ScienceLogic

Microsoft Teams SyncPack

Version 2.0.0

Table of Contents

Chapter 1

Introduction to the Microsoft Teams SyncPack

Overview

This chapter describes how you can use the "Microsoft Teams" SyncPack to automatically create channels and trigger event messages from SL1 to Microsoft Teams. The integration is bi-directional between SL1 and Microsoft Teams. Teams users can interact with SL1 events directly from the Microsoft Teams interface. For example, you can acknowledge, and add a note from Microsoft Teams which will be reflected in the SL1 platform in real time. If you acknowledge an event that appears in SL1 from SL1, and not from Teams, then the sync would be uni-directional and the event would not display in Teams.

This SyncPack uses the "Microsoft Teams Automation" PowerPack.

This chapter covers the following topics:

What Can I Do with this SyncPack?

The Microsoft Teams SyncPack is designed to streamline and automate communication between SL1 and Microsoft Teams, allowing teams to respond to SL1 events in real-time. This SyncPack enables the automated creation of Teams channels and messages based on specific SL1 events, fostering quick and effective collaboration around critical incidents and events.

The SyncPack provides a bidirectional integration between SL1 and Microsoft Teams, ensuring that updates can flow both ways. This means SL1 can trigger events in Microsoft Teams, while users can interact with SL1 events directly from Teams to provide a seamless experience for monitoring and responding to incidents.

You can configure this SyncPack to perform the following integrations:

- Create a channel in Microsoft Teams to post event notifications from SL1.
- Synchronize an event update made in Microsoft Teams with the event status in SL1.
- Synchronize other event activities and status changes between teams and SL1.
- Add notes in Microsoft Teams that will display in SL1 from the event acknowledgement.

For more information about how to configure these applications, see *Configuring Applications for the Microsoft Teams SyncPack*.

Contents of the Microsoft Teams SyncPack

This section lists the contents of the "Microsoft Teams" SyncPack.

PowerFlow Applications

- ***Create Channel For Event***. This application sends event notifications to Microsoft Teams by sending an event to a new channel with the event name, or to an already existing channel. The adaptive card in Microsoft Teams includes *action buttons for handling the event*, as well as the following fields:
    - ***Event URL***. Hyperlink to the event in SL1.
    - ***Event ID***. Unique identifier for the event.
    - ***Device ID***. Identifier of the device associated with the event.
    - ***Device Name***. Name of the device associated with the event.
    - ***Event Severity***. Severity level of the event.
    - ***Event Severity Label***. Label describing the severity level.
    - ***Event Message***. Detailed message describing the event.
    - ***Last Detected***. Timestamp of when the event was last detected.
    - ***First Detected***. Timestamp of when the event was first detected.
- ***Sync Event Status From Teams to SL1***. This application synchronizes event updates between Microsoft Teams and SL1. The application retrieves the event status from the Teams Application and delivers the same status to SL1.

For more information about how to configure these applications, see *Configuring Applications for the Microsoft Teams SyncPack*.

Configuration Object

- ***Sample Microsoft Teams Config***. This configuration object can be used as a template after the SyncPack is installed on the PowerFlow system. The configuration object includes the following:
    - Details for connecting to the SL1 API, including the URL, username, and password
    - Details for connecting to the Microsoft Teams API, including the base URL, private instance URL, and API keys
    - Details for connecting to an HTTP Proxy when communicating with Microsoft Teams, including the hostname or IP address, username, and password
    - Mapping between SL1 events and Microsoft Teams alerts including, device ID format, heartbeat and schedule name, default user, and maintenance object format information

Steps

The following steps are included in this SyncPack:

- Create Teams Channel
- Create SL1 Payload
- Resolve Event From SL1
- Fetch Channel ID From Name
- Fetch Microsoft Email

- Fetch Teams User Input Data
- Post Event Card To Teams
- Post Event Details To Channel
- Post Update to SL1

Contents of the Microsoft Teams Automation PowerPack

This section lists the contents of the "Microsoft Teams Automation" PowerPack, including run book automation policies as well as actions you can use to automate adding, updating, and clearing events, and to automate creating Microsoft Teams channels in SL1:

Run Book Automation Policies

- Microsoft Teams: Create Channel
- Microsoft Teams: Created Channel Notification - Acknowledge
- Microsoft Teams: Created Channel Notification - Clear

Run Book Actions

- Microsoft Teams: Create Channel
- Microsoft Teams: Created Channel Notification - Acknowledge
- Microsoft Teams: Created Channel Notification - Clear

Credentials

- "PowerFlow Microsoft Teams" sample credential

Chapter 2

Installing the Microsoft Teams SyncPack

Overview

This chapter describes how to install the "Microsoft Teams" SyncPack, and how to install the "Microsoft Teams Automation" PowerPack.

This chapter covers the following topics:

Prerequisites for the SyncPack

This SyncPack requires the following:

- The latest "Microsoft Teams Automation" PowerPack
- "SL1 Notifications Base" SyncPack version 1.0.3
- "Base Steps" SyncPack version 1.5.5
- Administrator access to the SL1 Administration Portal
- Administrator access to Microsoft Teams
- A secured SSL/TLS-enabled connection to PowerFlow

For the latest System Requirements, see the *SL1 PowerFlow SyncPack Release Notes*.

