SL1 PowerFlow Platform Release Notes

Version 2.2.0

Overview

The *SL1PowerFlow Platform* version 2.2.0 includes enhancements to the **Transform** component, a new **Trigger Application** component for building groups of steps in a PowerFlow application, and a new set of solution types for licensing the PowerFlow platform. This release also includes a number of addressed issues.

> **WARNING**: If you made any customizations to default applications or steps that shipped with previous versions of PowerFlow, you will need to make those customizations compatible with Python 3.6 or later *before* upgrading to version 2.0.0 or later from version 1.8.x of PowerFlow.

> **WARNING**: If you made any modifications to the nginx configuration or to other service configuration files outside of the **docker-compose.yml** file, you will need to modify those custom configurations before upgrading, or contact ScienceLogic Support to prevent the loss of those modifications.

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

This document describes:

- *Features included in version 2.2.0*
- *The issues addressed in version 2.2.0*
- *Known issues in version 2.2.0*
- *System requirements*
- *Installing the PowerFlow Platform*
- *Upgrading to this version of the PowerFlow Platform*

Features Included in this Release

The following features were included in *PowerFlow Platform* version 2.2.0:

- The **Transform** component in the PowerFlow builder user interface lets you take data from one step and modify or transform the data to fit into the next step in a PowerFlow application. You can also combine data from multiple steps and use that data in later steps in the application.

  The **Transform** component was updated with the following new features:

  - On the **Transform Wizard** page, you can drag and drop objects and arrays when building a transformation for an application. You can also view output data on this page after first loading the application, and error handling on this page was improved.

  - You can use the *Jinja/UI* toggle on the **Transform Wizard** page to switch between the *Jinja* code-only view or the *UI* drag-and-drop view. The default setting for the *Jinja/UI* toggle is the *UI* drag-and-drop view.

  - A *Search* field was added to the left-hand panel on the **Transform Wizard** page to help you find the data you want to transform.

  - The **Transform Wizard** page was updated to prevent a user from mixing filters and data types (strings, numbers, or arrays) when creating a Jinja template.

- The **Trigger Application** component in the PowerFlow builder user interface lets you launch another PowerFlow application from within a new or existing PowerFlow application. Step data from the triggered application is available in the parent application that triggered the application.

  The **Trigger Application** component includes the following features:

  - After you drag the **Trigger Application** component from the **[Advanced]** tab of the **Steps Registry** page, the **Trigger App** wizard page appears. From this page, you can select and configure an existing PowerFlow application that will be triggered by the current application.

  - You can drag and drop data and filters into specific parameters on the **Trigger App** wizard page.

  - You can also use Jinja2 templates in the step parameters for the **Trigger Application** component.

  - If a step triggers a child application, a blue information icon appears in the upper left corner of the step. Click the icon to display the triggered application's run as a link in a pop-up window, which lists the run IDs by Success, Failure, or In Progress. You can click the link in the pop-up window to view the detail page for the triggered application. If no run ID is present, the pop-up window displays "No runs available".

- The licensing for the PowerFlow platform was separated into three solution types:

  - *Standard*: This solution lets you import and install Synchronization PowerPacks published by ScienceLogic and ScienceLogic Professional Services, and to run and schedule PowerFlow applications from those PowerPacks. You cannot customize or create applications or steps with this solution type. Features that are not available are displayed in gray text.

- *Advanced*: This solution contains all of the Standard features, and you can also build your own Synchronization PowerPacks and upload custom applications and steps using the command-line interface. You can create PowerFlow applications using the PowerFlow command-line interface, but you cannot create and edit applications or steps using the PowerFlow builder.

- *Premium*: This solution contains all of the Advanced features, and you can also use the PowerFlow builder, the low-code/no-code, drag-and-drop interface, to create and edit PowerFlow applications and steps.

  For more information, see [ScienceLogic Pricing](#).

> **NOTE:** If you are upgrading from PowerFlow version 2.x.x to this release, you will need to upgrade your license to the Advanced or Premium solution to be able to upload custom content (such as steps and applications) for Synchronization PowerPacks, or to the Premium solution if you want to use the PowerFlow builder. Please note that this licensing update does not impact any solutions that are already installed on the PowerFlow system, and you can continue to run and schedule existing content as needed. For more information about the Advanced and Premium solutions, please contact your Customer Success Manger.

- A license expiration text box appears in the PowerFlow user interface when the license is close to expiring, displaying how many days are left before the license expires. The license status and expiration date also displays on the **About** page in the PowerFlow user interface.

