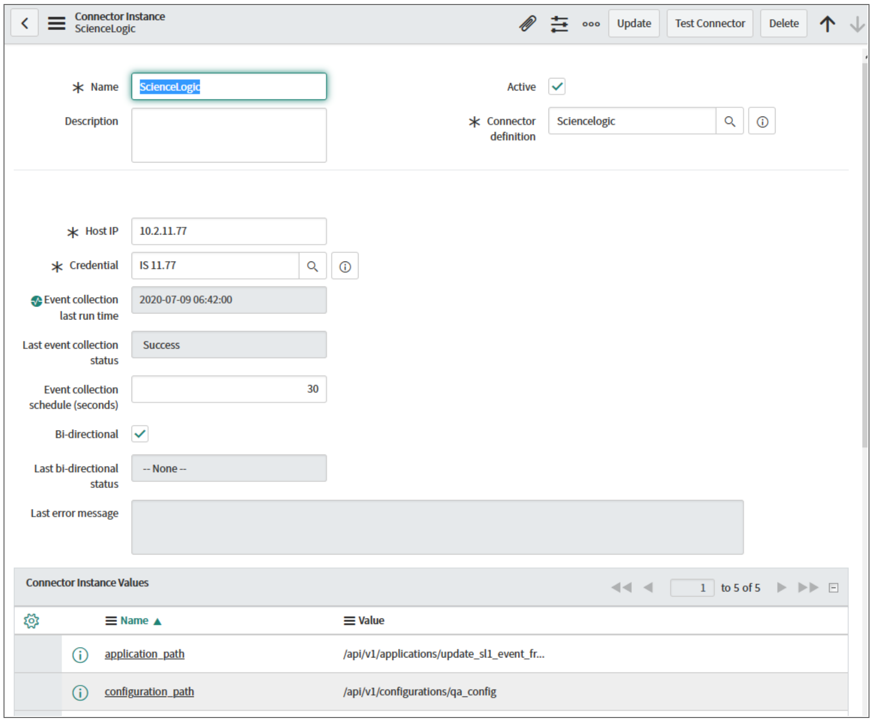
To configure the Connector Instance:

1. Navigate to **Event Management > Event Connectors (Pull) > Connector Instances**.
2. Click **[New]** and complete the following fields on the new record:

   - ***Name***. Type a unique name. Required.

     > WARNING: The value of the Connector Instance ***Name*** must be the *same* case-sensitive value used in the ***Source Instance*** (event_class) field. If the values do not match exactly, the bi-directional connection will not work. In the example below, "ScienceLogic" is the Connector Instance Name.

   - ***Connector definition***. Confirm that the Connector Definition *you created* is associated with this Connector Instance record.
   - ***Host IP***: Specify the host IP address for the PowerFlow service.
   - ***Credential***. Select the credential for connection to the PowerFlow service. For more information, see *Configuring the ServiceNow Credential*.
   - ***Bi-directional***. Select this option.



* Name	ScienceLogic
Description	
Active	✓
* Connector definition	Sciencelogic
* Host IP	10.2.11.77
* Credential	IS 11.77
Event collection last run time	2020-07-09 06:42:00
Last event collection status	Success
Event collection schedule (seconds)	30
Bi-directional	✓
Last bi-directional status	-- None --
Last error message	

Connector Instance Values

1 to 5 of 5

Name ▲	Value
application_path	/api/v1/applications/update_sl1_event_fr...
configuration_path	/api/v1/configurations/qa_config

3. In the ***Connector Instance Values*** section, ensure that the values have been configured:

Name	Example	Required
application_path	**/api/v1/applications/update_sl1_event_from_snow_trigger**	No
configuration_path	**/api/v1/configurations/** + Configuration Name	Yes
log_level	**30** (Default Log level)	No
run_path	**/api/v1/applications/run**	No
sl1_userid	**1** (Default user_id in SL1)	No

4. In the ***MID Servers for Connectors*** section, select the MID Server.
5. Click **[Submit]** to save the new record.
6. After the record is saved, open it again and click **[Test Connector]** to confirm the correct configuration. This option is only available after you submit the record.
7. If you have a successful test connection, check the ***Active*** field on the record.
8. Click **[Update]**.

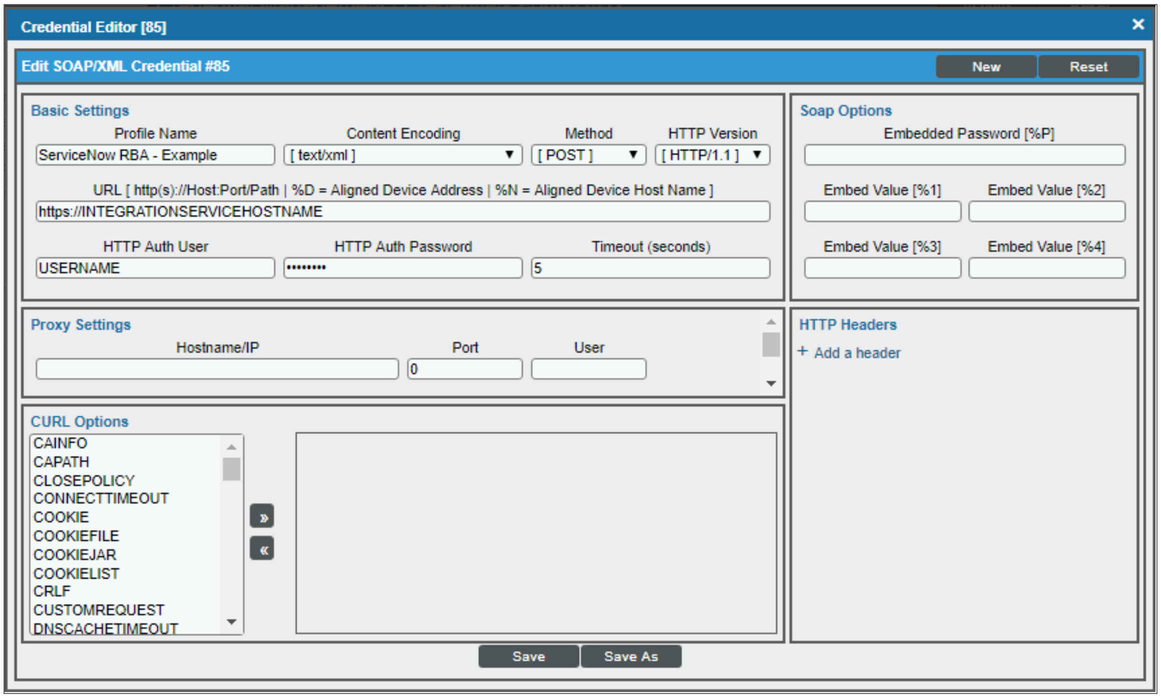
Creating a ServiceNow Credential in SL1

To configure SL1 to communicate with ServiceNow, you must first create a SOAP/XML credential. This credential allows the Run Book Automation scripts in the *ServiceNow Base Pack* PowerPack to connect with your ServiceNow instance.