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	Microsoft Teams API	Any	TCP 443	Microsoft Teams API Access

Downloading and Installing the Microsoft Teams SyncPack

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

Downloading the SyncPack

> **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

To import a SyncPack in the PowerFlow user interface:

1. On the **SyncPacks** page ( ) of the PowerFlow user interface, click **[Import SyncPack]**. The **Import SyncPack** page appears.
2. Click **[Browse]** and select the **.whl** file for the SyncPack you want to install. You can also drag and drop a **.whl** file to the **Import SyncPack** page.
3. Click **[Import]**. PowerFlow registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in PowerFlow. For more information, see the following topic.

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

Installing the SyncPack

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 SyncPack 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 Microsoft Teams Automation PowerPack

The "Microsoft Teams Automation" PowerPack includes run book automation policies, run book actions, and a sample credential that that can be used to integrate SL1 and Microsoft Teams.

NOTE: The "Microsoft Teams Automation" PowerPack requires PowerFlow platform version 2.5.0 or later and SL1 version 11.2.0 or later. For details on upgrading SL1, see the *SL1 Platform Release Notes*.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of 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 section on *Global Settings*.

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site.
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter 3

Configuring Applications for the Microsoft Teams SyncPack

Overview

This chapter describes how to set up the run book automations in SL1 and how to configure the PowerFlow applications in the "Microsoft Teams" SyncPack

This chapter covers the following topics:

Workflow for Configuring the SyncPack

The following workflows describe how to configure Microsoft Teams, SL1, and PowerFlow to work with the "Microsoft Teams" SyncPack.

Configuring Microsoft Teams

1. See the steps in *Configuring Microsoft Teams*, below.

Configuring SL1

1. *Create a SOAP/XML credential to access PowerFlow*
2. *Edit the Microsoft Teams run book actions*
3. *Enable the Microsoft Teams run book automations*

Configuring PowerFlow

1. *Create and align a PowerFlow configuration object*
2. *Configure the Microsoft Teams PowerFlow applications*
3. *Schedule the PowerFlow applications*

Configuring Microsoft Teams

To configure Microsoft Teams to work with PowerFlow, you must register an application in Azure Active Directory and perform some configuration steps for that application before using the "Microsoft Teams" SyncPack. To configure the application:

1. Follow the steps in the Microsoft documentation to register your application in Azure Active Directory.
2. Note the "app_id(client_id)", "client_secret", and "tenant_id" of your application.

> IMPORTANT: The "client_secret" value of your application will only be visible immediately after creating it, so be sure to copy it somewhere safe before continuing.

3. Grant the following scoped permissions to the application:
    - Channel.Create
    - ChannelMessage.Send
    - Channel.ReadBasic.All
    - offline_access
    - openid

- User.Read
- User.Read.All

4. Ensure the following Microsoft Graph permissions are assigned to allow proper functionality of the SyncPack:

API/ Permissions Name	Type	Description	Administrator Consent Needed
Channel.Create	Application	Create channels	Yes
ChannelMessage.Read.All	Delegated	Read user channel messages	Yes
ChannelMessage.Read.All	Application	Read all channel messages	Yes
ChannelMessage.ReadWrite	Delegated	Read and write user channel messages	Yes
ChannelMessage.Send	Delegated	Send channel messages	No
email	Delegated	View users' email address	No
openid	Delegated	Sign users in	No
profile	Delegated	View users' basic profile	No
User.Read	Delegated	Sign in and read user profile	No
User.Read.All	Application	Read all users' full profiles	Yes

5. Ensure the following Power Automate permissions are assigned to allow proper functionality of the SyncPack:

API/ Permissions Name	Type	Description	Administrator Consent Needed
Activity.Read.All	Delegated	Allow the application to read activities	No
Flows.Read.All	Delegated	Allow the application to read flows	No
User	Delegated	Access Microsoft Flow as signed in user	Yes

Required Licenses and Configuration

The following licenses and access permissions are required in order to configure the workflows necessary for the sync between Microsoft Teams and SL1:

- ***F3 License (or higher)***. This is required to enable access to Microsoft Teams, and to create and interact with workflows.
- ***Power Automate Premium License***. This is required for building and automating workflows, such as posting adaptive cards based on webhook events and processing responses.
- ***Administrator access to the SL1 Administration Portal and Microsoft Teams***. This is required to configure and manage workflows across both platforms.
- ***SL1 Integration/Automation License (if required)***. Depending on how SL1 is being integrated, specific licensing may be needed for automation or API integration with third-party services like Microsoft Teams.

Additional Configuration Required:

- Ensure that all required IP addresses are added to the firewall rules and that the rules are enabled to allow communication between the necessary services.
- The workflow app must be added to Teams or Power Automate, which requires administrator approval. Your admin team should assist with this approval.

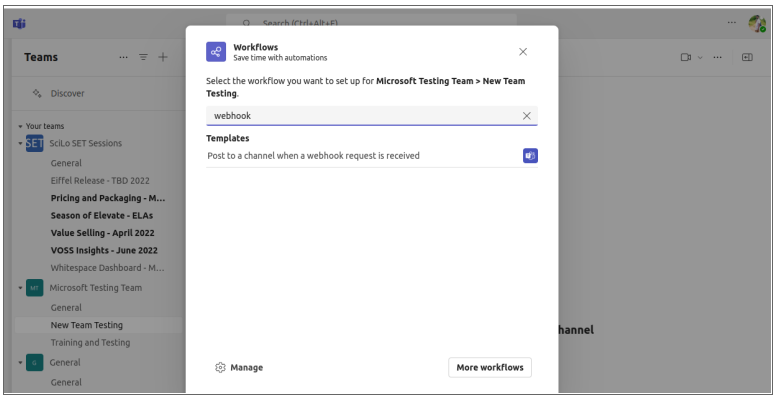
Configuring Microsoft Teams Workflows

To configure Microsoft Teams workflows:

1. In Microsoft Teams, click the three dots icon on the left navigation bar to access additional apps.
2. Click **[Workflows]** to view the available options for creating and managing workflows.