- This release significantly improves performance related to task status querying, reducing the API memory requirements and increasing the responsiveness of the PowerFlow user interface. When a PowerFlow system queries for the state of a task, instead of pulling all relevant data from that task, the system only pulls the metadata that is needed at that time. For more information, see the "Configuring PowerFlow for Scalability" topic in the *Installing and Configuring SL1 PowerFlow* chapter of the **SL1PowerFlow Platform** manual.

> **NOTE:** If you are using a version of PowerFlow before version 2.2.0, or if you are upgrading to version 2.2.0, ScienceLogic recommends that you assess the current utilization of your PowerFlow system, because there will be a greater impact on the Couchbase database by default. If you find the database, or database node, is running at max CPU (80% or greater), you should be prepared to increase CPUs allocated to the Couchbase nodes, or deploy an additional database node to accommodate for the additional load. Alternatively, if the benefits of this feature are not desired, and you cannot afford to increase CPU in a large scale environment, you can disable it by setting **enable_state_backend: false** in the **environments** section in the **docker-compose.yml** file on the contentapi, steprunner, and syncpack_steprunner services. Disabling this feature will no longer store metadata in the Couchbase database, which removes the additional overhead on Couchbase, but you will not experience the benefits listed above.

- Added three new configuration variables to the **docker-compose** file that let you configure the number of retries for saving logs and log statuses in Couchbase. You can edit the default values of these variables to

improve the access to Couchbase when saving the application and step logs. The new variables include the following:

- *logs_timeout_retries*. Use this configuration variable as **env_var** in the **docker-compose** file for the workers, if any other number of retries is needed. The default is 3.
- *logs_delay_retries*. The delay of retries. The default is 2.
- *state_timeout_retries*. Number of retries for Celery. This value can be changed using the **env_var** in the **docker-compose** file for the workers, if any other number of retries is needed. The default is 3.

- Additional updates were made to the SL1PowerFlow user interface and the files related to PowerFlow to complete the branding transition from the "Integration Service" to "SL1 PowerFlow" and "PowerFlow builder":

  - Updated the PowerFlow user interface with PowerFlow branding, including changing the **Integrations** page to the **Applications** page and updating the **About** page.
  - Updated the RPM name, ISO name, and related scripts with PowerFlow branding. The RPM file name is now **sl1-powerflow-2.2.0-1.x86_64.rpm**, and the ISO file name is now **sl1-powerflow-2.2.0.iso**.
  - Updated the ISO installation wizard screens with PowerFlow branding.

- A "Refresh timer countdown" option was added to PowerFlow**Application** detail pages. This option lets you choose different update speeds when you run the application. The update speed specifies how soon the PowerFlow user interface reloads the application page from the API while the applications is running. You can specify intervals of 15, 30, or 60 seconds, and the default is 60 seconds.

- Added a **[Manual refresh]** button on the **Applications** page, along with a counter that counts down how many seconds remain before the next time PowerFlow automatically refreshes the page. Also, all **Application** detail pages were updated with a **[Manual refresh]** button that you can click to update the contents of the page.

- Added a **[Stop]** button to PowerFlow**Application** detail pages that lets you stop a started application and end all running tasks for that application.

- Added a new **[Delete all Messages]** button to the **Notification Center** pane.

- The **[Replay]** button on a **PowerFlow Application** page now has the following options:

  - *Debug Replay*. Replay the application in Debug Mode, which provides more log data in the Step Logs to help you with troubleshooting.
  - *Custom Replay*. Replay the run with a logging level that you specify, along with any customer parameters you might want to use.

- Added a **[Copy Log]** button to the **Step Log** pane in the PowerFlow user interface to help users copy logs and send them to ScienceLogic Support.

- The error messaging and error handling were improved for issues with Synchronization PowerPack installation.

- The "syncpack" property was added to application steps to identify the corresponding Synchronization PowerPack when you create a PowerFlow application in the PowerFlow user interface.

- The "start_time" property for the step logs is only available when **enable_state_backend** is set to **true** in the **environments** section of the **docker-compose.yml** file on the workers.

- The *Base Steps* Synchronization PowerPack version 1.2.0 is included with PowerFlow version 2.2.0, and the Synchronization PowerPack includes the following updates:

  - Updated the logging output of the "Trigger Application" step by removing duplicate log outputs, moving summary logs to Flow level, and moving more verbose logs to Info level.

  - Added a new step called "QueryREST: OAuth" was created that lets you input an OAuth 2.0 access token that will be used to authenticate with the PowerFlow API endpoint. This step is included in *the*

  - The "Query REST" step JSON code was updated to include the following help text for ***url_from_ input***: "These input fields for the URL must be added to the relative_url, not the prefix_url."