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:



Credential Editor [85]

Edit SOAP/XML Credential #85

Basic Settings

Profile Name	Content Encoding	Method	HTTP Version
ServiceNow RBA - Example	[text/xml]	[POST]	[HTTP/1.1]

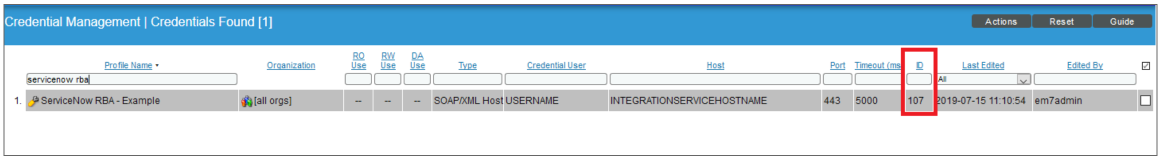
URL [http(s)://Host:Port/Path \| %D = Aligned Device Address \| %N = Aligned Device Host Name]: https://INTEGRATIONSERVICEHOSTNAME

HTTP Auth User	HTTP Auth Password	Timeout (seconds)
USERNAME	••••••••	5

Proxy Settings

Hostname/IP	Port	User
	0	

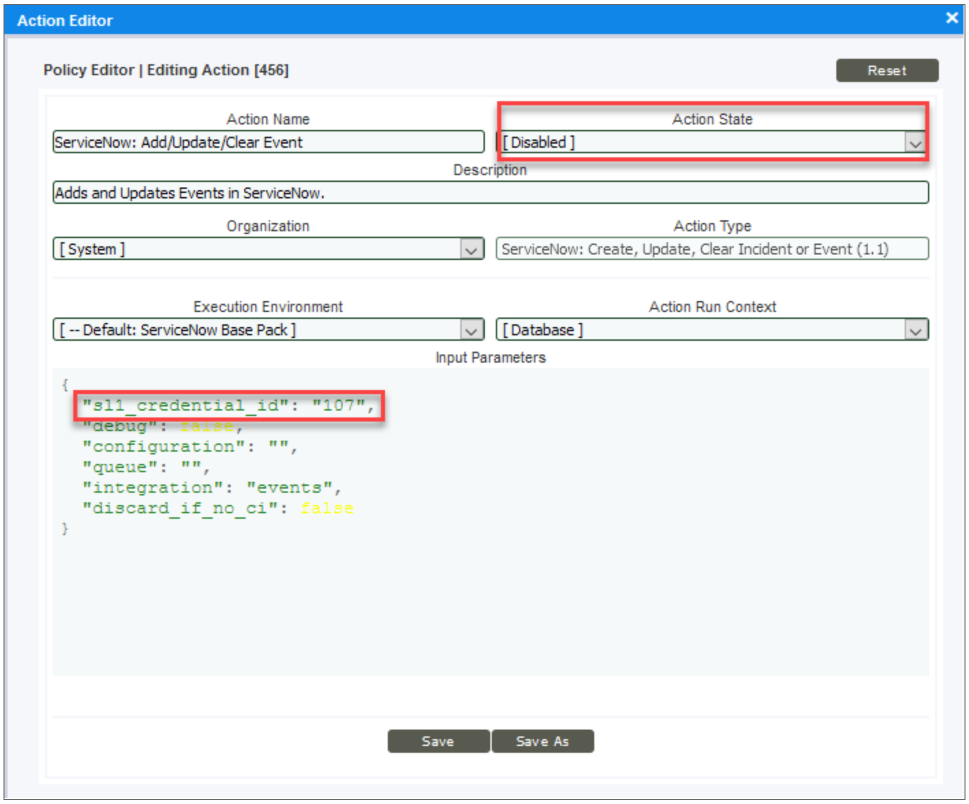
3. Complete the following fields:
   - ***Profile Name***. Type a new name for the ServiceNow credential.
   - ***Content Encoding***. Make sure *text/xml* is selected.
   - ***Method***. Make sure *POST* is selected.
   - ***HTTP Version***. Select *HTTP/1.1*.
   - ***URL***. Type the URL for your Integration Service instance.
   - ***HTTP Auth User***. Type the username of your Integration Service instance.
   - ***HTTP Auth Password***. Type the password of your Integration Service instance.
   - ***Timeout***. Type "5".
4. Click **[Save As]**.
5. When the confirmation message appears, click **[OK]**.
6. 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	☑
	servicenow rba											All		
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 **sl1_credential_id** parameter when you *enable the snippet code of the "ServiceNow: Add/Update/Clear Event" Run Book Action Policy*:



Action Editor

Policy Editor | Editing Action [456]

Action Name: ServiceNow: Add/Update/Clear Event

Action State: [Disabled]

Description: Adds and Updates Events 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

```
{
  "sl1_credential_id": "107",
  "debug": false,
  "configuration": "",
  "queue": "",
  "integration": "events",
  "discard_if_no_ci": false
}
```

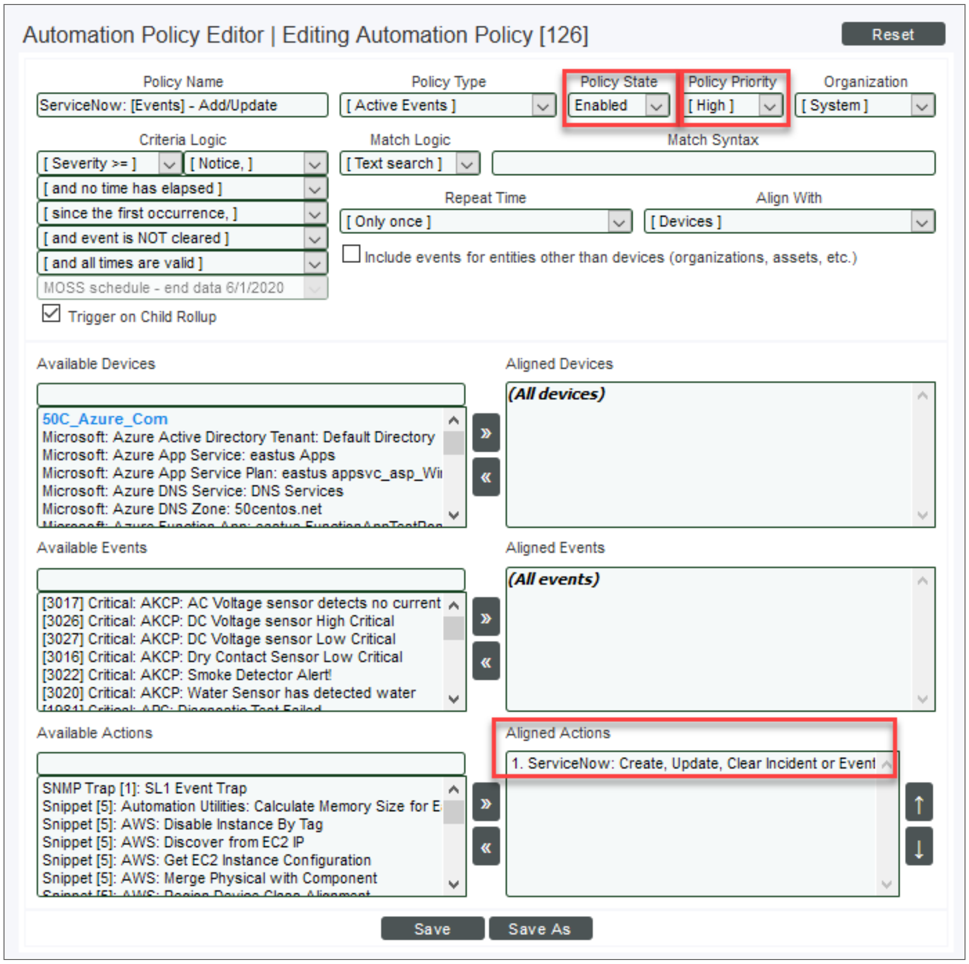
Enabling the Run Book Automation Policies in SL1

