ScienceLogic

ServiceNow Service Catalog SyncPack

Version 1.1.0 (Documen Revision 1)

Table of Contents

Chapter

1

Introduction to the ServiceNow Service Catalog SyncPack

Overview

This chapter describes the "ServiceNow Service Catalog" SyncPack, which lets you initiate a discovery session to discover devices in SL1 from ServiceNow. This SyncPack also lets you place SL1 devices into and out of maintenance mode using ServiceNow change requests.

This chapter covers the following topics:

What Can I Do with this SyncPack?

The "ServiceNow Service Catalog" SyncPack is the ScienceLogic integration with the ServiceNow Service Catalog Request Application. This SyncPack lets you initiate a discovery session to discover devices in SL1 from ServiceNow. You can also use this SyncPack to place SL1 devices into and out of maintenance mode using ServiceNow change requests.

ScienceLogic created this SyncPack to improve maintainability and certification time. This SyncPack includes the Discovery module, which was previously available with the "ServiceNow CMDB" SyncPack, and the Change module, which was available with the "ServiceNow Change Management" SyncPack.

You can use this SyncPack with or without a ServiceNow MID Server.

This SyncPack includes the following integrations:

- **Discovery Sync**. The Discovery Sync integration lets you initiate a discovery session to discover devices in SL1 from ServiceNow. With Discovery Sync, you start an SL1 discovery session from ServiceNow and then sync the newly discovered SL1 devices or virtual devices and their data with ServiceNow. You can also remove devices from being monitored.

  You will need to configure the following PowerFlow applications to run the Discovery Sync:

    - Sync Catalog Requirements

    - Sync Discovery Session Status from SL1 to ServiceNow

    - Sync Service Requests from ServiceNow to SL1

- **Change Management Sync**. The Change Management Sync integration facilitates event generation and moving devices in and out of maintenance mode in SL1. The SL1 event contains data about the change request from ServiceNow, including a hyperlink to the ServiceNow change record. The event is aligned with a device in SL1, and if the device is part of a business service in SL1 version 12.2.0 or later, the event displays on the **[Changes]** tab for that business service in SL1.

  You will need to configure the following PowerFlow applications to run the Change Management Sync:

    - Sync Catalog Requirements

    - Update SL1 Event With ServiceNow Change Request Info

  If using a MID Server:

    - Create or Update Maintenance Schedule from ServiceNow Trigger

    - Trigger Device Maintenance Updates via MID Server

  If you are not using a MID Server:

    - Sync Maintenance Schedules from ServiceNow to SL1

For more information, see *Configuring Applications for the ServiceNow Service Catalog SyncPack*.

Contents of the SyncPack

This section lists the contents of the "ServiceNow Service Catalog" SyncPack.

PowerFlow Applications

The following applications are included with the "ServiceNow Service Catalog" SyncPack, and you will need to configure these applications before you can run Discovery or Change Management Syncs:

- *Sync Discovery Session Status from SL1 to ServiceNow*. Collects and processes Discovery sessions from SL1, and collects Discovery session logs.

- *Sync Service Requests from ServiceNow to SL1*. Processes Discovery sessions and posts Discovery sessions and new virtual devices to SL1. Also enables device decommissioning for devices you no longer want to monitor.

- *Sync Catalog Requirements*. Sends configuration data from PowerFlow to ServiceNow to use with change management and other processes.

- *Sync Discovery Templates from SL1 to ServiceNow*. Syncs SL1 discovery sessions that contain a configured string to ServiceNow and creates Service Catalog templates in ServiceNow. You can use those templates for discovering or monitoring CIs.

- *Sync Maintenance Schedules from ServiceNow to SL1*. Syncs maintenance windows from ServiceNow change requests (CHG)s to SL1 devices to place the synced devices into maintenance mode for the scheduled change window. If you are scheduling this application, ScienceLogic recommends that you schedule it to run every hour or so, depending on your environment. This process does *not* require the use of a ServiceNow MID Server.

- *Trigger Device Maintenance Updates via MID Server*. Lets you sync maintenance windows from ServiceNow to SL1 using a ServiceNow Management, Instrumentation, and Discovery (MID) Server.

- *Create or Update Maintenance Schedule from ServiceNow Trigger*. Handles scheduling and canceling maintenance schedules in SL1 using a trigger from a ServiceNow Management, Instrumentation, and Discovery (MID) Server. When triggered, the application gathers data about the change request in ServiceNow and generates an SL1 Event with a severity of Notice.

- *Update SL1 Event with ServiceNow Change Request Info*. Updates an SL Event with ServiceNow Change Request info

Of the applications listed above, the Discovery Sync only requires you to configure:

- Sync Catalog Requirements

- Sync Discovery Session Status from SL1 to ServiceNow

- Sync Service Requests from ServiceNow to SL1

The Change Management Sync requires you to configure:

- Sync Catalog Requirements
- Update SL1 Event With ServiceNow Change Request Info

If using a MID Server:

- Create or Update Maintenance Schedule from ServiceNow Trigger
- Trigger Device Maintenance Updates via MID Server

If you are not using a MID Server:

- Sync Maintenance Schedules from ServiceNow to SL1

For more information, see *Configuring Applications for the ServiceNow Service Catalog SyncPack*.

PowerFlow Applications (Internal)

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

- *Cancel Maintenance*. Cancels a scheduled maintenance in SL1.
- *Create Discovery Session in SL1*. Creates and starts a Discovery session in SL1 and updates the ServiceNow service request.
- *Create Maintenance*. Creates a scheduled maintenance in SL1.
- *Create Virtual Device in SL1*. Creates a virtual device in SL1 and updates the Requested Item (RITM) value.
- *Modify Maintenance*. Updates a scheduled maintenance in SL1.
- *Post Discovery Template Data to ServiceNow*. Posts the discovery template date to ServiceNow.
- *Post Attribute Rest Calls to SL1*. Posts attribute REST calls to SL1.
- *Process Create Schedule Requests*. Schedules and posts maintenance tasks in ServiceNow.
- *Process Remove Device Requests from ServiceNow to SL1*.
- *Pull and Post Discovery Logs*. Pulls Discovery session logs from SL1 and posts updates to ServiceNow.
- *Remove Maintenance*. Removes a scheduled maintenance in SL1.
- *Run MySql Queries*. Creates notices for event changes that come in from ServiceNow.
- *Schedule Maintenance*. Creates a scheduled maintenance in SL1.
- *Update Devices and Post Maintenance Status to ServiceNow*. Updates SL1 devices and posts Maintenance Status to ServiceNow.
- *Sync Configs from PowerFlow to ServiceNow*. Syncs configuration data from PowerFlow to ServiceNow.

- ○ *Sync Device Groups from SL1 to ServiceNow*. Collects all device groups and group IDs from SL1 and posts device group data to ServiceNow. To prevent errors when running this application or a device sync, make sure that the device group names are not already being used by existing groups in ServiceNow.

