



Skylar One Base SyncPack

Version 1.1.0

Table of Contents

Introduction to the Skylar One Base SyncPack	3
Contents of the Skylar One Base SyncPack	4
MCP Components in Version 1.1.0	4
MCP Components in Version 1.0.0	6
Installing the Skylar One Base SyncPack	7
Prerequisites for the Skylar One Base SyncPack	8
Downloading the SyncPack	8
Importing the SyncPack	8
Installing the SyncPack	9
Configuring the Fetch Skylar One Organizations Sync for the Skylar One Base SyncPack	10
Creating and Aligning a Configuration Object	11
Creating a Configuration Object	11
Aligning a Configuration Object	12
Syncing Skylar One Organization Data	13
Scheduling Skylar Automation Applications	14

Chapter

1

Introduction to the Skylar One Base SyncPack

Overview

This chapter describes the "Skylar One Base" SyncPack, which contains a set of tools used by the Model Context Protocol (MCP) server introduced in version 3.4.0 of Skylar Automation.

This chapter covers the following topics:

<i>Contents of the Skylar One Base SyncPack</i>	4
---	---

Contents of the Skylar One Base SyncPack

This section lists the contents of the "Skylar One Base" SyncPack.

The following MCP tools and their filters work in conjunction with the MCP server introduced in version 3.4.0 of Skylar Automation. For more information, see the [Configuring the Skylar Automation MCP Server](#) chapter in the *Skylar Automation* manual.

IMPORTANT: All MCP tools listed below require a ***config_name*** that points to a Skylar Automation configuration object with the following fields:

- ***sl1_host***. The hostname or IP address of the Skylar One instance.
- ***sl1_basic_auth_token***. Base64-encoded Basic authorization credentials.

MCP Components in Version 1.1.0

The following tools let an AI assistant query your Skylar One environment directly, including devices, business services, topology, metrics, events, and run book automations.

NOTE: 19 of the 20 tools in this release are read-only. The ***initiate_event_runbook_automation*** changes external state as it executes an automation against your Skylar One instance.

- Devices
 - ***get_device_status***. Returns current status and severity for one device (***device_id***).
 - ***find_devices***. Finds devices by ***name***, ***ip_address***, or ***device_class***; partial match on any combination, at least one is required.
 - ***get_device_summary***. Returns an operational summary for one device (***device_id***): hostname, IP, status, severity, device class, organization, date added.
 - ***get_direct_related_devices***. Returns direct topology relationships for a device (***device_id***), paginated via cursor.
- Business Services
 - ***find_business_services***. Finds business services by name or label, at least one required.
 - ***get_business_service_summary***. Returns operational detail for one business service (***service_id***): health, availability, risk, description, organization.
 - ***list_device_business_services***. Lists business services directly related to a device (***device_id***), paginated.

- ***get_business_service_topology***. Returns a bounded hierarchy (nodes/edges) for a business service (**service_id**), with configurable **depth** (1-4) and **max_nodes** (up to 250); flags `truncated: true` if the hierarchy exceeds the node cap.
- Locations
 - ***get_entity_locations***. Returns configured latitude and longitude for a mix of devices (**device_ids**) and business services (**business_service_ids**, up to 25 each). Entities without configured coordinates land in unlocated with a reason instead of failing the call.
- Metrics and Performance
 - ***list_available_device_metrics***. Lists collected metrics for a device (**device_id**), including which Dynamic Application each comes from. Optional name filter.
 - ***get_device_metrics***. Returns time-series history for up to 10 named metrics on one device (**device_id**, **metric_names**, **duration** such as 1h/24h).
 - ***compare_device_metrics***. Compares up to 5 named metrics across 2-10 devices on one time series. Devices that do not collect a requested metric land in **unavailable_devices** with a reason rather than failing the whole call.
- Events and Automations
 - ***list_available_event_runbook_automations***. Lists enabled, user-selectable Run Book Automation policies currently available for given **event_ids**. Read-only, never initiates anything.
 - ***list_device_events_with_runbook_automations***. Lists active events for one or more devices (**device_ids**, up to 10) plus the automations available for each, with optional **minimum_severity** filter.
 - ***initiate_event_runbook_automation***. Not read-only. Executes a specific, user-selected run book automation (**runbook_policy_id**) against a specific active event (**event_id**). Revalidates that the automation is currently enabled and available for that exact event before executing, and rejects stale or invalid policy IDs.
- Organization and Inventory
 - ***list_organizations***. Lists Skylar One organizations (ID, name; description is always null. The SL1 organization resource has no description field).
 - ***list_device_classes***. Lists device classes (vendor/model), with name and description populated.
 - ***get_device_interfaces***. Returns network interfaces for a device (**device_id**): name, status, speed, IP address, MAC address.
 - ***get_device_logs***. Returns log entries for a device (**device_id**): timestamp, severity level, message, source.
 - ***get_device_config_data***. Returns configuration data collections for a device (such as BGP peers, OSPF neighbors, IP SLA operations), with human-readable parsed values (such as BGP state "Established" rather than a raw code). Optional **config_name_filter**; set `fetch_data=false` to get collection metadata only.