Creating a Webhook Workflow

1. In the Workflows window, search for "Webhook".
2. Select the ***Post to a channel when a webhook request is received*** template.



3. Once selected, an HTTP POST request URL will be generated.

NOTE: Copy this URL, since it will be used later to send messages to the appropriate Teams channel.

4. Click **[Done]**.

Necessary Workflows

Two key workflows have been added to enable the integration between Microsoft Teams and PowerFlow, "Posting Adaptive Cards to a Teams Channel" and "When someone responds to an adaptive card"

Configuring the "Posting Adaptive Cards to a Teams Channel" workflow

The "Posting Adaptive Cards to a Teams Channel" automatically posts an adaptive card to a specific Teams channel when a webhook request is received, providing real-time updates or messages as events occur.

1. In the Workflows window, search for "Posting Adaptive Cards to a Teams Channel".
2. Select the workflow and follow the prompts given. When done, you will see a confirmation message saying "Workflow added successfully!"
3. Click the three dots icon next to the workflow and select ***Edit*** to view the flowchart of the workflow's execution and make necessary modifications to the steps.
4. Select the appropriate authentication option from the ***Who can trigger the flow?*** field:
    - *Anyone*. Publicly accessible; anyone can trigger the flow. Authentication is not required since the URL includes an SAS Token.
    - *Any user in my tenant*. Restricts the flow trigger to users within your organization. Any user within the same tenant can call this webhook.
    - *Specific users in my tenant*. Allows only designated users within your tenant to trigger the flow.
5. Under the **Send each adaptive card** section, enter the following values in the listed fields to ensure that adaptive cards are posted in specific Teams channels with the appropriate actions:
    - ***Team***. `items('Send_each_adaptive_card')?['tid']`
    - ***Channel***. `items('Send_each_adaptive_card')?['chid']`
    - ***Card Type ID***. Ensure that the value specified here is the same value used in the "When someone responds to an adaptive card" workflow.
6. Click the **[Save]** button in the top right corner.

Configuring the "When someone responds to an adaptive card" workflow

This workflow is triggered when a user takes action on the adaptive card (e.g., Acknowledge, Resolve). The workflow initiates a premium HTTP request to POST response data to PowerFlow, which triggers further automated processes in SL1.

1. In the Workflows window, search for "When someone responds to an adaptive card".
2. Select the workflow and follow the prompts given. When done, you will see a confirmation message saying "Workflow added successfully!"
3. Click the three dots icon next to the workflow and select ***Edit*** to view the flowchart of the workflow's execution and make necessary modifications to the steps.
4. Under the **When someone responds to an adaptive card** action, ensure that the value in the ***Card Type ID*** field is the same value used in the "Posting Adaptive Cards to a Teams Channel" workflow.
5. Click the plus sign icon.
6. Click the ***Actions*** Tab.

7. Select *HTTP* from the ***Actions*** tab.
8. Under the **HTTP** action, supply the following values in the listed fields to ensure that data is properly sent to PowerFlow when a button on the adaptive card is clicked in Teams:
    - ***Method***. Select *POST*.
    - ***URI***. Enter the PowerFlow intermediary URL. For more information on generating this URI, see the section on *Generating the PowerFlow Intermediary URL*.
    - ***Body***. Enter `addProperty(triggerBody()?['entity']?['cardOutputs'], 'user', triggerBody()?['entity']?['teamsFlowRunContext']?['messagePayload']?['from']?['User']?['Id'])`.

NOTE: You must be a PowerAutomate Premium user to configure the HTTP action on the workflow.

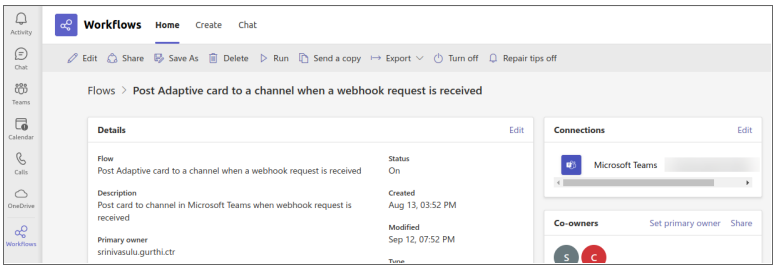
9. Click the **[Save]** button in the top right corner.

NOTE: You can design multiple workflows to meet your organization's requirements, allowing seamless integration of multiple PowerFlow and SL1 base environments to deliver notifications to either the same or different channels.

Managing and Editing Workflows

Created or added workflows can me edited, deleted, copied, or run manually by any user who is an owner of the workflow. To manage workflows:

1. In Microsoft Teams, click the three dots icon on the left navigation bar to access additional apps.
2. Click **[Workflows]** and then select ***Manage workflows***.
3. The following options for managing workflows are available here:
4. Locate the workflow you want to manage and click on it.
5. At the top of the workflow you will see options to ***Edit***, ***Share***, ***Delete***, ***Run***, and more.