- ○ *Sync Organizations from SL1 to ServiceNow Dependents Table*. Sends SL1 organizations to the Discovery Dependents table in ServiceNow. This is not a bi-directional sync , and SL1 is the source of truth. This sync does not interact with the core company table in ServiceNow.

Steps

- Add External Ticket Reference to SL1Event
- Check SL1 Device Schedules
- Compare Organizations
- Check SL1 Device Schedules
- Delete Tasks or Schedules from SL1
- Disable Devices in SL1
- Parse SL1 Event Details
- Post Discovery Session
- Post Maintenance Task to SL1
- Post Maintenance Status to ServiceNow
- Post Remove Schedule and Task Results
- Post Virtual Device
- Process Change Requests
- Process Maintenance Schedule Triggers
- Process Maintenance Triggers
- Pull and Process Active Discovery Sessions
- Process Collectors and Collector Groups
- Pull and Process Credentials from SL1
- Reconcile SL1 Make and Model with ServiceNow
- Process Device Templates
- Pull and Process Discovery Sessions
- Pull and Process PF Configs
- Pull and Process Service Requests
- Process Virtual Device Classes
- Pull Change Tasks
- Pull Devices and Validate Delete Requests
- Pull Discovery Session Logs from PF Cache and SL1

- Task and Schedule Call to SL
- Update and Process Devices

Chapter

2

Installing the ServiceNow Service Catalog SyncPack

Overview

This chapter lists the prerequisites and describes how to install the "ServiceNow Service Catalog" SyncPack in PowerFlow, along with the other required components.

> **TIP:** In ServiceNow, you can follow the Guided Setup on the "ScienceLogic Integration with Catalog Request" page to configure ServiceNow and PowerFlow for this SyncPack.

This chapter covers the following topics:

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

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 *SL1 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.

In addition, you will need to add the following role to the ServiceNow Integration user to give the user access to the ServiceNow Service Catalog Request Application:

- **x_sclo_sl1_catalog.sl1catalogaccess**

If you need to create templates, add the following role to the ServiceNow Integration user:

- **x_sclo_sl1_catalog.catalog_item_template**

Downloading and Installing 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.

Activating and Installing a SyncPack

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: Service Catalog Automation" Application in ServiceNow

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

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: Service Catalog Automation" application:

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

2. Select the "ScienceLogic SL1: Service Catalog Automation" 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: Service Catalog Automation" 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: Service Catalog Automation" 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. You might need to log out of ServiceNow and log in again to see the updated left-hand navigation menu.

Chapter

3

Configuring Applications for the ServiceNow Service Catalog SyncPack

Overview

This chapter describes how to configure this SyncPack to use SL1 to discover ServiceNow devices. This chapter also explains how to use the SyncPack to place SL1 devices into and out of maintenance mode.

This chapter covers the following topics:

Workflow for Configuring the SyncPack

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

> TIP: In ServiceNow, you can follow the Guided Setup on the "ScienceLogic Integration with Catalog Request" page to configure ServiceNow and PowerFlow for this SyncPack.

Configuring the ServiceNow MID Server

> IMPORTANT: If you are not using a MID Server, go to *Configuring PowerFlow*.

1. *Install the MID Server*
2. *Configure the Connector Definitions*
3. *Configure the Connector Instance*
4. *Configure the MID Web Server*
5. *Run an external test of the MID Server configuration*
6. *Review the MID Server Script Include (ScienceLogic_JS)*

Configuring PowerFlow

1. *Create a configuration object in the PowerFlow user interface*
2. *Align the new configuration file with the relevant set of PowerFlow applications* (depending on whether you use a ServiceNow MID Server)
3. *Configure and Run the Catalog Requirements Sync*
4. *Configure the Discovery Sync*
5. *Configure the Change Management Sync*
6. Schedule the applications as needed

Configuring the ServiceNow MID Server

> IMPORTANT: If you are not using a MID Server, skip this topic and go to *Configuring PowerFlow*.

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 Service Catalog" SyncPack.

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.

> **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]**.

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

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

   - *Javascript to run*. Attach the JavaScript file provided by the Certified application: [ScienceLogic SL1: Event Integration](#).

   - *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. For more information, see *External Testing of the MID Server Configuration.*

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

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]**.

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.

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.

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, PowerFlow, 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**.

The following topics cover how to set up your SL1 instance for the "ServiceNow Service Catalog" SyncPack.

> **IMPORTANT:** You must install the "ServiceNow Base Pack" PowerPack before you can perform the following procedures. For more information, see *Installing the ServiceNow Base Pack PowerPack*.

Configuring PowerFlow

The following topics cover how to set up your PowerFlow instance for the "ServiceNow Service Catalog" 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:
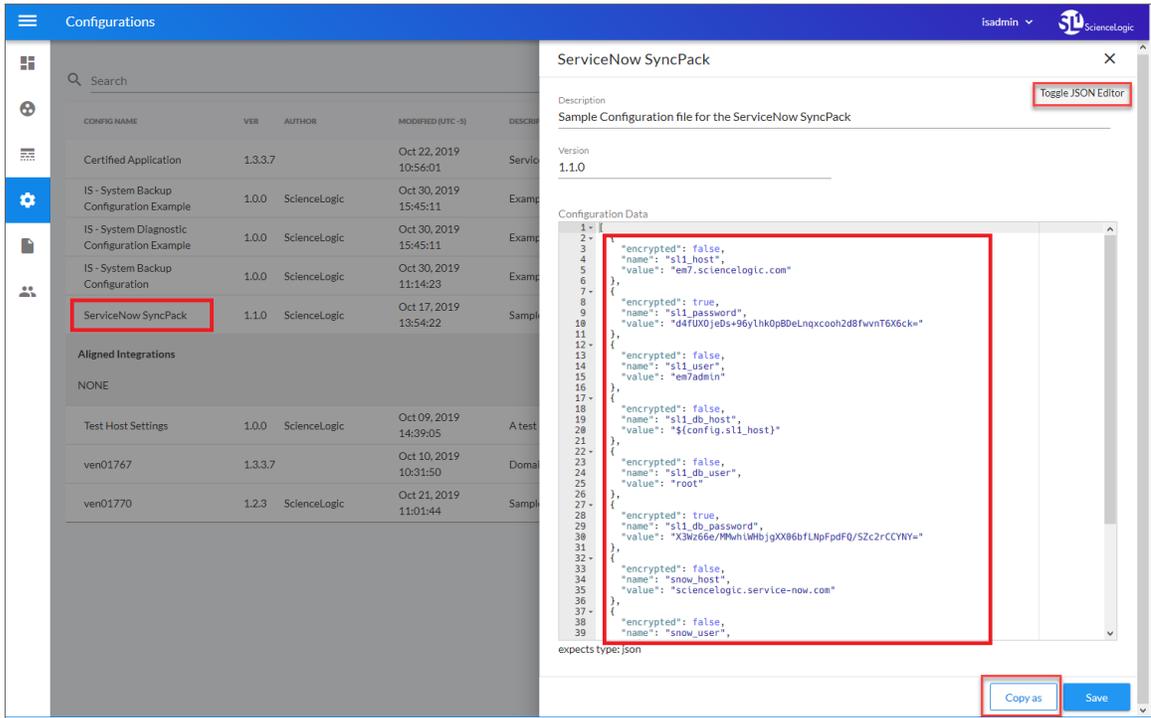
> **TIP:** Click **[Toggle JSON Editor]** to show the JSON code. Click the button again to see the fields.