  - You can configure the existing "Query REST" step to use bearer authentication by adding the bearer token to the request headers in the **Configuration** pane for that step. The ***headers*** field should look like the following:
    ```
    {
        "Authorization": "Bearer <token_id>",
        "accept": "application/json",
        "content-type": "application/json"
    }
    ```

- The following services are included in this release of PowerFlow:

  - **contentapi**. sciencelogic/pf-api:release-2.2.0

  - **couchbase**. image: sciencelogic/pf-couchbase:6.0.2-3

  - **dexserver**. image: sciencelogic/pf-dex:2.22.0-1

  - **steprunner**. image: sciencelogic/pf-worker:release-2.2.0

  - **flower**. image: sciencelogic/pf-worker:release-2.2.0

  - **scheduler**. image: sciencelogic/pf-worker:release-2.2.0

  - **gui**. image: sciencelogic/pf-gui:release-2.2.0

  - **pypiserver**. image: sciencelogic/pf-pypi:4.8.1-5

  - **rabbitmq**. image: sciencelogic/pf-rabbit:3.7.27-4

  - **redis**. image: sciencelogic/pf-redis:4.0.14-2

  - **syncpacks_steprunner**. image: sciencelogic/pf-worker:release-2.2.0

TIP: To view release notes and manuals for previous versions of the SL1 PowerFlow Platform and PowerFlow Synchronization PowerPacks, see [SL1 PowerFlow Documentation](#).

Issues Addressed in this Release

The following issues were addressed in *PowerFlow Platform* version 2.2.0:

- Addressed an issue where you could not add or edit the **Customer CI Relation Override** field in the "Sync Devices from SL1 to ServiceNow" application. (Cases: 00115449, 00124366. JIRA IDs: EM-38992, INT-3154)

- Addressed an issue where adding invalid JSON values to the **Customer CI Relation Override** field in the "Sync Devices from SL1 to ServiceNow" application did not generate an error message. A red "X" icon now appears next to invalid JSON values on the **Show JSON Configs** pane for this application, and if you try to save an invalid JSON value, a pop-up appears to tell you that the error must be corrected before you can save.

- Addressed an issue where you could not save an application if the **Show JSON Configs** pane was visible.

- Addressed an issue where an unlicensed PowerFlow user was not able to create a configuration object, align a configuration object to a PowerFlow application, and save and run the application.

- Addressed an issue where users (including unlicensed users) could not edit the configuration settings for a PowerFlow application.

- Addressed an issue where some condition labels for an application did not display properly.

- Addressed an issue where the steps that came after a **Condition** operator in a PowerFlow application did not display as started or successfully run until you refreshed the **Application** page.

- Updated the *Troubleshooting SL1 PowerFlow* chapter of the **SL1 PowerFlow Platform** manual with a new FAQ topic about what to do when the Couchbase disk is full and the indexer is failing.

- Updated the "Troubleshooting Upgrade Issues" topic in the *Installation* chapter of the **SL1 PowerFlow Platform** manual with the following information about virtual environments: If the Docker container does not properly mount the virtual environment, or the virtual environment is not accessible to the environment, you might need to remove and re-deploy the service to resolve the issue.

Known Issues

This release contains the following known issues:

- If your PowerFlow system uses self-signed certificates, you will need to manually accept the certificate before you can upload Synchronization PowerPacks. Go to **https://<IP address of PowerFlow>:3141/isadmin**, accept the certificate, and then log into PowerFlow. After you log in, you will be able to upload Synchronization PowerPacks.

- The *latest* tag does not exist after the initial ISO installation. This situation only affects users with custom services that point to the *latest* tag. To work around this issue, run the tag latest script manually after running the ./pull_start_iservices.sh command:
  ```
  python /opt/iservices/scripts/system_updates/tag_latest.py
  /opt/iservices/scripts/docker-compose.yml
  ```

Known Issues in Previous Versions

The following known issue first appeared in version 2.0.2, and it was addressed in version 2.1.2:

- For scheduled PowerFlow applications, PowerFlow might run a scheduled application twice if the application is scheduled to run often, such as every 60 seconds. This situation can also occur if there are connection errors between the workers, the PowerFlow scheduler, or Couchbase.

The following known issues first appeared in version 2.1.0, and they were addressed in version 2.1.1:

- If you are using a new (non-upgrade) ISO installation of Integration Service version 2.1.0, new Synchronization PowerPack installations might fail with the following error message: "pip._ internal.exceptions.DistributionNotFound: No matching distribution found for ipaascommon". These Synchronization PowerPack installation failures occur if your Integration Service instance is connected to the Internet, as the 2.1.0 version of the Integration Service is trying to connect to a faulty version of pip.