Adding Co-Ownwers to a Workflow

You can grant access to specific workflows by adding users as co-owners. To add a co-owner to a workflow:

1. In Microsoft Teams, click the three dots icon on the left navigation bar to access additional apps.
2. Click **[Workflows]** and then select ***Manage workflows***.
3. Locate the workflow you want to add a co-owner to, click the three dot icon next to it, and select ***Share***.
4. In the **Co-owners** section, type the email address of the user or group you want to add as a co-owner and select the appropriate result from the drop-down menu.
5. The added user will now have full control over the workflow, including the ability to edit, delete, or manage the run history of the workflow.

Retrieving the Team ID and Channel ID from a Team

When *setting up the configuration object in PowerFlow* to enable syncing between Microsoft Teams and SL1, you will need to supply the team_id and channel_id so that the event notification is posted to the right place in Teams. You must retrieve these values from Teams before setting up the configuration object in PowerFlow. To retrieve these values:

1. In Microsoft Teams, select ***Teams*** from the navigation menu on the left.
2. Click on the three dots icon to the right of the team name you want to retrieve the "team_id" value from.
3. Click ***Get link to team***.
4. Copy the URL in the window that appears and paste into a text editor. For example, `https://teams.microsoft.com/l/team/19%3Ad2a0d920c47f4b4e9b82a61218deed0d%40thread.tacv2/conversations?groupId=b21c662a-e8a2-489f-a78d-a2771d478e0f&tenantId=4df927fc-2cb7-4222-a6a5-c12dab55b4a6`
5. To separate out the ***team_id*** value, copy the part of the url from `groupId` to `&`. Using the example above, the "team_id" would be `b21c662a-e8a2-489f-a78d-a2771d478e0f`.
6. In Microsoft Teams, find the channel you want to retrieve the "channel_id" value from.
7. Click on the three dots icon to the right of the channel name.
8. Click ***Get link to channel***.
9. Copy the url in the window that appears and paste into a text editor. For example, `https://teams.microsoft.com/l/channel/19%3Ad2a0d920c47f4b4e9b82a61218deed0d%40thread.tacv2/General?groupId=b21c662a-e8a2-489f-a78d-a2771d478e0f&tenantId=4df927fc-2cb7-4222-a6a5-c12dab55b4a6`
10. To separate out the ***channel_id*** value, copy the part of the url from `channel/` to `/`. Using the example above, the "channel_id" would be `19%3Ad2a0d920c47f4b4e9b82a61218deed0d%40thread.tacv2`.
11. Set aside the values for the "team_id" and "channel_id" to be used when setting up the configuration object in PowerFlow.

Retrieving the Object ID from your Azure Active Directory Application

The Object ID from your Azure Active Directory application is needed when setting up the configuration object in PowerFlow. It is used to populate the "microsoft_user_ids" value. To retrieve the Object ID:

1. Open Azure Active Directory.
2. Select ***Enterprise applications*** from the navigation menu on the left.
3. Search for the name of your application.
4. Take note of the ***Object ID*** value. This value should be entered in the "microsoft_user_ids" field in PowerFlow.

Generating the PowerFlow Intermediary URL

When configuring the workflow to handle acknowledgment actions in Teams, you will need to generate an Intermediary URL from PowerFlow:

1. Generate an API key in Powerflow. For more information, see the Creating and Using API Keys in SL1 PowerFlow section in the PowerFlow Platform manual.
2. Insert your PowerFlow IP address and the generated API key into the following URL:

```
<https://<Your> PF IP Address>/api/v1/applications/sync_event_status_from_
teams_to_sl1/run?PF-APIKEY=<API Key>
```

IMPORTANT: This URL is crucial for enabling bi-directional sync between Microsoft Teams and SL1, particularly for acknowledgments triggered from adaptive card actions.

Configuring SL1

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

Creating a SOAP/XML Credential to Access PowerFlow

You will need to create a SOAP/XML credential so that the action policies included in the "Microsoft Teams Automation" PowerPack can access your PowerFlow system.

The PowerPack includes the "PowerFlow Microsoft Teams" credential, which you can use as a template for the SOAP/XML credential.

To define a SOAP/XML credential using the example credential:

1. In SL1, go to the **Credentials** page (Manage > Credentials).
2. Click the wrench icon () for the "PowerFlow Microsoft Teams" credential. The **Credential Editor** modal window appears.
3. Complete the following fields:
    - ***Profile Name***. Type a new name for the credential.
    - ***URL***. Type the URL for your PowerFlow system.
    - ***HTTP Auth User***. Type the username for your PowerFlow system.
    - ***HTTP Auth Password***. Type the password for your PowerFlow system.

4. Click **[Save & Close]**.
5. Take note of the SL1-assigned ID number for the new credential on the **Credentials** page, in the **ID** column. You will need the ID number when editing the input parameters of the run book actions included in the PowerPack, below.

Editing the Microsoft Teams Run Book Actions

The "Microsoft Teams Automation" PowerPack includes three run book actions that use the "Run Integration Service Application" action type to trigger the PowerFlow applications that send and receive data to and from Microsoft Teams. You can specify the credential ID in a JSON structure that you enter in the ***Input Parameters*** field in the **Action Policy Editor** modal.

After you edit the action and trigger the event policy, the new event log will be added to the respective device on the **Event Console** page.

To edit the actions included in the PowerPack:

1. In SL1, go to the **Actions** page (Registry > Run Book > Actions).
2. Locate the automation action that you want to use, and then click its wrench icon (). The **Editing Action** page appears.
3. In the ***Input Parameters*** field, change the values of the following parameters:
    - ***credential_id***. Change the value to the credential ID that you noted earlier when creating a credential for your PowerFlow system. This parameter is required.
    - ***include_event***. Leave the value as "true".
    - ***application_name***. Leave the default application value.
    - ***params***. Leave the default parameter value.
4. Make sure the ***Action State*** is set to *Enabled*, and then click **[Save]**.