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.

Discovery Sync

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

To run a Discovery Sync session:

1. In PowerFlow, run the *"Sync Catalog Requirements" application*.

2. In ServiceNow, configure a service request for Discovery Sync. For more information, see *Configuring a ServiceNow Service Request for Discovery Sync*.

3. In PowerFlow , run the applications listed in the *Discovery Sync Workflow*.

Running the Catalog Requirements Sync

To prepare SL1 and ServiceNow for a Discovery Sync, run the "Sync Catalog Requirements" application in the PowerFlow user interface. This application exports information from SL1 to populate the information in the Discovery Dependent table and ServiceNow request form. You must run this application before you can create a discovery sync or change management session in ServiceNow.

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

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

> NOTE: The *region* field is populated by the configuration object you aligned with this application. The region value must match the value in the *SL1 Region* field in ServiceNow. If you need to update this value, you will need to define the *region* variable in the configuration object that is aligned with this application, or align a different configuration object that has the correct *region* value.

- *snow_request_limit*. Specify the number of CIs fetched from ServiceNow in each request. The default is 500 objects per chunk.

- *read_timeout*. Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 30 seconds.

- *sl1_chunk_size*. Specify the number of objects to include in each chunk sent from SL1 to ServiceNow. The default is 500.

- *snow_chunk_size*. Specify the number of objects to send in each chunk from ServiceNow. The default is 150 objects per chunk.

> TIP: As a best practice, schedule the "Sync Catalog Requirements" application to run every hour or so, depending on your environment.

Configuring a ServiceNow Service Request for Discovery Sync

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

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

To configure the ServiceNow service requests for Discovery Sync:

1. In ServiceNow, go to the **Catalog Items** page (Service Catalog > Catalog Definitions > Maintain Items).

2. Type "ScienceLogic" in the *Category* column. The **Device Discovery** and **Monitoring Removal** service requests appear in the search results.

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

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

---

4. If you need to update these fields, click the **[Edit in Catalog Builder]** button at the top of the detail page.

5. Update the fields and click the **[Update]** button to save your changes.

6. From the **Catalog Items** page, click the check box for the **Device Discovery** service request and click the **[Activate]** button at the bottom of the **Catalog Items** window.

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

---

7. Navigate to the relevant catalog for the service request. For example, if you selected *Service Catalog* for one or both requests, then type "Service Catalog" in the filter navigator, or select **Self-Service > Service Catalog** to view the new service requests. Type "device discovery" in the *Search catalog* field to quickly locate the request.

8. Configure and run the applications listed in the *Discovery Sync Workflow* before creating the Device Discovery catalog request in ServiceNow.

Discovery Sync Workflow

1. *Sync Service Requests from ServiceNow to SL1*. This application sends the request forms to SL1. This application uses one or more of the following options from the **Configuration** pane:

   - *Configuration*. Select the relevant configuration object to align with this application.

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

- *Open_State*. The State ID from ServiceNow that specifies which Requested Items (RITMs) to pull and process. The default is 1.
- *Closed_Success_State*. The State ID for a successfully created virtual device. The State ID for a successful run changes from 1 to 2 and then ends with 4. The default is 3.
- *Closed_Failed_State*. The State ID for failed discoveries or failed virtual device creation, usually caused by invalid payloads. The State ID for a failed run changes from 1 to 2 and then ends with 4. The default is 4.
- *In_Progress_State*. The State ID for RITMs for a running discovery. The default is 2.
- *target_vcug*. Leave this field blank.
- *verify_snow_ssl*. Toggle on (blue) this option to enable verification of the SSL certification when you run this application.
- *recursively_disable_children*. Leave this field blank.

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

- *Configuration*. Select the relevant configuration object to align with this application. Required.
- *read_timeout*. Specify the maximum amount of time in seconds that the application should wait for a response before timing out. The default is 20 seconds.
- *retry_max*. The maximum number of times that 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_backoff_max*.The maximum time interval for the *retry_backoff* option, in seconds. For example, if you have *retry_max* set to 15, the delays will be 1, 2, 4, 8, 16, 32, 64, 120, 240, 480, 600, 600, 600, 600, and 600. The default is 600.
- *Closed_Success_State*. The State ID for a successfully created discovery. The State ID for a successful run changes from 1 to 2 and then ends with 4. The default is 3.
- *Closed_Failed_State*. The State ID for failed discoveries, usually caused by invalid payloads. The State ID for a failed run changes from 1 to 2 and then ends with 4. The default is 4.
- *sys_id_target*. Takes the *sys_id* value from the CI in the ServiceNow Service Request and populates it in the relevant field in SL1, such as *c-sys_id*.
- *ci_class_target*. Takes the *ci_class* value from the CI in the ServiceNow Service Request and populates it in the relevant field in SL1, such as *c-ci_class_target*.

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

- *retry_jitter*. When selected, instead of using a defined interval between retries, the PowerFlow system will retry the step execution at random intervals. By default, this option is not selected.

- *retry_backoff*. When selected, instead of using a defined interval between retries, PowerFlow will incrementally increase the interval between retries. By default, this option is not selected.

- *verify_snow_ssl*. Toggle on (blue) this option to enable verification of the SSL certification when you run this application.

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

   - *Configuration*. Select the relevant configuration object to align with this application. Required.

   - *sl1_chunk_size*. Specify the number of object to include in each chunk sent from SL1 to ServiceNow. The default is 500.

   - *snow_chunk_size*. Specify the number of objects to send in each chunk from ServiceNow. The default is 150 objects per chunk.

   - *template_prefix*. Specify the prefix string that PowerFlow will search for in SL1. Any Discovery Sessions that contain that string will be used in ServiceNow to create a service catalog template. The default string is **ServiceNow Template:**, but you can configure this as needed. In SL1, go to the **Discovery Sessions** page (Devices > Discovery Sessions) and search for the discovery session or sessions that you want to use as a template. The start of the name in the **Name** field should match the value in the **template_prefix** field, above.

   - *read_timeout*. Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 30 seconds.

   - *verify_snow_ssl*. Toggle on (blue) this option to enable verification of the SSL certification when you run this application.

4. Run all three applications in PowerFlow, in the order listed above. When the applications finish running, PowerFlow sends the status of those applications to ServiceNow.

Creating a Discovery Request in ServiceNow

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

To create a discovery request in ServiceNow:

1. In ServiceNow, go to the **Catalog Request** page (Self-Service > ScienceLogic SL1: Catalog Automation > Catalog Request). The Describe Needs page will appear.

2. Select the SL1 environments you would like to initiate device discovery for under **Select SL1 PowerFlow platform**. You can select multiple environments by holding the "Shift" or "Ctrl" key on your keyboard and then clicking the environments you want to use to discover devices.