- If you use the **ispasswd** script to change the Linux Host OS SSH password for a PowerFlow server, the script will fail and will not update the couchbase and pypiserver services password.

The following known issue first appeared in version 2.0.1, and it was addressed in version 2.0.2:

- You must edit the code in the JSON Configuration editor if you want to add data to the **customer_ci_ relation_overrides** application variable in the "Sync Devices from SL1 to ServiceNow" application. Click the **[Show JSON Configs]** button on the **Configuration** pane to access the JSON code.

System Requirements

The PowerFlow platform does not have a specific minimum required version for SL1. However, certain PowerFlow Synchronization PowerPacks have minimum version dependencies. Please see the documentation for those Synchronization PowerPacks for more information on those dependencies.

The following table lists the port access required by PowerFlow:

Source IP	PowerFlow Destination	PowerFlow Source Port	Destination Port	Requirement
PowerFlow	SL1 API	Any	TCP 443	SL1 API Access
SL1 Run Book Action	PowerFlow	Any	TCP 443	Send SL1 data to PowerFlow
Devpi	PowerFlow	Any	TCP 3141	Internal Python package repository for Synchronization PowerPacks; check for self-certification for PowerFlow
Dex Server	PowerFlow	Any	TCP 5556	Enable authentication for PowerFlow
PowerFlow	SL1 Database	Any	TCP 7706	SL1 Database Access
powerflowcontrol (pfctl, formerly called iservicecontrol) command-line utility	PowerFlow	Any	22 (on all host nodes)	Log in and perform admin tasks on nodes
Couchbase Dashboard	PowerFlow	8091	n/a	Couchbase Dashboard (use your PowerFlow credentials)
RabbitMQ Dashboard	PowerFlow	15672	n/a	Rabbit MQ Dashboard (use *guest/guest* for credentials)

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

CAUTION: While PowerFlow can manage PowerFlow applications between multiple SL1 systems with multiple third-party applications (such as ServiceNow, Cherwell, and Restorepoint), you can integrate only *one* SL1 system with a single third-party system. For example, if you were using two ServiceNow systems with PowerFlow, you would need *two* SL1 systems.

CAUTION: The site administrator is responsible for configuring the host, hardware, and virtualization configuration for the PowerFlow server or cluster. If you are running a cluster in a VMware environment, be sure to install open-vm-tools and disable vMotion.

NOTE: The default internal network used by PowerFlow services is **172.21.0.1/16**. Please ensure that this range does not conflict with any other IP addresses on your network. If needed, you can change this subnet in the **docker-compose.yml** file.

NOTE: The PowerFlow operating system is an Oracle Linux distribution, and all patches are provided within the standard Oracle Linux repositories. The patches are not provided by ScienceLogic.

TIP: For more information about system requirements for your PowerFlow environment, see the System Requirements page at the ScienceLogic Support site.

Installing PowerFlow

For detailed steps about installing PowerFlow, see the "Installing and Configuring PowerFlow" chapter in the *SL1PowerFlow Platform* manual.

> **WARNING:** If you are using the *ServiceNow CMDB* Synchronization PowerPack, this release of PowerFlow requires version 3.1.1 or later of that Synchronization PowerPack. You can download this Synchronization PowerPack from the [PowerPack Downloads](#) page.

Upgrading to this Version of PowerFlow

This topic explains at a high level how to upgrade to this version of PowerFlow. For the complete procedures for upgrading PowerFlow, see the "Upgrading PowerFlow" topic in the "Installing and Configuring SL1PowerFlow" chapter in the *SL1PowerFlow Platform* manual.

> **TIP:** As a best practice, you should *always* upgrade to the most recent version of PowerFlow that is currently available at the [PowerFlow Support](#) page.

> **WARNING:** If you made any customizations to default PowerFlow applications or steps that shipped with previous versions of PowerFlow, you will need to make those customizations compatible with Python 3.6 or later *before* upgrading to version 2.0.0 or later from version 1.8.x of PowerFlow.

> **WARNING:** If you made any modifications to the nginx configuration or to other service configuration files outside of the **docker-compose.yml** file, you will need to modify those custom configurations before upgrading, or contact ScienceLogic Support to prevent the loss of those modifications.