Enabling the Microsoft Teams Run Book Automations

The "Microsoft Teams Automation" PowerPack includes three run book automation policies that you will need to enable.

These policies update the Microsoft Teams alert or SL1 event with the state of the associated event or alert, create a maintenance object, or populate SL1 with on-call details. When the event is first detected in SL1 or Microsoft Teams, the "triggered" state is sent. When the event is acknowledged in SL1 or Microsoft Teams, the "acknowledged" state is sent. When the event is cleared in SL1 or Microsoft Teams, the "resolved" state is sent. When the on-call policy is triggered, on-call details are added to the SL1 event notes. When a maintenance object is required, a maintenance object is created in Microsoft Teams.

The following table shows the automation policy, its aligned events, and the automation action that runs in response to the events.

Automation Policy Name	Aligned Events	Automation Action
Microsoft Teams: Create Channel	(All Events)	Microsoft Teams: Create Channel
Microsoft Teams: Create Channel Notification -	(All Events)	Microsoft Teams: Created Channel Notification - Acknowledge

Automation Policy Name	Aligned Events	Automation Action
Acknowledge		
Microsoft Teams: Create Channel Notification - Clear	(All Events)	Microsoft Teams: Create Channel Notification - Clear

To enable the run book automation policies:

1. In SL1, go to the **Automation** page (Registry > Run Book > Automation).
2. Locate a Microsoft Teams automation policy and click its wrench icon ( ). The **Automation Policy Editor** page appears.
3. Update the following fields:
   - ***Policy State***. Select *Enabled*.
   - ***Policy Priority***. Select *High* to ensure that this PowerFlow automation policy is added to the top of the queue.
   - ***Available Actions***. If it is not already selected, select the "Run Integration Service Application: *<name>*" action that corresponds with the Ansible automation policy you selected in step 2, and click the arrows to move it to ***Aligned Actions***.

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 remaining Microsoft Teams run book automation policies.

Configuring PowerFlow

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

Creating a Configuration Object

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

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

For this SyncPack, you can make a copy of the "Sample Microsoft Teams Config" configuration object, which is the sample configuration file that was installed with the *Microsoft Teams* SyncPack.

TIP: The "Sample Microsoft Teams Config" configuration object contains all of the required variables. Simply update the variables from that object to match your SL1 and Microsoft Teams settings.

To create a configuration object based on the "Sample Microsoft Teams Config" configuration object:

1. In the PowerFlow user interface, go to the **Configurations** page ( ).
2. Click the **[Edit]** button for the "Sample Microsoft Teams Config" configuration object. The **Configuration** pane appears.
3. Click **[Copy as]**. The **Create Configuration** pane appears.
4. Complete the following fields:
    - ***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 Values*** fields, update the default variable definitions to match your PowerFlow configuration:
    - ***sl1_host***. The hostname or IP address of the SL1 system the alerts will synchronize to.
    - ***sl1_user***. The username for your SL1 system.
    - ***sl1_password***. The password for your SL1 system.
    - ***client_id***. The ID of the registered app in Azure Active Directory.
    - ***client_secret***. The secret code given for the corresponding app in Azure Active Directory.
    - ***tenant_id***. Tenant ID of the Azure Active Directory.
    - ***team_id***. ID of your team in Microsoft Teams.
    - ***channel_id***. Provide the channel ID if you want to enable sending messages to existing channels. Otherwise, this field is optional.

NOTE: To retrieve the "team_id" and "channel_id" values, see the *Retrieving the Team ID and Channel ID from a Team* section.

- ***ms_teams_webhook_url***. The URL of the webhook you created in Azure Active Directory.
- ***microsoft_user_ids***. Microsoft Teams user ID used for group calling. For a group call, enter multiple user IDs in your PowerFlow configuration separated by commas. For example:

```
{

  "encrypted": false,
```

```
name":"microsoft_user_ids",

"value": "dce672e6-c430-43f9-b17e-bfe07e51816d,30127c40-
12b3-436b-8523-115d3b814ea9"

}
```

> **NOTE:** To retrieve the "microsoft_user_ids" value see the *Retrieving the Object ID from your Azure Active Directory Application* section.

- ***default_ack_user***. The email address of the default user to acknowledge events and alerts.
- ***intermediary_url***. URL endpoint of the intermediary between Microsoft Teams and PowerFlow for sending user interaction data.
- ***region***. This is used for saving the SL1 user data to cache.

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

Updating the Configuration Object to Send Messages to an Existing Team Channel

You can configure the Microsoft Teams SyncPack to send messages to any existing team channel in Microsoft Teams:

1. In the configuration object, enter the channel ID of the existing Microsoft Teams channel where you want to send SL1 event messages in the ***channel_id*** field.
2. Toggle on (blue) the enable_existing_channel configuration option. This setting allows the SyncPack to send messages directly to the pre-defined channel instead of creating a new one.
3. Click **[Save]**. You can now align this configuration object with one or more applications.

Aligning a Configuration Object and Configuring PowerFlow Applications

With this SyncPack, any status changes made to an SL1 event are sent to Microsoft Teams to update the corresponding alert. Any status changes to the Microsoft Teams alert are synced back to the corresponding SL1 event. You will need to align the Microsoft Teams applications with the relevant configuration object in PowerFlow, and, if needed, update any other fields on the **Configuration** pane for the applications.