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

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

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: [Events] Add/Update" automation policy and click its wrench icon (🔧). The **Automation Policy Editor** page appears:



Automation Policy Editor | Editing Automation Policy [126]

Policy Name: ServiceNow: [Events] - Add/Update

Policy Type: [Active Events]

Policy State: Enabled

Policy Priority: [High]

Organization: [System]

Criteria Logic: [Severity >=] [Notice,]
[and no time has elapsed]
[since the first occurrence,]
[and event is NOT cleared]
[and all times are valid]
MOSS schedule - end data 6/1/2020

Match Logic: [Text search]

Match Syntax:

Repeat Time: [Only once]

Align With: [Devices]

☐ Include events for entities other than devices (organizations, assets, etc.)

☑ Trigger on Child Rollup

Available Devices

50C_Azure_Com
Microsoft: Azure Active Directory Tenant: Default Directory
Microsoft: Azure App Service: eastus Apps
Microsoft: Azure App Service Plan: eastus appsvc_asp_Wi
Microsoft: Azure DNS Service: DNS Services
Microsoft: Azure DNS Zone: 50centos.net

Aligned Devices

(All devices)

Available Events

[3017] Critical: AKCP: AC Voltage sensor detects no current
[3026] Critical: AKCP: DC Voltage sensor High Critical
[3027] Critical: AKCP: DC Voltage sensor Low Critical
[3016] Critical: AKCP: Dry Contact Sensor Low Critical
[3022] Critical: AKCP: Smoke Detector Alert!
[3020] Critical: AKCP: Water Sensor has detected water

Aligned Events

(All events)

Available Actions

SNMP Trap [1]: SL1 Event Trap
Snippet [5]: Automation Utilities: Calculate Memory Size for E
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

Aligned Actions

1. ServiceNow: Create, Update, Clear Incident or Event

3. Update the following fields:
   - ***Policy State***. Select *Enabled*.
   - ***Policy Priority***. Select *High* to ensure that this PowerFlow service 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 and add it to the **Aligned Actions** column.

> 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: Event Acknowledged" and "ServiceNow: Event Cleared" Run Book Automation policies.

Enabling and Customizing the Run Book Action Policy

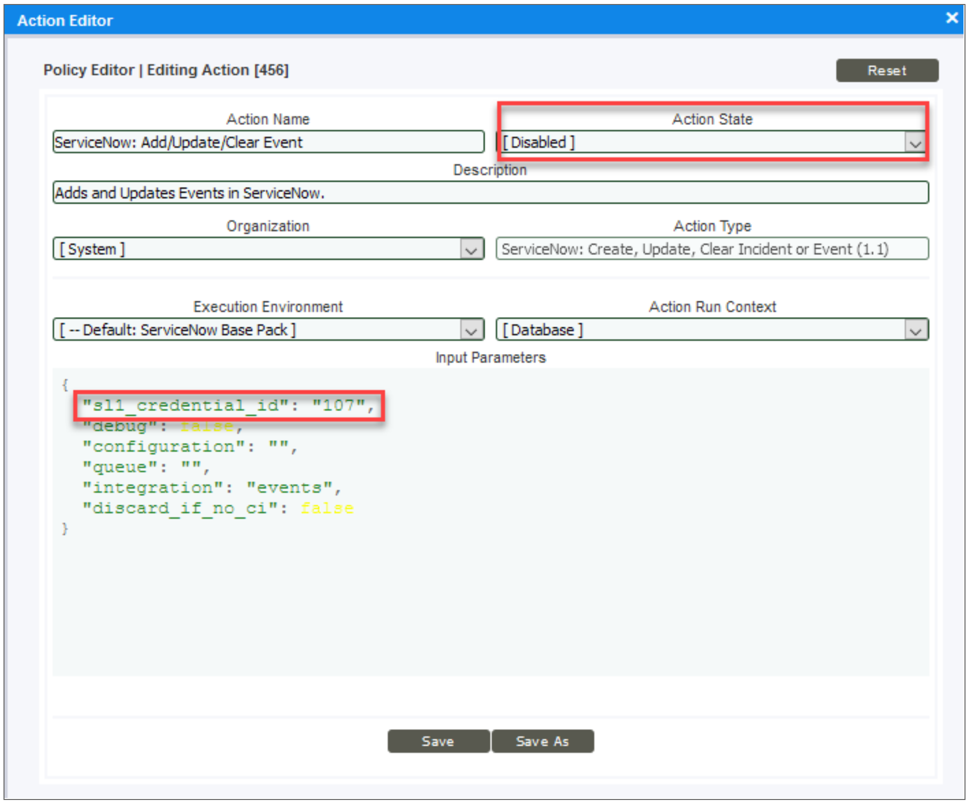
The "ServiceNow: Add/Update/Clear Event" Run Book Action policy contains several default values in the snippet code for the policy that you can customize to use with the *ServiceNow Events* Synchronization PowerPack.

You can edit these values in the **Input Parameters** pane of the **Action Policy Editor** page for this policy.

For example, the ***Correlation Type*** is a value that you can use to correlate an SL1 event with a ServiceNow incident. For the Correlation Type, 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.

To enable and customize the Run Book Automation policy:

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



Policy Editor | Editing Action [456]

Action Name: ServiceNow: Add/Update/Clear Event

Action State: [Disabled]

Description: Adds and Updates Events 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

```
{
  "sl1_credential_id": "107",
  "debug": false,
  "configuration": "",
  "queue": "",
  "integration": "events",
  "discard_if_no_ci": false
}
```

3. For the ***Action State*** filed select *Enabled*.
4. In the **Input Parameters** pane, 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 the Integration Service, but there are still three Automation policies.

You can customize the following values in the "ServiceNow: Add/Update/Clear Incident" 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.
- ***debug***. A true/false value that determines if the action is logged in SL1 and if the application is run in Debug Mode on the PowerFlow service. Troubleshooting logs are written to **/data/tmp/servicenow_rba.log**.
- ***configuration***. Specifies the ID of the configuration object used on the PowerFlow service. 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 PowerFlow service API, make a GET request on this endpoint: `https://<PowerFlow_service_hostname>/api/v1/configurations`.

- ***queue***. Specifies the worker queue on which the application runs. Leave this as default.
- ***discard_if_no_ci***. Specifies whether the PowerFlow service should create 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, the PowerFlow service will *not* create an 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, the PowerFlow service*will* create an incident in ServiceNow and update SL1.