3. Click the right arrow button to add the selected environments to the **Selected** table.

4. Click the **[Choose Options]** button. The **Choose Options** page appears.

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

   ○ If you selected *Discover Device(s)*, go to step 5.

   ○ If you selected *Create Virtual Device*, go to step 6.

6. If you previously created a catalog item template that contains your device discovery information, select it from the **Select Template** field.

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

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

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

     ■ One or more *single IPv4 addresses* separated by commas and a new line. Each IP address must be in standard IP notation and cannot exceed 15 characters. For example, "10.20.30.1, 10.20.30.2, 10.20."

     ■ One or more *ranges of IPv4 addresses* with "-" (dash) characters between the beginning of the range and the end of the range. Separate each range with a comma. For example, "10.20.30.1 – 10.20.30.254".

     ■ One or more IP address ranges in *IPv4 CIDR notation*. Separate each item in the list with a comma. For example, "192.168.168.0/24".

     ■ One or more hostnames (fully-qualified domain names). Separate each item in the list with a comma.

   - **Credentials**. Select one or more SNMP credentials to allow SL1 to access a device's SNMP data and click the right arrow button to add them to the **Selected** table.

   - **Discover Non-SNMP**. Specifies whether or not SL1 should discover devices that don't respond to SNMP requests.

- *Model Devices*. Determines whether or not the devices that are discovered with this discovery session can be managed through SL1.

- *DHCP*. Specifies whether or not the specified range of IPs and hostnames use DHCP. If you select this option, SL1 performs a DNS lookup for the device during discovery and each time SL1 retrieves information from the device.

- *Device Model Cache TTL (h)*. Amount of time, in hours, that SL1 stores information about devices that are discovered but not modeled, either because the **Model Devices** option is not enabled or because SL1 cannot determine whether a duplicate device already exists. The cached data can be used to manually model the device from the **Discovery Session** window.

- *Collection Server*. Select an existing collector to monitor the discovered devices. Required.

- *What company is this for?*. Specify the company that will use this discovery data. Click the magnifying glass icon to locate a company. The **Region** field updates automatically based on the company you select.

- *Add Devices to Device Groups*. Select one or more existing device groups to which you want to add the discovered devices.

- *Apply Device Template* . Select an existing device template if needed. As SL1 discovers a device in the IP discovery list, that device is configured with the selected device template.

- *Initial Scan Level*. For this discovery session only, specifies the data to be gathered during the initial discovery session.

- *Scan Throttle*. Specifies the amount of time a discovery process should pause between each specified IP address (specified in the *IP Address/Hostname Discovery List* field). Pausing discovery processes between IP addresses spreads the amount of network traffic generated by discovery over a longer period of time.

- *Scan Default Ports*. Select this option to scan the default ports: 21,22,23,25,80. If you de-select this option, you can specify a different list of ports in the **Custom Port Scan** field that appears.

- *Port Scan All IPs*. For the initial discovery session only, specifies whether SL1 should scan all IP addresses on a device for open ports.

- *Port Scan Timeout*. For the initial discovery session only, specifies the length of time, in milliseconds, after which SL1 should stop trying to scan an IP address for open ports and begin scanning the next IP address (if applicable).

- *Interface Inventory Timeout (ms)*. Specifies the maximum amount of time that the discovery processes will spend polling a device for the list of interfaces. After the specified time, SL1 will stop polling the device, will not model the device, and will continue with discovery. The default value is 600,000 ms (10 minutes).

- *Maximum Allowed Interfaces*. Specifies the maximum number of interfaces per devices. If a device exceeds this number of interfaces, SL1 stops scanning the device, will not model the device, and will continue with discovery. The default value is 10,000.

- *Bypass Interface Inventory*. Select this option if you do *not* want SL1 to attempt to discover interfaces for each device in the discovery session.

8. If you selected *Create Virtual Device* in the **Request Type** field, complete the following fields:

- *Name*. Type a name for the virtual device.

- *Virtual Device Class*. Specify the device class of the virtual device. Click the magnifying glass icon to locate any classes aligned with your organization.

- *Collector Group*. Specify the SL1 collector group to use for the Discovery Sync. Click the magnifying glass icon to locate any collector groups aligned with your organization.

- *What company is this for?*. Specify the company that will use this discovery data. Click the magnifying glass icon to locate a company. The *Region* field updates automatically based on the company you select.

9. Click the **[Checkout]** button. On the **Checkout** page that appears, make a note of value in the *Request Number* field.

10. In the PowerFlow user interface, go to the **Applications** page and run the "Sync Service Requests from ServiceNow to SL1" application. For more information, see *Discovery Sync Workflow*.

11. When the application completes, go to **Self-Service > My Requests** in ServiceNow.

12. Click the **RITM** record link to go to the **Requested Item** page. The **State** field should update to *Closed Complete* and the request should be assigned to itself.

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

    - Scroll down to the **Activities** pane to verify that you have a comment stating the discovery completed.

    - In SL1, navigate to the **Discovery Control Panel** page (Registry > Manage > Discovery) and verify that SL1 created a new discovery session with that ID.

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

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

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

> **TIP**: As a best practice, schedule the "Sync Service Requests from ServiceNow to SL1" application to run every hour or so, depending on your environment.

Decommissioning Devices

If you want to quickly select one or more devices in ServiceNow to remove from monitoring (or "decommission") in SL1, you can use the *Device Removal* option from the Catalog Request form.

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

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

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

Activating the ServiceNow Service Request for Monitoring Removal

To activate the ServiceNow service request for Device Removal:

1.   1.   In ServiceNow, go to the **Catalog Request** page (Self-Service > ScienceLogic SL1: Catalog Automation > Catalog Request). The Describe Needs page will appear.

2.   Select the SL1 environments you would like to remove devices from under *Select SL1 PowerFlow platform*.

> **TIP:** You can select multiple environments by holding the "Shift" or "Ctrl" key on your keyboard and then clicking the environments to remove devices for.

3.   Click the right arrow button to add the selected environments to the *Selected* table.

4.   Click the **[Choose Options]** button. The **Choose Options** page appears.

5.   In the *Request Type* field, select *Device Removal*.

6.   Specify the company that will use this discovery data in the *What company is this for?* . Click the magnifying glass icon to locate a company. The *Region* field updates automatically based on the company you select.

7.   Under *Devices you want to remove from monitoring*, select the devices you want to remove. Click the right arrow to move them to the *Selected* table.

8.   Click the **[Checkout]** button. On the **Checkout** page that appears, make a note of value in the *Request Number* field.

9. In the PowerFlow user interface, select the "Sync Service Requests from ServiceNow to SL1" application from the **Applications** page and click **[Configure]** on the application detail page. The **Configuration** pane appears:

10. Complete the following fields:

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

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

- *recursively_disable_children*. Check this option to move all child devices of the devices you are decommissioning to the VCUG. If this option is not checked and a parent device is in the disable request, the parent device will be skipped with a warning message.