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

- Create Channel For Event
- Sync Event Status From Teams to SL1

To align the configuration object with the relevant PowerFlow applications:

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

2. From the ***Configurations*** drop-down, select the configuration object you want to use.

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

3. Click **[Save]** to align that configuration with the application.
4. Repeat this process for the other PowerFlow applications.

Using Microsoft Teams to Receive and Respond to Event Messages

When events are created in SL1 and notifications are sent to the Microsoft Teams channel, an adaptive card displays in the Microsoft Teams channel with event details and action buttons to respond to the event. You can respond to an event notification in Microsoft Teams to be updated in SL1 by adding a note, acknowledging the event, or resolving it. Upon taking action on the event from the adaptive card (Acknowledge, Add Note, or Resolve), the selected action is sent to SL1 to update the event status.

> **NOTE:** Currently, syncing event updates from SL1 to Microsoft Teams is not supported. This is planned as a future enhancement to keep Microsoft Teams updated with the latest event state from SL1.

Responding to Event Messages

When an event is posted in Teams, you can view details of the event, such as the Event ID, Event Severity, and more. You can respond to the event in Teams, to be updated in SL1 by adding a note, acknowledging the event, or resolving it:

- ***Add Note to SL1 Event***: Click this button, type the note you want to add to the event and press Enter. The note will be added to the event in SL1.
- **[Acknowledge]**: Click this button to acknowledge the event in SL1.
- **[Resolve]**: Click this button to resolve the event in SL1.

Configuring Time Zone of Event Messages

By default, Microsoft Teams uses your local time zone. If you want to change the time zone to UTC or any other time zone, see Microsoft Teams' official documentation for instructions on changing the time zone.

SL1 follows a structured format when sending event details to PowerFlow. Below is an example of an event payload generated when an event is created and the MS Teams automation is triggered:

```
{ "app_name":"create_teams_channel_for_event", "app_vars":{ "event_
details":{ "%D":"2025-02-06 11:08:29", "%d":"2025-02-06 11:08:29",
"%M":"CRITICAL: test 11", "%N":"Microsoft Teams: Create Channel",
"%S":"CRITICAL" }, "team_id":"${config.team_id}", "tenant_
id":"${config.tenant_id}" } }
```

The `event_details` section contains two key timestamps, both of which are in UTC format by default:

- ***%D (First Detected Time)***. Represents the initial event occurrence.
- ***%d (Last Detected Time)***. Represents the most recent occurrence.

To modify the time zone format from the SL1 side, update the SL1 server time zone settings. For more information on changing the time zone settings, see the SL1 *System Administration manual*.

Chapter 4

Troubleshooting

Overview

This chapter contains troubleshooting resources and procedures to use with the "Microsoft Teams" SyncPack.

This chapter covers the following topics:

Initial Troubleshooting Steps

The first step taken when troubleshooting should be to ensure that there are no issues with the data platforms with which PowerFlow is interacting. There might be additional configurations or actions enabled on Microsoft Teams or SL1 that result in unexpected behavior. Perform the steps in the following sections to troubleshoot SL1 PowerFlow and Microsoft Teams. For detailed information about how to perform the steps below, see *Resources for Troubleshooting*.

SL1 PowerFlow

1. Run `docker service ls` on the PowerFlow server.
2. Note the Docker container version and verify that the Docker services are running.
3. If a certain service is failing, make a note of the service name and version.
4. Run `docker service ps <service_name>` to see the historical state of the failing 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`

Microsoft Teams

1. Make a note of the Microsoft Teams version and SyncPack version, if applicable.
2. Make a note of the Microsoft Teams application that is failing on PowerFlow.
3. 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.

Troubleshooting the Microsoft Teams SyncPack

This section contains information and troubleshooting steps for common issues with the Microsoft Teams SyncPack.

System Requirements

Check to confirm you meet the minimum system requirements:

- ***SL1 Notifications Base SyncPack***. Version 1.0.3 or later
- ***Microsoft Teams Automation PowerPack***. Version 2.0.0 or later
- ***Base Step SyncPack***. Version 1.5.5 or later
- ***PowerFlow Platform***. Version 2.5.0 or later

Authentication Failures

PowerFlow fails during the MSAL authentication process in the "Create Channel For Event" application, specifically when acquiring an access token for the client. The error code "AADSTS700016" indicates that the Azure AD application with the specified identifier was not found in the ScienceLogic Dev directory. This issue may present in the following ways:

- PowerFlow logs may show an error related to the inability to acquire an access token or failed authentication.
- If no explicit exception is raised, subsequent Teams API requests receive "HTTP 401 Unauthorized" responses.

To resolve this issue:

1. Ensure that the "client_id", "client_secret", and "authority_url" parameters are valid and correctly configured in PowerFlow.
2. Confirm that the Teams application has proper API permissions in Azure AD, especially the Application permission for "https://graph.microsoft.com/.default".

Missing or Misconfigured Parameters

Failures occur when required parameters such as "team_id", "channel_id", or "client_id" are missing or misconfigured, leading to incorrect payloads or URLs in the API call. This issue may present in the following ways:

- PowerFlow logs will contain parameter errors such as "NoneType" errors.
- The Microsoft Graph API might return HTTP 400 Bad Request errors if required parameters are missing or invalid.