Customizing Logging in the Run Book Action

You can customize the following logging-related items in the "ServiceNow: Add/Update/Clear Event" 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: [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:

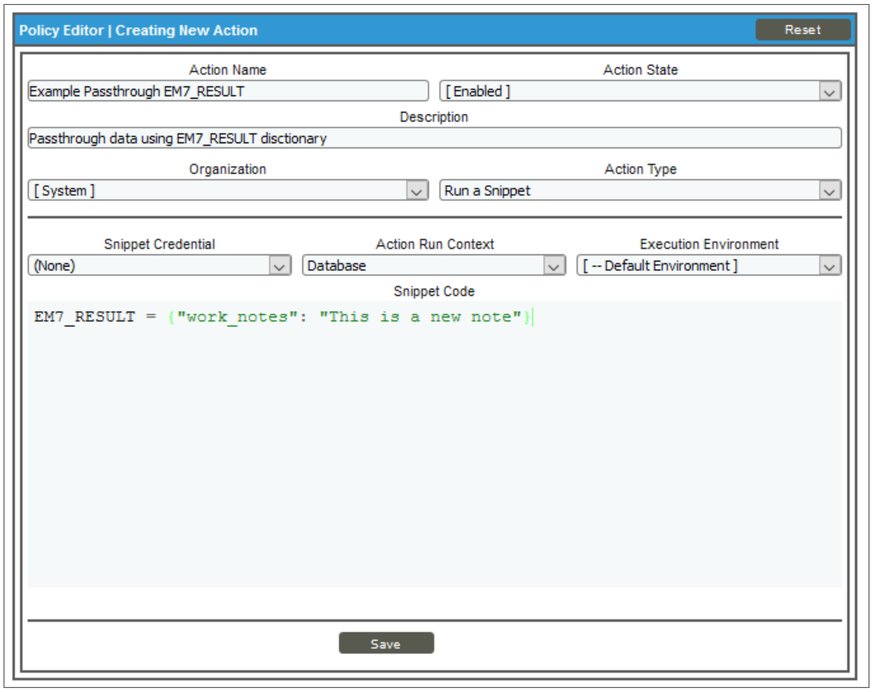
- Create a new Run Book Action that pulls the relevant data and adds it to a dictionary called EM7_RESULT.
- Add the new Run Book Action to the "ServiceNow: [Events] - Add/Update" Run Book Automation Policy, ahead of the "ServiceNow: [Events] - Add/Update/Clear Event" Run Book Action so that the new Action runs first, and then is consumed by the ServiceNow Action.

The following procedure describes how to configure the passthrough functionality.

Passing Custom Data to ServiceNow

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:



Policy Editor | Creating New Action

Action Name: Example Passthrough EM7_RESULT

Action State: [Enabled]