MCP Components in Version 1.0.0

The following tools let an AI assistant query your Skylar One environment directly:

- ***list_devices***. Returns a paginated list of devices from Skylar One. All filters are optional and can be combined. Filters include:
 - ***device_id***. Integer. Filters to a specific device.
 - ***severity_level***. Integer of one to five. The minimum severity to return. Five is the most severe.
 - ***device_class***. String. Filters by device class name.
 - ***collector_group_name***. String. Filters by collector group name.
 - ***limit***. Integer between 1 and 50. Defines results per page. The default value is 10.
 - ***cursor***. String. Pagination cursor from a previous response.
- ***list_events***. Returns a paginated list of active events, ordered by the most recent first. Filters include:
 - ***device_id***. Integer. Filters events to a specific device.
 - ***severity_level***. Integer of one to five. The minimum severity to return. Five is the most severe.
 - ***limit***. Integer between 1 and 50. Defines results per page. The default value is 10.
 - ***cursor***. String. Pagination cursor from a previous response.
- ***get_asset_by_device_id***. Returns asset records (hardware inventory) associated with a device. Filter includes:
 - ***device_id***. Integer. Required. The ID of the device that contains the asset records.
- ***list_business_services***. Returns a paginated list of business services. All filters are optional. Filters include:
 - ***risk_level***. Integer of one to five. The minimum risk level to return.
 - ***health_level***. Integer of one to five. The minimum health degradation level to return.
 - ***unavailable_only***. Boolean. If true, only returns unavailable services.
 - ***name_contains***. String. Filters by partial name matches.
 - ***limit***. Integer between 1 and 50. Defines results per page. The default value is 10.
 - ***cursor***. String. Pagination cursor from a previous response.
- ***get_appliance_status***. Returns status information for all appliances in the Skylar One instance (excludes Collector Unit and Message Collector appliance types). No additional parameters.

TIP: All paginated tools return a cursor in the response. Pass the cursor as the cursor parameter on the next call to fetch the next page.

Chapter

2

Installing the Skylar One Base SyncPack

Overview

This chapter describes how to download, import, and install the "Skylar One Base" SyncPack.

This chapter covers the following topics:

<i>Prerequisites for the Skylar One Base SyncPack</i>	8
<i>Downloading the SyncPack</i>	8
<i>Importing the SyncPack</i>	8
<i>Installing the SyncPack</i>	9

Prerequisites for the Skylar One Base SyncPack

This release requires the following ScienceLogic components:

- Skylar Automation platform version 3.4.0 or later
- Skylar One version 12.5.1 or later
- "Base Steps" SyncPack version 1.7.0 or later

IMPORTANT: You should always use the most recent version of a SyncPack.

For more information about configuring the MCP server, see the [Configuring the Skylar Automation MCP Server](#) chapter in the *Skylar Automation* manual.

Downloading the SyncPack

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

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

To locate and download the SyncPack:

1. Go to the ScienceLogic Support Center at <https://support.sciencelogic.com/s/>.
2. Go to the **SyncPacks** page (Skylar Automation > 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 Skylar Automation system using the Skylar Automation user interface.