> **NOTE:** After upgrading to PowerFlow version 2.2.0, ensure that the PowerFlow platform is stable and that you do not wish to roll back to a previous version before you start installing or updating Synchronization PowerPacks. After you update the content on the PowerFlow platform, rolling back to a previous platform version requires restoring from a backup. After upgrading the platform to 2.2.0, the database version is updated to couchbase-community 6.0.2.x . You can roll back PowerFlow versions, but you should not roll back the database version.

> **NOTE:** The upgrade process is the same if you are upgrading from a system with the *Integration Service* naming to the new *SL1PowerFlow* naming. Only user-facing elements were changed, and internal packages were not changed. Although the user interface and user components are now labeled as "PowerFlow", internal directories, configuration files, and code packages might still use the "is_" suffix. PowerFlow container images will be available with both the **is-** and **pf-** tag on existing systems, so any existing custom **docker-compose** file should be unaffected, and may continue to use the **is-** tag, such as **is-worker**. Although both tags are available, you might want to eventually change any custom compose settings to use images with the **pf-** prefix, such as **pf-worker**, as the **is-** prefix eventually will be deprecated.

Upgrading from Version 2.x.x

To upgrade to the latest version of PowerFlow from version 2.x.x:

1. Download the PowerFlow RPM and copy the RPM file to the PowerFlow system.

2. Either go to the console of or use SSH to access the server.

3. Log in as *isadmin* with the appropriate (root) password. You must be root to upgrade using the RPM file.

4. Type the following at the command line:

   ```
   sudo rpm -Uvh full_path_of_rpm
   ```

   where *full_path_of_rpm* is the name and path of the RPM file, such as **/home/isadmin/sl1-powerflow-2.x.x-1.x86_64**.

   > **NOTE**: If you are running PowerFlow in a clustered environment, install the RPM on all nodes in the cluster before continuing with the remaining steps.

5. If the upgrade process recommends restarting Docker, run the following command:

   ```
   systemctl restart docker
   ```

6. After the RPM is installed, run the following Docker command:

   ```
   docker stack rm iservices
   ```

   > **NOTE:** If you want to upgrade your services in place, without bringing them down, you may skip this step. Please note that skipping this step might take the services slightly longer to update.

7. Re-deploy the Docker stack to update the containers:

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

8. After you re-deploy the Docker stack, the services automatically update themselves. Wait a few minutes to ensure that all services are updated and running before using the system. You can use the visualizer at port 8080 to monitor the progress of the updates.

9. To view updates to each service and the service versions for services throughout the whole stack, type the following at the command line:

   ```
   docker service ls
   ```

   You will notice that each service now uses the new version of PowerFlow.

Upgrading from Version 1.8.x

> **TIP:** As a best practice, you should *always* upgrade to the most recent version of PowerFlow that is currently available at the [PowerFlow Support](#) page.

To upgrade from PowerFlow version 1.8.x:

1. Upgrade the host packages and Python 3.6 (previous versions of PowerFlow used Python 2.6).
2. Upgrade to Oracle 7.3 or later.
3. Upgrade to Docker version 18.09.2 or later.

> **NOTE:** PowerFlow version 2.0.0 or later requires the **docker-ce 18.09.2** or later version of Docker. The PowerFlow ISO installs the **docker-ce 19.03.13** version of Docker by default, but if you are upgrading to this version from the RPM, you must first upgrade Docker before you upgrade PowerFlow with the RPM.

4. Install the PowerFlow upgrade RPM.
5. Update the PowerFlow system from Basic Authentication to OAuth 2.0.
6. Set up licensing for PowerFlow. After installation, you must license your PowerFlow system to enable all of the features. For more information, see the "Managing Licenses for PowerFlow" chapter in the *SL1PowerFlow Platform* manual.

> **NOTE:** If you are not deploying PowerFlow on a production or pre-production environment, you can skip the licensing process.

Please note that upgrading to the latest version from 1.8.x *will* involve some downtime of PowerFlow.

Using the Upgrade Script

You can perform the upgrade steps manually, or you can run the **is_upgrade_to_v2.sh** script to perform the upgrade steps automatically. The script upgrades PowerFlow from 1.8.x to this release.

To locate the upgrade script:

1. At the [ScienceLogic Support site](#), click the **Product Downloads** tab and select *PowerFlow*. The **PowerFlow Release** page appears.
2. Click the "Integration Service 2.0" link. The **Release Version** page appears.
3. In the **Release Files** section, click the "1.8.X to 2.X.X Upgrade Script" link. A **Release File Details** page appears.
4. Click **Download File** on the **Release File Details** page. The **is_upgrade_to_v2.sh** script is in the **is_upgrade_tools.zip** file.