Description: Passthrough data using EM7_RESULT disctionary

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"}
```

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 Event work notes with a hardcoded note:

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

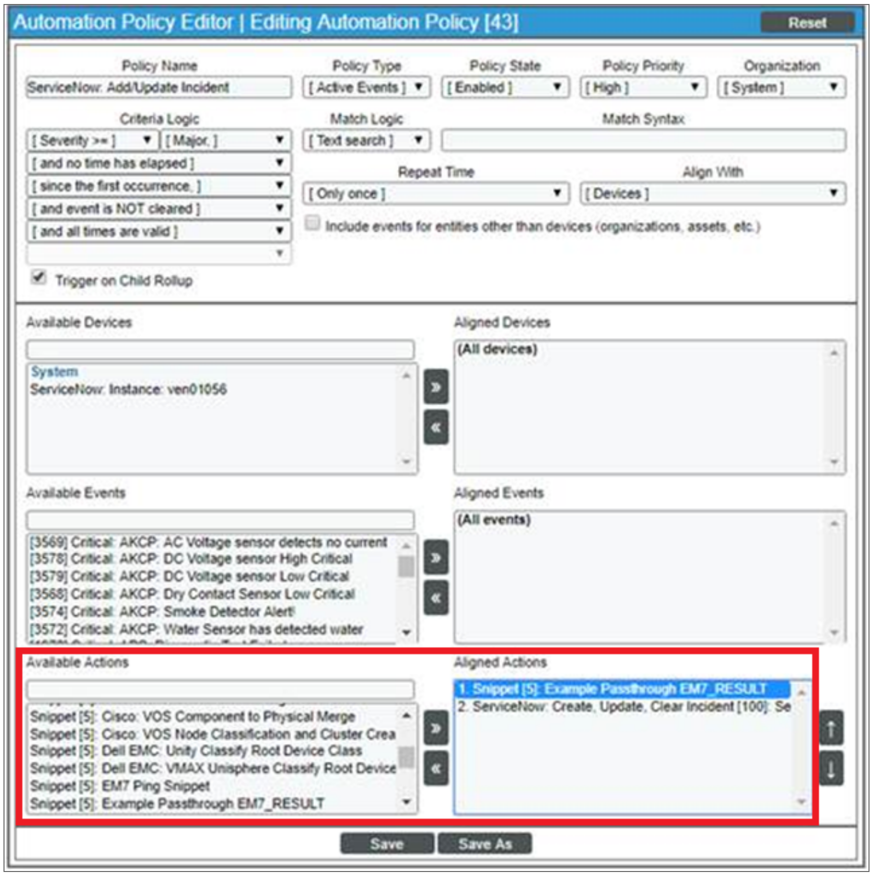
   Additional notes about the structure of the EM7_RESULT dictionary:

   - `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: [Events] - Add/Update" Run Book Automation Policy.

6. In the **Available Actions** section, add the new Run Book Action *before* the "ServiceNow: Create, Update, Clear Event" Run Book Action:



Automation Policy Editor | Editing Automation Policy [43]

Policy Name: ServiceNow: Add/Update Incident
Policy Type: [Active Events]
Policy State: [Enabled]
Policy Priority: [High]
Organization: [System]

Criteria Logic:
[Severity >=] [Major.]
[and no time has elapsed]
[since the first occurrence.]
[and event is NOT cleared]
[and all times are valid]

Match Logic: [Text search]
Match Syntax:
Repeat Time: [Only once]
Align With: [Devices]

□ Include events for entities other than devices (organizations, assets, etc.)

☑ Trigger on Child Rollup

Available Devices:
System
ServiceNow: Instance: ven01056

Aligned Devices:
(All devices)

Available Events:
[3569] Critical: AKCP: AC Voltage sensor detects no current
[3578] Critical: AKCP: DC Voltage sensor High Critical
[3579] Critical: AKCP: DC Voltage sensor Low Critical
[3568] Critical: AKCP: Dry Contact Sensor Low Critical
[3574] Critical: AKCP: Smoke Detector Alert!
[3572] Critical: AKCP: Water Sensor has detected water

Aligned Events:
(All events)

Available Actions:
Snippet [5]: Cisco: VOS Component to Physical Merge
Snippet [5]: Cisco: VOS Node Classification and Cluster Crea
Snippet [5]: Dell EMC: Unity Classify Root Device Class
Snippet [5]: Dell EMC: VMAX Unisphere Classify Root Device
Snippet [5]: EM7 Ping Snippet
Snippet [5]: Example Passthrough EM7_RESULT

Aligned Actions:
1. Snippet [5]: Example Passthrough EM7_RESULT
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 Event" Run Book Action, ensuring that the EM7_RESULT dictionary is passed through to ServiceNow. The "ServiceNow: Create, Update, Clear Event" 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.

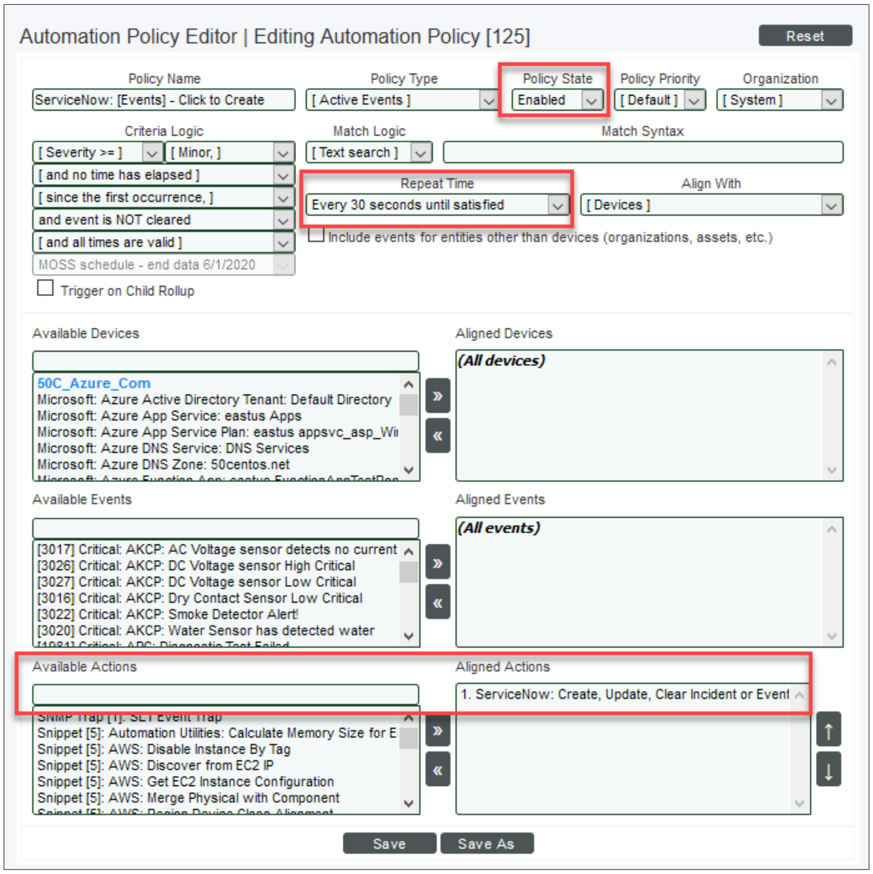
7. You can add additional Run Book Actions to the Run Book Automation Policy for any additional workflows that you might want to run. The Automation Policy execute these Actions in a sequential, top-down order. However, the "ServiceNow: Create, Update, Clear Event" Run Book Action only consumes the EM7_RESULT dictionary from the Run Book Action directly above it.

Configuring the "ServiceNow: [Events] - Click to Create" Automation Policy

The "ServiceNow: [Events] - Click to Create" Run Book Automation policy lets you manually create an alert in ServiceNow by clicking the life-preserver icon (🛟) in SL1. This policy is available in the *ServiceNow Base Pack* PowerPack.

To configure the "ServiceNow: [Events] - Click to Create" 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.
3. Go to the **Automation Policy Manager** page (Registry > Run Book > Automation).
4. Locate the **ServiceNow: [Events] - Click to Create** policy and click its wrench icon (🔧). The **Automation Policy Editor** page appears:



Automation Policy Editor | Editing Automation Policy [125]

Policy Name: ServiceNow: [Events] - Click to Create
Policy Type: [Active Events]
Policy State: Enabled
Policy Priority: [Default]
Organization: [System]

Criteria Logic:
[Severity >=] [Minor,]
[and no time has elapsed]
[since the first occurrence,]
and event is NOT cleared
[and all times are valid]
MOSS schedule - end data 6/1/2020
□ Trigger on Child Rollup

Match Logic: [Text search]
Match Syntax:
Repeat Time: Every 30 seconds until satisfied
Align With: [Devices]
□ Include events for entities other than devices (organizations, assets, etc.)

Available Devices:
50C_Azure_Com
Microsoft: Azure Active Directory Tenant: Default Directory
Microsoft: Azure App Service: eastus Apps