To resolve this issue:

1. Double-check that all necessary parameters are correctly added in the PowerFlow user interface or workflow.
2. Implement checks to validate the presence and validity of parameters in the "CreateChannel" class.

Resource Limit Exceeded Error

Microsoft Teams limits the number of channels per team to 200 standard channels and 30 private channels. To work around this issue:

- ***Remove unused channels***. Delete any old or unused channels to free up space for new ones.
- ***Create a new team***. If more channels are needed, and you cannot delete any old ones, consider creating a new team.

NOTE: Check Microsoft's official documentation or the Teams Admin Center for exact channel limits, as they may vary based on licensing and updates.

Message Size Exceeded or Rate Limiting Throttling

Microsoft Teams limits messaging in the following ways:

- ***Message Size Limit***.All messages (including adaptive cards) have a maximum size limit of 28 KB.
- ***Rate Limits***. Avoid sending more than 30 messages per second to prevent throttling.

ScienceLogic recommends batching messages or cards when possible and to avoid spamming a channel with too many messages in a short period.

Premium License Required Error

The error message indicates that your workflow requires a premium license due to the use of "premium capabilities" in Microsoft Power Automate. This premium requirement applies to actions that connect to external APIs or require more advanced functionality within Power Automate. To resolve this issue:

- ***Request a Premium License***. Contact your organization's admin to request a Power Automate premium license.
- ***Trial Option***. Microsoft often provides a 90-day free trial for premium connectors.
- ***Explore Alternatives***. If possible, consider using alternative, non-premium actions.

Unauthorized API Key was not found Error

The "Unauthorized, APIKey was not found" error occurs because the HTTP action in your workflow is missing an API key in the request headers. To resolve this issue:

- Check that you have a valid API key for the external service being called in PowerFlow.
- Verify that the API key value is correct and hasn't expired.
- If the API key is correctly configured but still shows this error, verify the access permissions of the API key or any additional security configurations on the external service side.

Unable to Retrieve Cached Users

If you receive the "Unable to Retrieve Cached Users" error when clicking the "Acknowledge" button on an event notification, you must run the "Cache SL1 Users" application in the "Base Steps SyncPack" to validate all user account details in SL1.

Missing Required Parameters

If required parameters such as "event_id", "action", or "user" are not provided, it may lead to a "KeyError" or similar exceptions. For example:

- "Required parameter 'event_id' is missing."
- "Required parameter 'action' is missing."
- "Required parameter 'user' is missing."

Invalid Parameter Types

If parameters do not match expected types, it may cause runtime errors. For example:

- "Invalid type for parameter 'event_id'. Expected String."
- "Invalid type for parameter 'user'. Expected String."

Logging Errors

If the logger fails to log the information (e.g. if the logging system is misconfigured), it might not raise a fatal error, but it could lead to missing logs. The error message displayed will be "Logging failed for user ID: {user}. Check logging configuration."

Microsoft Teams Workflow Error: Adaptive Card Response Failure

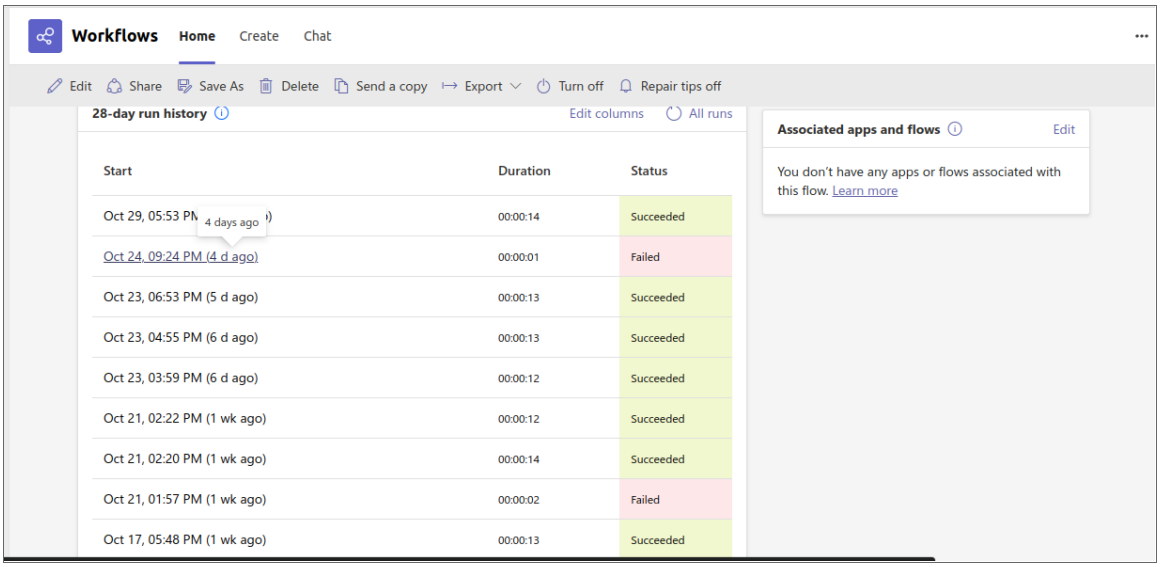
If you receive the "Invalid application: sync_event_status_from_teams_to_sl1" error message, you need to supply the "event_id" variable in App Configuration in Microsoft Teams. If the "event_id" variable is not present, the workflow will not execute. To resolve this issue:

1. Open ***App Configuration*** in Microsoft Teams.
2. Locate the ***event_id*** variable and assign it a valid value.
3. Re-run the workflow and then check the workflow history to verify the workflow executes correctly.