- *target_vcug*. Specify the ID of the SL1 Virtual Collection Group (VCUG) you created to hold the devices on the **Collector Group Settings** page (System > Settings > Collector Groups). If this value is null, the application will attempt to pull the value from the *target_vcug* field in the "Delete Devices from SL1" application.

11. Click **[Save]**. The **Configuration** pane automatically closes after this message appears.

12. Click **[Run]** to run the application.

> **TIP**: As a best practice, schedule the "Sync Service Requests from ServiceNow to SL1" application to run every hour or so, depending on your environment.

Configuring Change Management Sync

The Change Management Sync integration lets you configure SL1 to generate an event when a maintenance-related change occurs. The SL1 event contains data about the change request from ServiceNow, including a hyperlink to the ServiceNow change record. The event is aligned with a device in SL1, and if the device is part of a business service in SL1 version 12.2.0 or later, the event displays on the **[Changes]** tab for that business service in SL1.

You will need to configure the following PowerFlow applications to run the Change Management Sync:

- Create or Update Maintenance Schedule from ServiceNow Trigger

- Sync Catalog Requirements

- Sync Maintenance Schedules from ServiceNow to SL1Trigger Device Maintenance Updates via MID Server

- Update SL1 Event With ServiceNow Change Request Info

Syncing Device Maintenance from ServiceNow to SL1

You can use the following methods to put one or more devices into maintenance mode from ServiceNow to SL1:

- Use the "Sync Maintenance Schedules from ServiceNow to SL1" PowerFlow application if you are *not* using a MID Server, and you want to create a change request in ServiceNow to perform *scheduled* maintenance through a maintenance window in ServiceNow. For more information, see *Scheduling Device Maintenance (No MID Server)*.

- If you are looking to leverage a ServiceNow Management, Instrumentation, and Discovery (MID) Server, there are two separate ways of placing a Device in maintenance:

    ○ Use the "Create or Update Maintenance Schedule from ServiceNow Trigger" application to create and cancel maintenance schedules in SL1 from a ServiceNow MID Server trigger. When triggered, the application gathers data about the change request in ServiceNow and generates an SL1 Event with a severity of Notice.

    ○ Use the "Trigger Device Maintenance Updates via MID Server" application if you are using a ServiceNow MID Server and you want to *immediately* enable or disable maintenance on a device.

    For more information, see *Using a MID Server to Place a Device Directly in Maintenance or Schedule a Maintenance Window*.

Viewing SL1 Events for Change Requests

When the "Create or Update Maintenance Schedule from ServiceNow Trigger" application is triggered, the application gathers data about the change request in ServiceNow and generates an SL1 Event with a severity of Notice.

For an SL1 Event that was generated by an update to a change request in ServiceNow, you can click the link for the ticket number in the **Ticket External Reference** column on the **Events** page (or click the life-ring icon (🛟) in the classic user interface) to go back to the change request in ServiceNow:

For more information about SL1 events, see the *Events* manual.

Viewing Change Events for SL1 Business Services

If you are using SL1 version 11.1.0 or later with the Change Events feature enabled, and the device aligned with the event is in an SL1 Business Service, the event will display on the **[Change Events]** tab for that Business Service in SL1:

The **Change Events** tab displays a list of events that are created when PowerFlow pulls change data from ServiceNow, including both active and cleared change events.

For more information about SL1 Business Services, see the *Business Services* manual.

Scheduling Device Maintenance (No MID Server)

You can create a change request in ServiceNow to perform *scheduled* maintenance through a maintenance window in ServiceNow. You use the "Sync Maintenance Schedules from ServiceNow to SL1" PowerFlow

application to sync maintenance windows from ServiceNow change requests (CHG)s to SL1 devices to place the synced devices into maintenance mode for the scheduled change window.

This process does *not* require the use of a ServiceNow MID Server.

> **WARNING**: PowerFlow only syncs maintenance schedules that are aligned with devices that are already synced with ServiceNow. Before setting up maintenance schedule sync, you must first sync devices between SL1 and ServiceNow using the *ServiceNow CMDB* SyncPack.

When you create a change request (CHG) in ServiceNow and sync it to SL1, the scheduled maintenance that SL1 creates has collections disabled by default. This is not configurable.

The SL1 Scheduler supports maintenance windows of at least one minute or more.

> **NOTE**: If you update the scheduled times in a ServiceNow change request, you will need to cancel the change request, which also cancels the maintenance window in SL1. You will then need to create a new change request with the new time window and sync that change request to SL1.

To set up maintenance sync:

1. In ServiceNow, type "change" in the filter navigator and navigate to **Change > Create New**.

2. Click **[New]** to create a new change request of type "Normal". A new Change Request record appears:

Scheduling Device Maintenance (No MID Server)
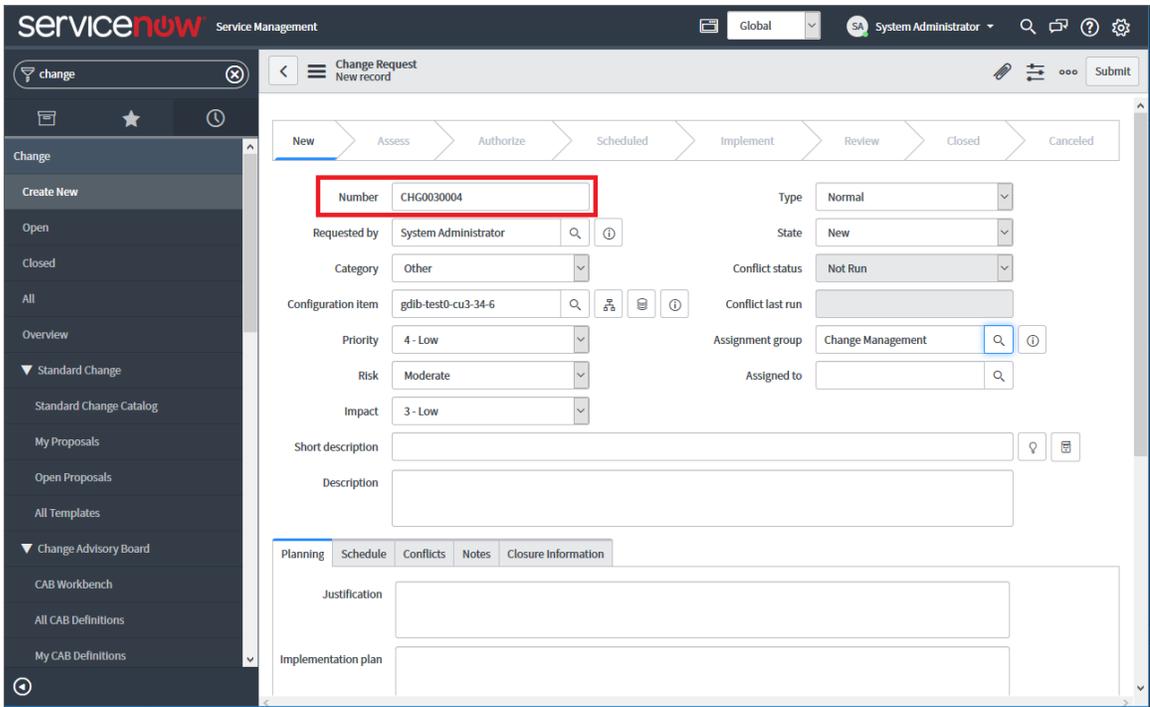
3. Make a note of the change request number in the **Number** field. You will use this later to verify that the maintenance sync was created. In this example, the value is *CHG0030004*.