Microsoft: Azure App Service Plan: eastus appsvc_asp_Wi
Microsoft: Azure DNS Service: DNS Services
Microsoft: Azure DNS Zone: 50centos.net

Aligned Devices:
(All devices)

Available Events:
[3017] Critical: AKCP: AC Voltage sensor detects no current
[3026] Critical: AKCP: DC Voltage sensor High Critical
[3027] Critical: AKCP: DC Voltage sensor Low Critical
[3016] Critical: AKCP: Dry Contact Sensor Low Critical
[3022] Critical: AKCP: Smoke Detector Alert!
[3020] Critical: AKCP: Water Sensor has detected water

Aligned Events:
(All events)

Available Actions:
Snippet [5]: Automation Utilities: Calculate Memory Size for E
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

Aligned Actions:
1. ServiceNow: Create, Update, Clear Incident or Event

Configuring the "ServiceNow: [Events] - Click to Create" Automation Policy 36

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]**.

Creating and Aligning a Configuration Object in the PowerFlow Service

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

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

To use this Synchronization PowerPack, you will need to use an existing configuration object n the PowerFlow service user interface or create a new configuration object. Next, you need to align that configuration object to the relevant integration 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 *PowerFlow service: ServiceNow* Synchronization PowerPacks.

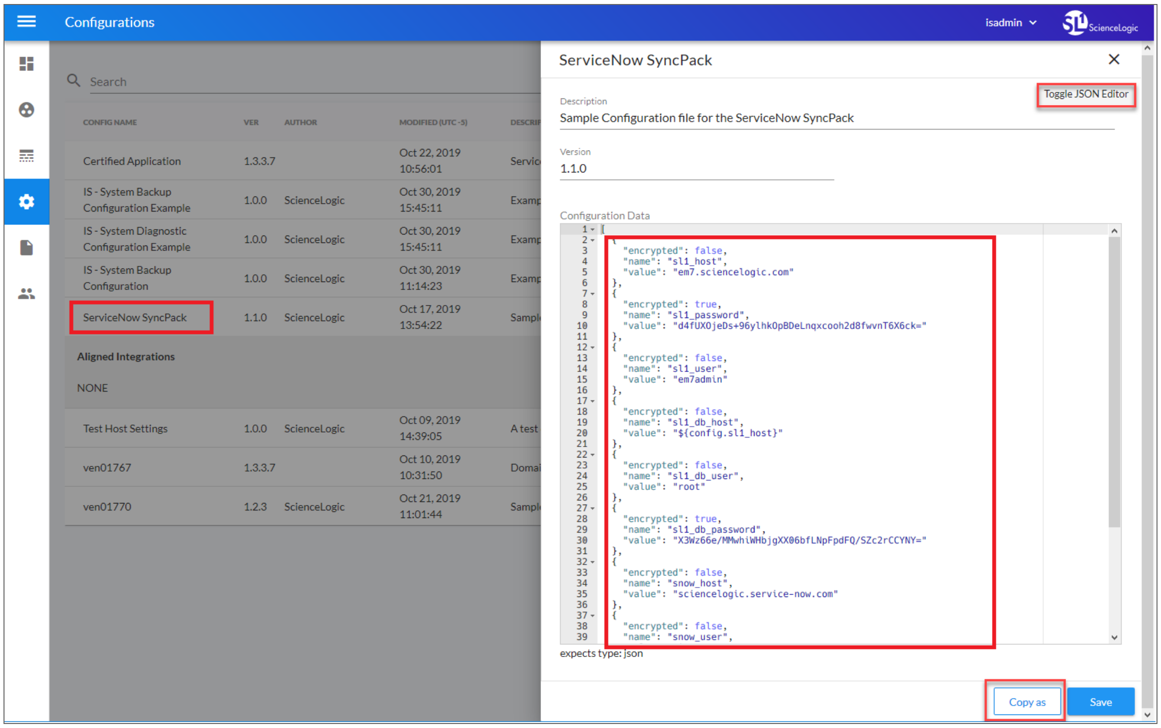
Creating a Configuration Object

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

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

1. In the PowerFlow service user interface, go to the **Configurations** page (⚙).

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



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

Description
Sample Configuration file for the ServiceNow SyncPack

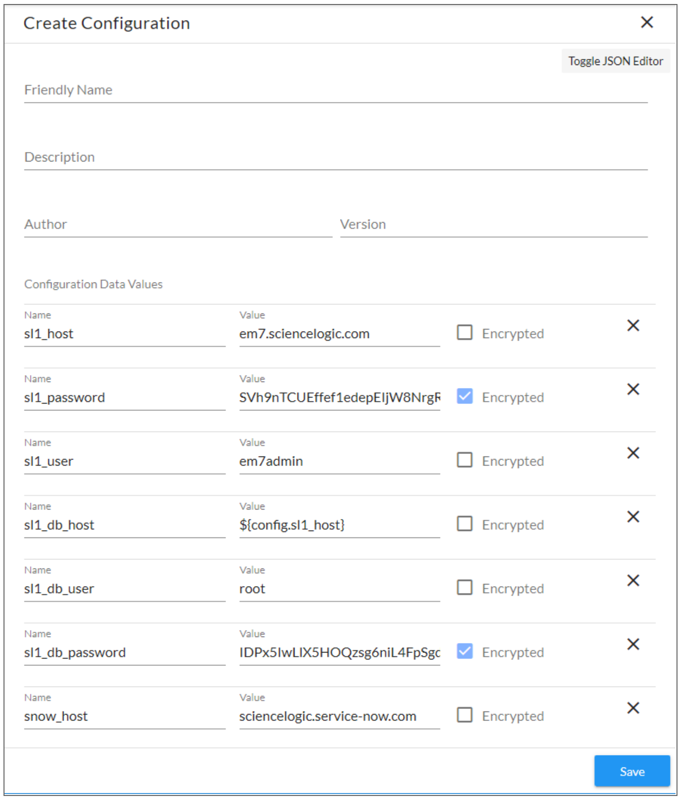
Version
1.1.0

Configuration Data

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

expects type: json

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



Create Configuration

Toggle JSON Editor

Friendly Name

Description

Author

Version

Configuration Data Values

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

Save

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

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

For example:

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

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

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

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

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

8. To create a configuration variable, define the following keys:
   - ***encrypted***. Specifies whether the value will appear in plain text or encrypted in this JSON file. If you set this to "true", when the value is uploaded, the PowerFlow service encrypts the value of the variable. The plain text value cannot be retrieved again by an end user. The encryption key is unique to each PowerFlow service systems. The value is followed by a comma.
   - ***name***. Specifies the name of the configuration file, without the JSON suffix. This value appears in the user interface. The value is surrounded by double-quotes and followed by a comma.
   - ***value***. Specifies the value to assign to the variable. The value is surrounded by double-quotes and followed by a comma.
9. Click **[Save]**. You can now align this configuration object with one or more integration applications.

Aligning a Configuration Object

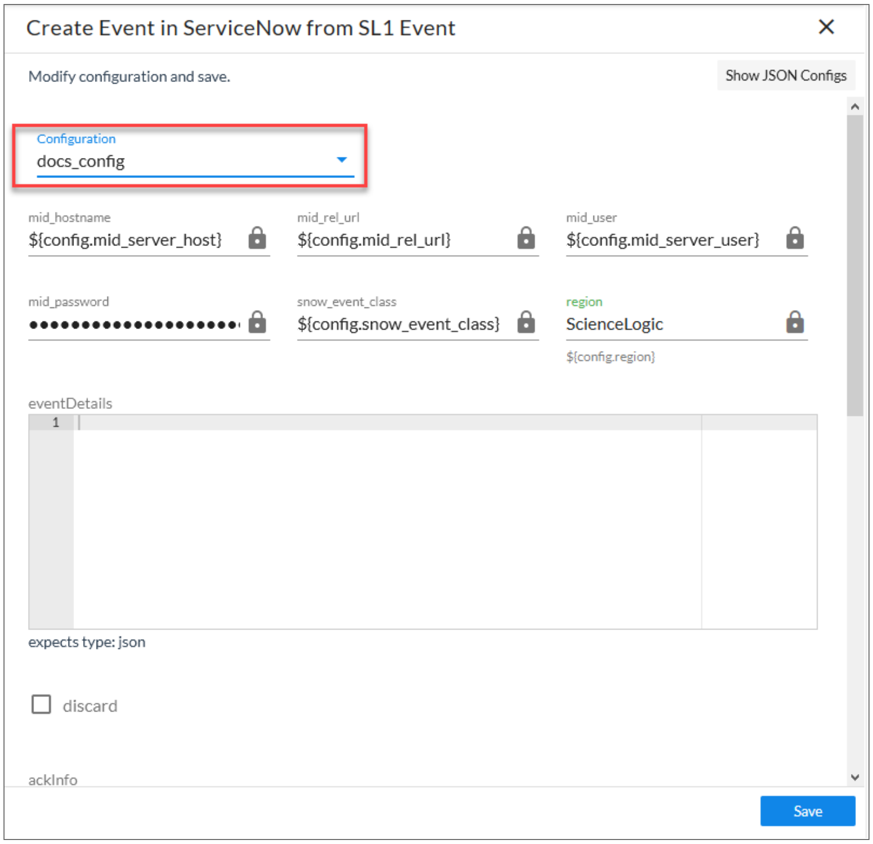
To run Incident Sync, you must "align" the configuration object to run with the following integration applications:

- "Create Event in ServiceNow from SL1 Event"
- "Update SL1 Event from ServiceNow Trigger"
- "Update SL1 Events"

WARNING: Do not manually run or schedule these integration applications. These applications should only be triggered by Run Book Automations in SL1 or by the ServiceNow MID Server.

To align the configuration object with the relevant integration applications:

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



Create Event in ServiceNow from SL1 Event

Modify configuration and save.

Configuration: docs_config

mid_hostname	mid_rel_url	mid_user
${config.mid_server_host}	${config.mid_rel_url}	${config.mid_server_user}

mid_password	snow_event_class	region
••••••••••••••••••••	${config.snow_event_class}	ScienceLogic ${config.region}

eventDetails

expects type: json

□ discard

ackInfo

2. From the ***Configurations*** drop-down, select the configuration object you want to use.
3. Click **[Save]** to align that configuration with the integration application.
4. Wait until you see the "App & Confg modifications saved" pop-up message before proceeding.
5. Repeat this process for the two remain integration applications.

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 integration 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.

Chapter

4

Troubleshooting the ServiceNow Events Synchronization PowerPack

Overview

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

This chapter covers the following topics:

Initial Troubleshooting Steps

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

Integration Service

1. Run `docker service ls` on the PowerFlow service 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

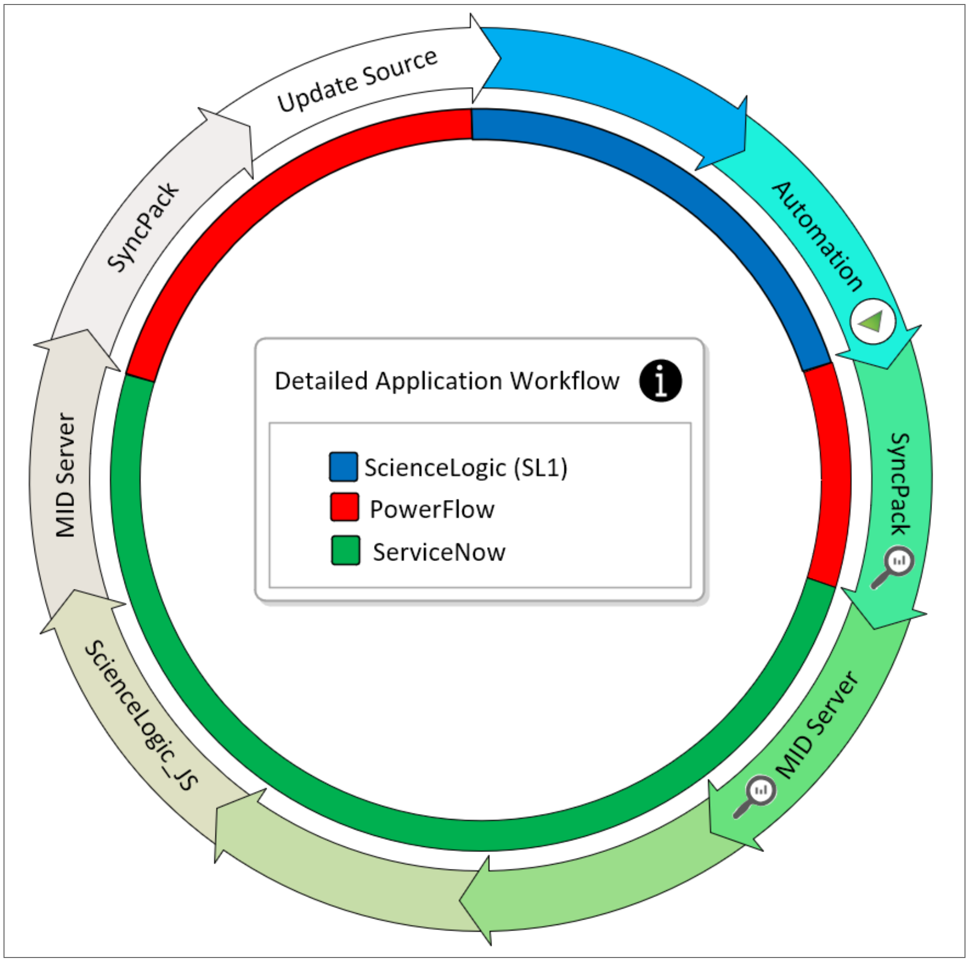
1. Make a note of the ServiceNow version and Synchronization PowerPack version, if applicable.
2. Make a note of whether the user is running an update set or a version of the Certified/Scoped application, if relevant.
3. Make a note of the ServiceNow integration application that is failing on the PowerFlow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

Troubleshooting Event Delivery and Update Sender Transactions

There are two distinct parts to troubleshooting this Synchronization PowerPack as a result of the two different transaction methods for **Event Delivery** and **Update Sender**.

Troubleshooting the Event Delivery Transaction

To aid with troubleshooting, enable the debug flag in the "ServiceNow: Add/Update/Clear Event" Run Book Action in SL1. See the *Automation* step in the following diagram:



Changing the flag in the **Input Parameters** section to `"debug": true,` will allow for additional logging in the SyncPack step, because the SyncPack integration application is triggered, not scheduled. Enabling the debug flag in the Run Book Action expands the debug logging in the SyncPack.

NOTE: There is no return information other than the HTTP status on a submit to the MID Server Instance when delivering event payloads.

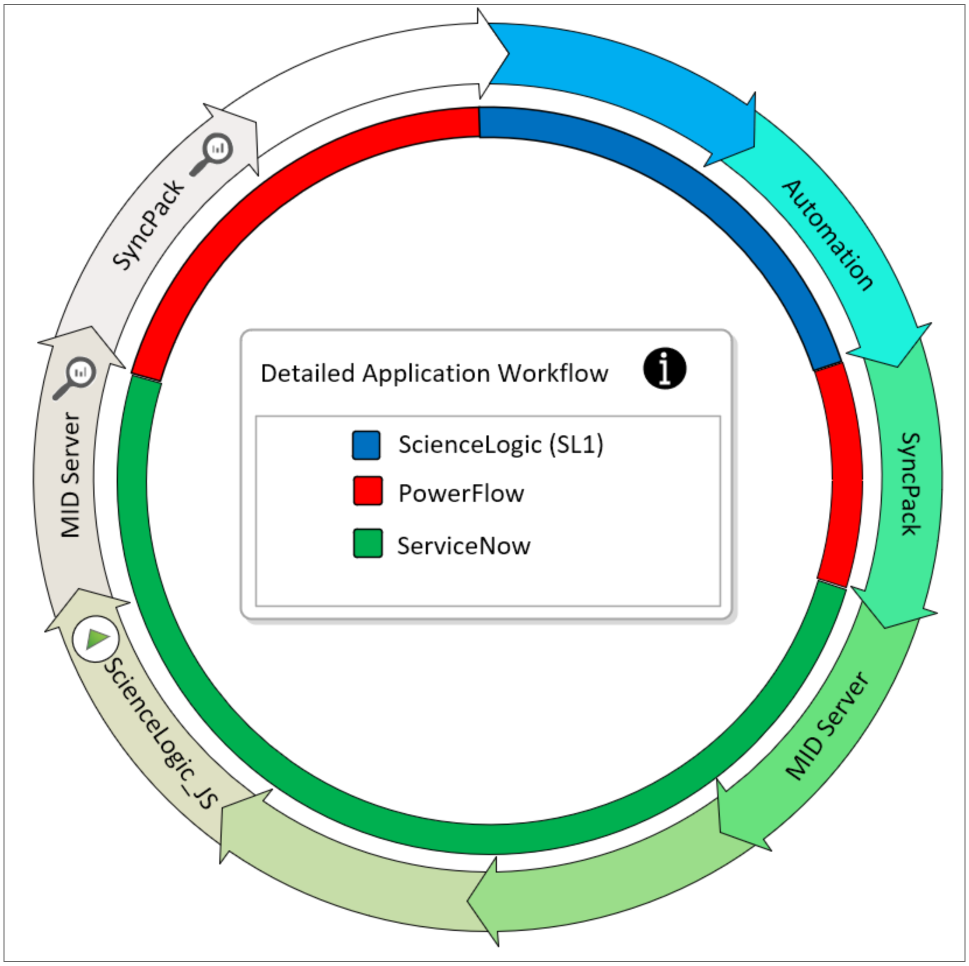
Logging at the *MID Server* step is handled by JSON that is run via the URL used. By default, the Event integration uses the provided ServiceNow generic JSON.

For more information, see:

- Send events via web service
- Event collection via MID using push
- Configure em context extension

Troubleshooting the Update Sender Transaction

You can configure connector parameters with a specific log level to allow for additional logging in the SyncPack step as well as in the MID Server logs. See the *ScienceLogic_JS* step in the following diagram:



> **NOTE:** The log level trigger is defined in the *Sciencelogic_JS* step because the SyncPack integration application is triggered, not scheduled.

MID Server logs can be found on the MID Server in the **agent0.log.0** logs.

For more information, see:

- Access MID Server Logs
- Troubleshooting a MID Server video

Resources for Troubleshooting

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

Useful Integration Service Ports

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

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

Helpful Docker Commands

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

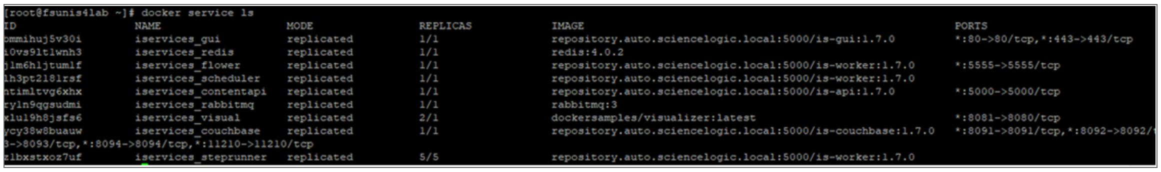
Use the following Docker commands for troubleshooting and diagnosing issues with the PowerFlow service:

Viewing Container Versions and Status

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