Checking the Run History of a Workflow

You can view the Run History of a workflow to check workflow logs, review execution steps, and confirm successful syncing between Teams and PowerFlow. To view the run history of a workflow:

1. In Microsoft Teams, click the three dots icon on the left navigation bar to access additional apps.
2. Click **[Workflows]** and then select ***Manage workflows***.
3. Select the workflow from the list that you want to view the history for.
4. Scroll down until you see ***28-day run history***.
5. Under 28-day run history you will see every time the workflow ran, with a timestamp, duration, and status.
6. To view more detail about a specific time the workflow ran, click on the timestamp.



Additional Troubleshooting Tips

ScienceLogic recommends also checking the following items when troubleshooting issues with the Microsoft Teams SyncPack:

- ***Authentication Issues (Incorrect Client ID, Secret, or URL)***. Verify that your credentials and authority URL are correct.
- ***Authorization Problems (Token Issues)***. Confirm the token permissions and renew tokens as necessary.
- ***Invalid or Missing User Information***. Regularly check for the accuracy of user IDs.
- ***Rate Limiting (Too Many Requests)***. Monitor and distribute requests to avoid hitting rate limits.
- ***Troubleshooting Visibility***. Review logs for detailed error messages.

Resources for Troubleshooting PowerFlow

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. Use the following for login: guest guest for the login.
    - Username: guest
    - Password: guest

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

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

Helpful Docker Commands

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

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

Viewing Container Versions and Status

To view the PowerFlow version, SSH in 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.

Viewing Logs for a Specific Service

You can use the Docker command line to view the logs of any currently-running service in the PowerFlow cluster. To view the logs of a 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
```

Application logs are stored on the central database as well as on all 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 relate only to the services running on that host. For this reason, using the Docker service logs is the best way to get logs from all hosts at once.

Clearing RabbitMQ Volume

RabbitMQ is a service that distributes tasks to be executed by PowerFlow workers.

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

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

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

```
docker service rm iservices_rabbitmq
```

```
docker volume rm iservices_rabbitdb
```

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

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

Re-deploy PowerFlow by running the following command:

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

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

Viewing the Process Status of All Services

Run the following command:

```
docker ps
```

Deploying Services from a Defined Docker Compose File

Run the following command:

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

Dynamically Scaling for More Workers

Run the following command:

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

Completely Removing Services from Running

Run the following command:

```
docker stack rm iservices
```

Diagnostic Tools

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

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

1. Select the PowerFlow application from the **Applications** page.
2. Hover over **[Run]** from and select *Debug Run* from the pop-up menu. PowerFlow executes the application in Debug Mode with a log level of 10.

To run an application in Debug Mode using the API:

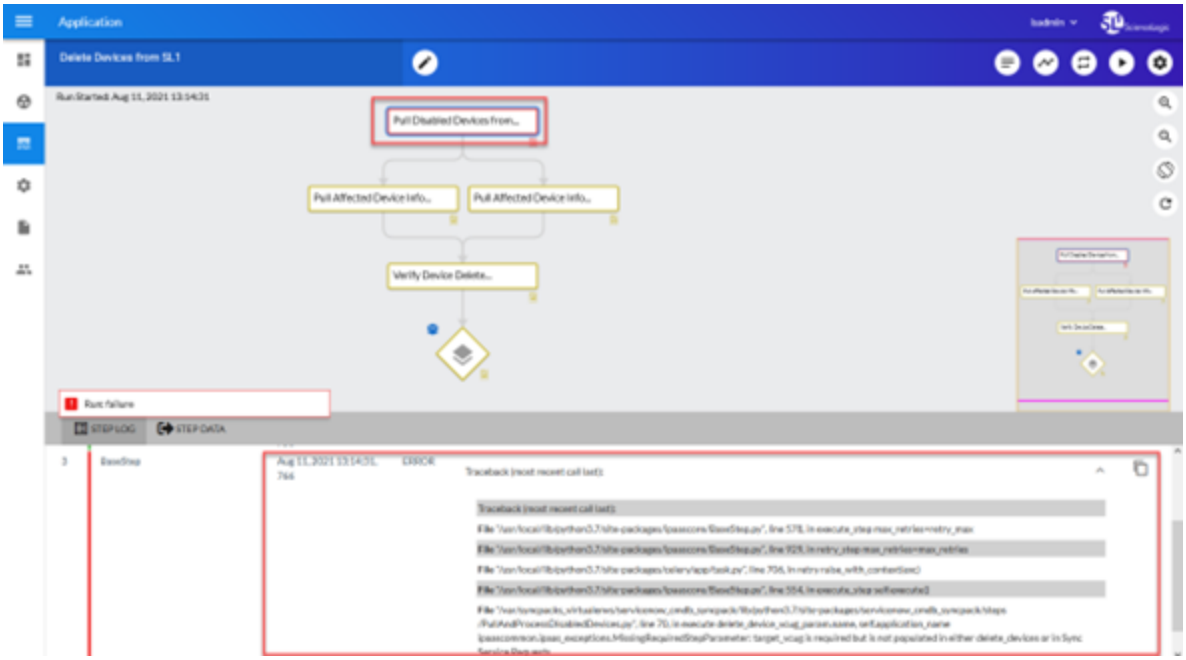
1. POST the following to the API endpoint:

```
https://<PowerFlow>/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 in to the PowerFlow instance.
2. Run the following command:

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

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.

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

International: +1-703-354-1010