4. Update the following fields in the record:

   - *Configuration Item*. Select the CI you want to configure for maintenance sync.

   - *Assignment group*. Select the group for the CI.

   > **NOTE**: The aligned CI must have the *SL1 Monitored* field selected before the PowerFlow can use the maintenance schedule for that CI.

5. Click the **[Submit]** button. The change request is saved, and you are returned to the **Change Requests** page.

6. Select the change request you just created, and in the change request record, right-click the **State** label and select *Show Choice List*. The Choices list displays a list of the configurable choices and values:

		Table ▲	Element	Language	Value	Label	Inactive	Sequence	Updated
	ⓘ	change_request	state	en	-5	New	false	1	2015-04-24 14:29:54
	ⓘ	change_request	state	en	-4	Assess	false	2	2015-04-24 15:07:16
	ⓘ	change_request	state	en	-3	Authorize	false	3	2015-04-24 15:07:23
	ⓘ	change_request	state	en	-2	Scheduled	false	4	2015-04-24 15:07:32
	ⓘ	change_request	state	en	-1	Implement	false	5	2015-04-24 15:13:43
	ⓘ	change_request	state	en	0	Review	false	6	2015-04-24 15:13:54
	ⓘ	change_request	state	en	3	Closed	false	7	2015-04-24 14:31:24
	ⓘ	change_request	state	en	4	Canceled	false	8	2015-04-24 14:32:46

> **NOTE**: You need Administrator privileges to access this list.

7. Make a note of the values in the *Value* and *Label* fields. These values map to the ***New_Change_Request_State*** and ***Canceled_Change_Request_State*** fields in the "Sync Maintenance Schedules from ServiceNow to SL1" application.

8. Return to your new change request and scroll down in the change request to the **[Affected CIs]** tab, where you can click the **[Add]** button to add additional synced CIs to the maintenance sync:
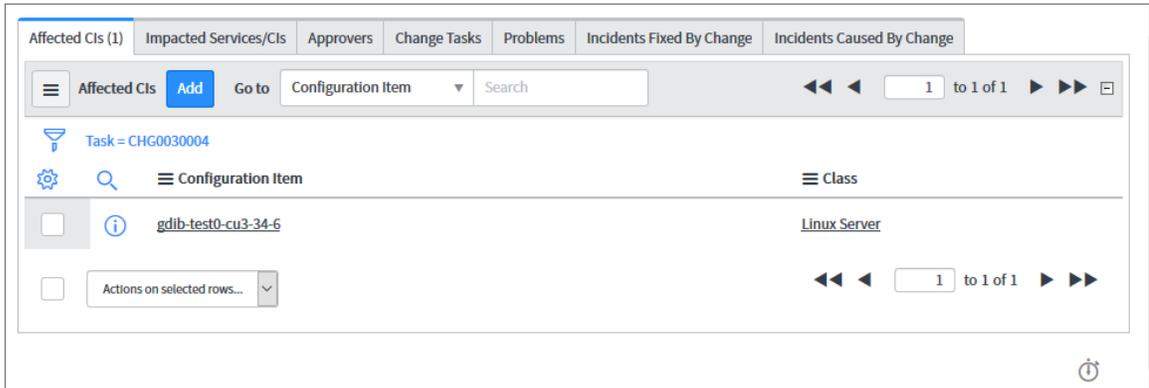
9. In the PowerFlow user interface, go to the **Applications** page and select the "Sync Maintenance Schedules from ServiceNow to SL1" application.

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

11. As needed, update the following options from the **Configuration** pane:

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

- *New_Change_Request_State*. The State ID from ServiceNow of the scheduled change request that this application accesses to pull to schedule maintenance windows in SL1.The default is -2.

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

- *Canceled_Change_Request_State*: The State ID for a canceled change request that this application accesses to pull to schedule maintenance windows in SL1. The default is 4.

- *New_Change_Task_State*: The State ID of the scheduled change task that this application accesses to pull to cancel maintenance windows in SL1. The default is 1.

- *Canceled_Change_Task_State*: The State ID for a canceled change task that this application accesses to pull to cancel maintenance windows in SL1. The default is 4.

- *Process_Change_Tasks*: Select this option to enable change task processing, which allows the request to be broken down into Change Tasks, which results in two extra pulls of the data. The default is unselected.

- *collection_polling*. When selected, leaves synced devices in a collection enabled state after a synced schedule completes. The devices will be put in maintenance but the devices will remain enabled for polling. By default, the devices are left in a collection disabled state.

- *patch_window*. Specifies a patch window from 0 to 60 minutes.

12. Verify that the value from the *New_Change_Request_State* field matches the value in the *Value* field from ServiceNow, and the value from the *Canceled_Change_Request_State* field matches the value from the *Label* field from ServiceNow. These values must match for the maintenance sync to work.

13. Click **[Save]**. The **Configuration** pane automatically closes after this message appears.

14. Click **[Run]** ( ) to run the application.

15. While the "Sync Maintenance Schedules from ServiceNow to SL1" application runs, you can monitor the status of the maintenance process by clicking the branch icon ( ) on the "Schedule Maintenance" step. Click the triggered application's run ID in the pop-up window, and then click the branch icon on the "Create SL Maintenance" or "Modify Maintenance" steps for more information.

16. After the "Sync Maintenance from ServiceNow to SL1" application completes, navigate to the **Schedule Manager** (Registry > Schedules > Schedule Manager) in SL1 to view the change requests.

17. Verify that the **Schedule Summary** field contains the same value from the ServiceNow **Number** field. In this example, the value in SL1 matches the value from ServiceNow: *CHG0030004*.

18. You can also verify that the schedule was created for a device by navigating to the **Device Manager** (Registry > Devices), clicking the wrench icon for the device, and clicking the **[Schedule]** tab.

19. If you want to cancel the scheduled time for the maintenance sync, open the change request in ServiceNow, click the **Additional actions** menu button ( ), and select *Cancel Change*. The next time the "Sync Maintenance from ServiceNow to SL1" application runs, the application cancels that maintenance sync.

> **TIP**: As a best practice, schedule the "Sync Maintenance from ServiceNow to SL1" application to run every hour or so, depending on your environment.

Using a MID Server to Place a Device Directly in Maintenance or Schedule a Maintenance Window

You can use this SyncPack with a ServiceNow MID Server for the following scenarios:

- Placing a device directly in maintenance.

- Scheduling a maintenance window

To set up either scenario, perform the following:

1. *Configure and run the "Sync Catalog Requirements" application* in PowerFlow.