```
docker service ls
```

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



```
[root@fsunis4lab ~]# docker service ls
ID                  NAME                     MODE                REPLICAS            IMAGE                                                          PORTS
bmmihuj5v30i        iservices_gui            replicated          1/1                 repository.auto.sciencelogic.local:5000/is-gui:1.7.0           *:80->80/tcp,*:443->443/tcp
i0vs91t1wnh3        iservices_redis          replicated          1/1                 redis:4.0.2
jlm6hljtumlf        iservices_flower         replicated          1/1                 repository.auto.sciencelogic.local:5000/is-worker:1.7.0        *:5555->5555/tcp
lh3pt218lrsf        iservices_scheduler      replicated          1/1                 repository.auto.sciencelogic.local:5000/is-worker:1.7.0
ntimltvg6xhx        iservices_contentapi     replicated          1/1                 repository.auto.sciencelogic.local:5000/is-api:1.7.0           *:5000->5000/tcp
ryln9qgsudmi        iservices_rabbitmq       replicated          1/1                 rabbitmq:3
xlul9h8jsfs6        iservices_visual         replicated          2/1                 dockersamples/visualizer:latest                                *:8081->8080/tcp
ycy38w8buauw        iservices_couchbase      replicated          1/1                 repository.auto.sciencelogic.local:5000/is-couchbase:1.7.0     *:8091->8091/tcp,*:8092->8092/
3->8093/tcp,*:8094->8094/tcp,*:11210->11210/tcp
zlbxstxoz7uf        iservices_steprunner     replicated          5/5                 repository.auto.sciencelogic.local:5000/is-worker:1.7.0
```

Restarting a Service

Run the following command to restart a single service:

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

Stopping all PowerFlow Service Services

Run the following command to stop all PowerFlow service services:

```
docker stack rm iservices
```

Restarting Docker

Run the following command to restart Docker:

```
systemctl restart docker
```

NOTE: Restarting Docker does not clear the queue.

Viewing Logs for a Specific Service

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

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

Some common examples include the following:

```
docker service logs –f iservices_couchbase

docker service logs –f iservices_steprunner

docker service logs –f iservices_contentapi
```

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

Clearing RabbitMQ Volume

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

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

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

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

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

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

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

Re-deploy the PowerFlow service by running the following command:

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

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

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

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

Dynamically Scaling for More Workers

Run the following command:

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

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnosis Tools

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

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

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

Retrieving Additional Debug Information (Debug Mode)

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

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

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

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

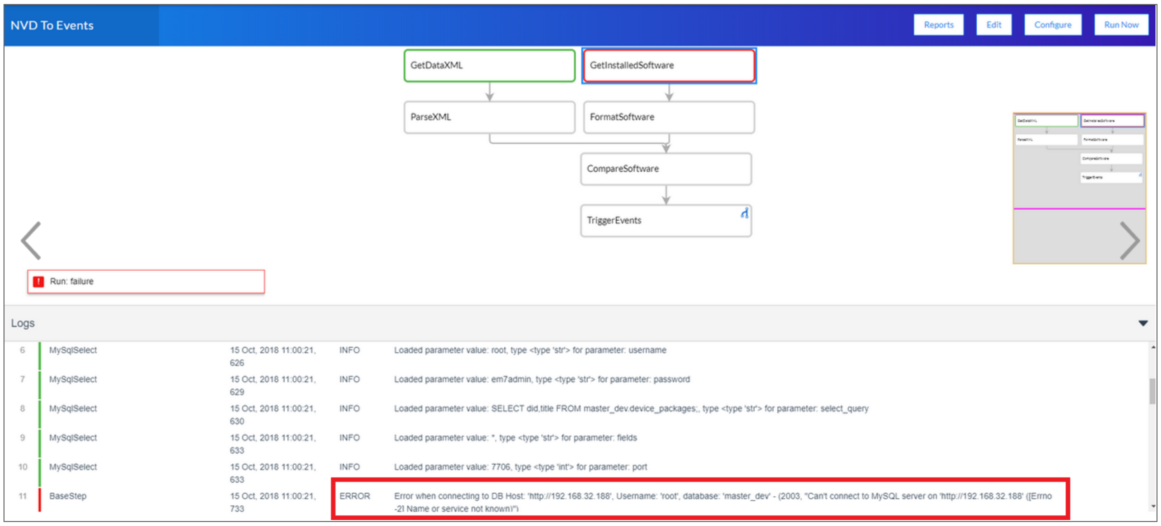
```
https://<integration_service>/api/v1/applications/run
```

Request body:

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

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

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



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

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

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

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

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

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

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

ScienceLogic

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

International: +1-703-354-1010