2. *Create a ServiceNow Credential and Credential Affinity in ServiceNow,* or

   Configure ServiceNow to use an API Key for PowerFlow Authentication

Configuring and Running the "Sync Catalog Requirements" Application in PowerFlow

Prerequisites for the MID Server workflow:

1. In PowerFlow, open the "Sync Catalog Requirements" application and click the **[Configure]** button (⚙).

   The **Configuration** pane appears.

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

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

   > NOTE: The *region* field is populated by the configuration object you aligned with this application. The region value must match the value in the *SL1 Region* field in ServiceNow. If you need to update this value, you will need to define the *region* variable in the configuration object that is aligned with this application, or align a different configuration object that has the correct *region* value.

   - *snow_request_limit*. Specify the number of CIs fetched from ServiceNow in each request. The default is 500 objects per chunk.

   - *read_timeout*. Specify the maximum amount of time in seconds the application should wait for a response before timing out. The default is 30 seconds.

   - *sl1_chunk_size*. Specify the number of object to include in each chunk sent from SL1 to ServiceNow. The default is 500.

   - *snow_chunk_size*. Specify the number of objects to send in each chunk from ServiceNow. The default is 150 objects per chunk.

3. Click **[Save]**.

4. Click the **[Run]** button. The application syncs the configuration objects from PowerFlow to ServiceNow so that ServiceNow can tell PowerFlow to which SL1 system it needs to send the change request.

5. In ServiceNow, create a credential for the sync. For more information, see *Creating a ServiceNow Credential and Credential Affinity* or *Configure ServiceNow to use an API Key for PowerFlow Authentication*.

> **NOTE:** You do *not* need to run or schedule the "Create or Update Maintenance Schedule from ServiceNow Trigger" application or the "Trigger Device Maintenance Updates via MID Server" application in PowerFlow. Those applications are triggered by the MID Server, which is triggered first by a registered event in ServiceNow Event Management. For more information about registered events, including examples of other triggering events you can define in ServiceNow, see *ServiceNow Registered Events*.

Creating a ServiceNow Credential and Credential Affinity

Before you can set up maintenance sync with the MID Server in ServiceNow, you need to create a credential for the MID Server. You should have access to the "Integration Services" section of the **Discovery Dependents** page (***x_sclo_scilogic_discovery_dependent***) in ServiceNow, which is provided by the "Integration Services" import set.

To create a credential to connect to PowerFlow:

1. In ServiceNow, go to **Connections & Credentials > Connection & Credential Aliases**.

2. From the **Connection & Credential Aliases** list, select **ScienceLogic**. The **Connection & Credential Aliases page** appears:
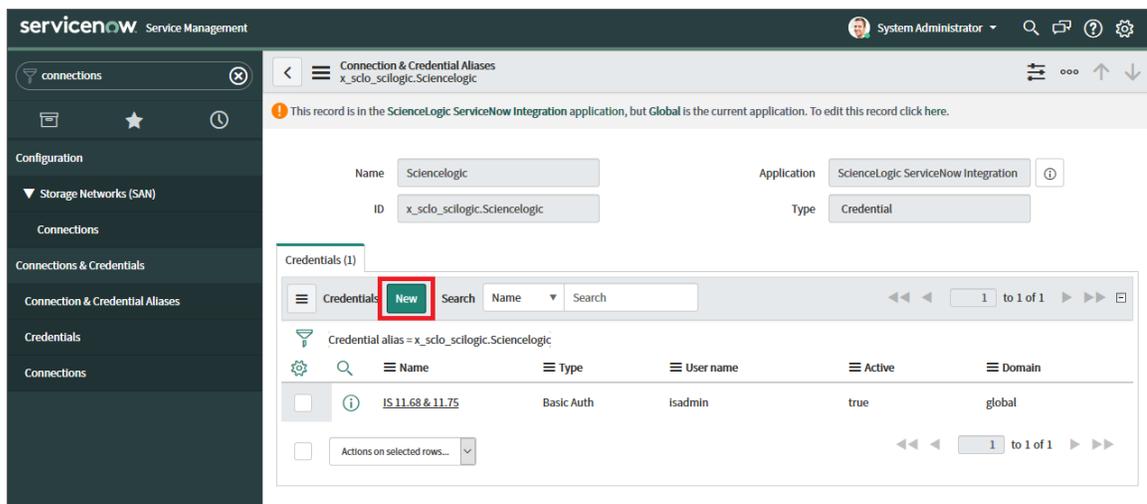
3. Click **[New]** to create a new credential. The **Credentials** page appears.

4. From the list of credentials, select **Basic Auth Credentials**. This is currently the only type of credential that is supported. The **Basic Auth Credentials** page appears.

5. Complete the fields related to the PowerFlow on the **Basic Auth Credentials** page. Make sure that the *Credential alias* field is set to *x_sclo_scilogic.Sciencelogic*.

5. Click **[Submit]**. The credential is added to the **Connection & Credential Aliases** page.

6. Select the new credential. The **Basic Auth Credentials** page for that credential appears.

7. Click **[New]** to create a Discovery IP Affinity record. A new **Credential Affinity** page appears.

8. Complete the following fields:

- *MID server*. The name of the MID Server you want to use.
- *IP address*. Use the PowerFlow IP address that was listed in the *IP* field on the relevant record on the **Discovery Dependents** page in ServiceNow. To quickly find the relevant record on the **Discovery Dependents** page, right-click the **Type** column and select *Group By Type*, and then expand *Type: Integration Services*. Use the IP value from the record that matches the Region for the devices you want to use.
- *Credential ID*. This field should be completed for you.

9. Click **[Submit]**.

Configuring ServiceNow to use an API Key for PowerFlow Authentication

If you want to use an API key for authentication instead of using the default "isadmin" user, you will need to add an HTTP header to the **Integration Services Application Run** record in ServiceNow.

1. In PowerFlow, create an API key on the **API Keys** page and save the key for step 9, below.

2. In ServiceNow, go to **System Web Services > Outbound > REST Message** and search for *Integration Services Application Run*.

3. Select the **Integration Services Application Run** record that is connected to the *ScienceLogic SL1: Service Catalog Request Integration* in the **Application** column. The **Integration Services Application Run** record appears:

4. In the **HTTP Methods** section, click **IS Application Run**.

5. Click the **[HTTP Request]** tab.

6. If needed, click *Insert* from the **Additional actions** menu at the top left of the record.

7. In the *Use MID Server* field, select the MID server you want to use.

8. In the **HTTP Headers** section, type *PF-APIKEY* in the *Name* field.

9. In the **HTTP Headers** section, paste the API key from step 1 in the *Value* field.

10. Click the **[Update]** button in the lower left of the window.

ServiceNow Registered Events

This section describes the commands that you can use to generate registered events in ServiceNow. These events can trigger actions in PowerFlow, such as specifying one or more CIs for monitoring, or putting a CI into maintenance.

> **WARNING:** This section is recommended for advanced ServiceNow administrators.

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

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

For more information about registered events, see the ServiceNow documentation: https://developer.servicenow.com/dev.do#!/learn/learning-plans/quebec/new_to_servicenow/app_store_learnv2_automatingapps_quebec_what_are_events.

Change Request Events

The following events can be triggered to queue to the event manager. The actual trigger for the following events is not included in the application, but you can configure a custom trigger. An example is provided in the *Change Request Events Example*, below.

Direct to Device: x_sclo_scilogic.device_maintenance

The "Trigger Device Maintenance Updates via MID Server" application receives a group of one or more synced devices from the ServiceNow MID Server and checks for the *enable_maintenance* and *disable_maintenance* actions on those devices. If the application encounters devices with one of those actions, it will enable or disable the user maintenance status of those devices as needed.

Command

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

Event Fields

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

Scheduled Device Maintenance: x_sclo_scilogic.device_maintenance_skd

The "Create or Update Maintenance Schedule from ServiceNow Trigger" application handles scheduling and canceling maintenance schedules in SL1 from a trigger from a ServiceNow Management, Instrumentation, and Discovery (MID) Server.

The "Create or Update Maintenance Schedule from ServiceNow Trigger" receives a group of one or more synced devices from the ServiceNow Management, Instrumentation, and Discovery (MID) Server and checks for the schedule and cancel actions on those devices. If the application encounters devices with one of those actions, it will enable or disable the user maintenance status of those devices as needed.

As a result, in ServiceNow you can bring change requests out of a scheduled state and update them, and all of the updates to those change requests are synced back to SL1 in real time, even if those change requests were already scheduled.

Command

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

Event Fields

Field	Description
`x_sclo_scilogic.device_maintenance_skd`	Unique name of the event.
`current`	The table to which the event applies.
`schedule`	Parm1: Accepts the following variables: • `schedule`: Creates a schedule in SL1. • `cancel`: Deletes any currently scheduled maintenance. • `event`: Creates an event in SL1.
`current.getUniqueValue()`	Parm2: The `sys_id` of the change request.

Change Request Events Example

There are many different ways to utilize the registered events that are supplied in the certified application. Below is on possible way of triggering by using business rule on the change request table.

1. Navigate to the change request table and create a business rule:
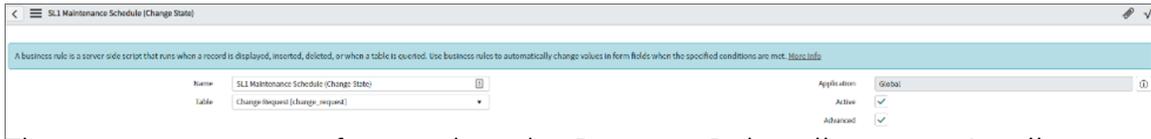
2. Configure the Business Rule to trigger when the state of the change request changes to *Scheduled* or *Canceled*:
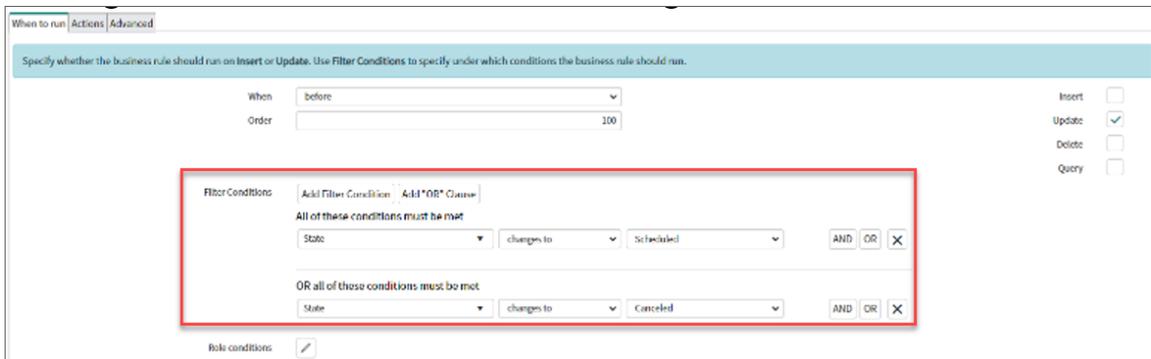
3. Click the **[Advanced]** tab, where you can configure the trigger for the event queue:
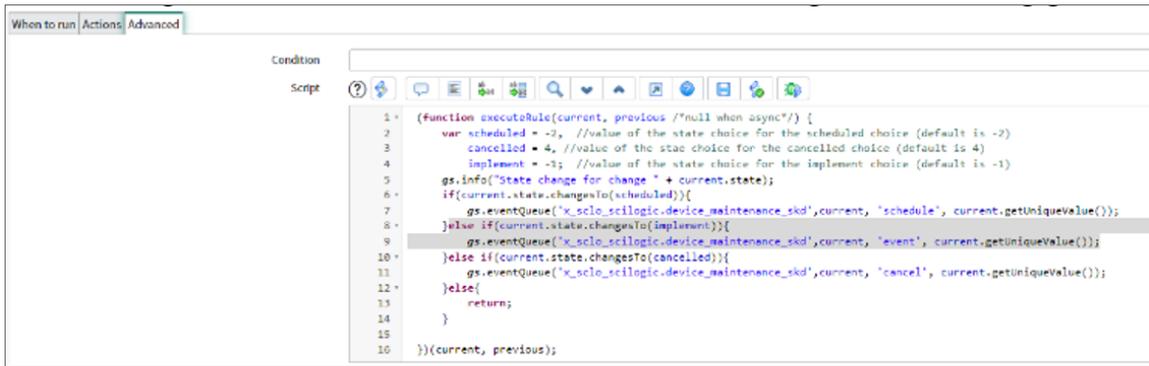
This script block is set up to trigger three different times, depending on the state change:

```
(function executeRule(current, previous /*null when async*/) {
        var scheduled = -2,  //value of the state choice for the scheduled
choice (default is -2)
                cancelled = 4, //value of the stae choice for the cancelled choice
(default is 4)
                implement = -1;  //value of the state choice for the implement
choice (default is -1)
        gs.info("State change for change " + current.state);
        if(current.state.changesTo(scheduled)){
                gs.eventQueue('x_sclo_scilogic.device_maintenance_skd',current,
'schedule', current.getUniqueValue());
        }else if(current.state.changesTo(implement)){
                gs.eventQueue('x_sclo_scilogic.device_maintenance_skd',current,
'event', current.getUniqueValue());
        }else if(current.state.changesTo(cancelled)){
                gs.eventQueue('x_sclo_scilogic.device_maintenance_skd',current,
'cancel', current.getUniqueValue());
        }else{
                return;
        }

})(current, previous);
```

Chapter

4

Troubleshooting the ServiceNow Service Catalog SyncPack

Overview

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

This chapter covers the following topics:

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:

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:

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}}'
```

Resources for Troubleshooting

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. For more information, see https://sciencelogic.com/company/legal.

ScienceLogic

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

International: +1-703-354-1010