Importing the SyncPack

To import a SyncPack in the Skylar Automation user interface:

1. On the **SyncPacks** page (☹) of the Skylar Automation 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]**. Skylar Automation registers and uploads the SyncPack. The SyncPack is added to the **SyncPacks** page.
4. You will need to activate and install the SyncPack in Skylar Automation. 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 Skylar Automation user interface:

1. On the **SyncPacks** page of the Skylar Automation 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 Skylar Automation 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 highlighted version in the **Installed SyncPack** 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, go to the hidden "activate & Install SyncPack" application on the **Applications** page and check the step logs.
4. If you have other versions of the same SyncPack on your Skylar Automation 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.

Chapter

3

Configuring the Fetch Skylar One Organizations Sync for the Skylar One Base SyncPack

Overview

This chapter describes the how to configure and run the "Fetch Skylar One Organizations" Sync in the "Skylar One Base" SyncPack.

This chapter covers the following topics:

<i>Creating and Aligning a Configuration Object</i>	<i>11</i>
<i>Syncing Skylar One Organization Data</i>	<i>13</i>
<i>Scheduling Skylar Automation Applications</i>	<i>14</i>

Creating and Aligning a Configuration Object

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

You can create as many configuration objects as you need. A Skylar Automation 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 create one or more configuration objects in the Skylar Automation user interface and align that configuration object to the applications that let you sync data between Skylar One and Skylar Automation.

NOTE: Each unique stack that includes a Skylar One and Skylar Automation instance requires an individual configuration object with a unique region.

Creating a Configuration Object

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

TIP: The "Skylar One Base Config" 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 Skylar One settings.

To create a configuration object based on the "Skylar One Base Config" configuration object:

1. In the Skylar Automation user interface, go to the **Configurations** page ().
2. For the "Skylar One Base Config" configuration object, click the **[Actions]** button () 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.

IMPORTANT: This step is required. Do *not* use the original configuration object to run Skylar Automation 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": "sll_db_host",
  "value": "${config.sll_host}"
},
```

For example:

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

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

NOTE: The **region** value is a user-defined variable that identifies your Skylar One instance.

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

Aligning a Configuration Object

Before you can run the applications in this SyncPack, you must first "align" a configuration object with the application you want to use.

To align a configuration object with an application:

1. From the **Applications** page of the Skylar Automation user interface, open the relevant application and click **[Configure]**. The **Configurations** pane for that application appears.
2. From the **Configurations** drop-down, select the configuration object you want to use.
3. Click **[Save]** to align that configuration with the application.
4. Repeat this process for every other application you want to use.

NOTE: The values for the parameters that display in the **Configuration** pane with a padlock icon () are populated by the values in the configuration object.

Syncing Skylar One Organization Data

The "Fetch Skylar One Organizations" application retrieves organization data from the Skylar One API, serializes it, and stores all organization records in the cache. The cache uses the region name as a unique identifier to ensure region-specific data isolation. The cached organization data serves as a shared source for all other SyncPacks, allowing them to retrieve and utilize the appropriate organization information based on the specified region.

IMPORTANT: If you want to use the domain-level filtering and routing added in version 3.2.0 of the "Skylar Compliance" SyncPack to allow multiple Skylar One instances to connect to a single Skylar Compliance instance, you must configure and run the "Fetch Skylar One Organizations" application first.

To sync Skylar One organization data:

1. In the Skylar Automation user interface, go to the **Applications** page and select the "Fetch Skylar One Organizations" application. The **Application** page for that application appears.
2. Click **[Configure]**. The **Configuration** pane appears.
3. Complete the following fields, as needed:
 - **Configuration.** Select the configuration object with the relevant Skylar One and credentials to align with this application. You cannot edit fields that are populated by the configuration object. Required.

IMPORTANT: The **region** field in the configuration object is linked to this application. It serves as a unique identifier for the Skylar Compliance, Skylar Automation, and the Skylar One stack.

- **sl1_chunk_size.** Specify the number of organizations to include in each chunk pulled from Skylar One.

- ***read_timeout***. Specify the maximum amount of time in seconds that the application should wait for a response before timing out.
4. Click **[Save]**. The **Configuration** pane automatically closes after this message appears.
 5. Click **[Run]** to run the application.
 6. When the application completes, open the **Step Log** and review the **Outgoing Data** to view the organization data collected from Skylar One. As needed, select the other steps to review the logs on the **Step Log** for those steps.

Scheduling Skylar Automation Applications

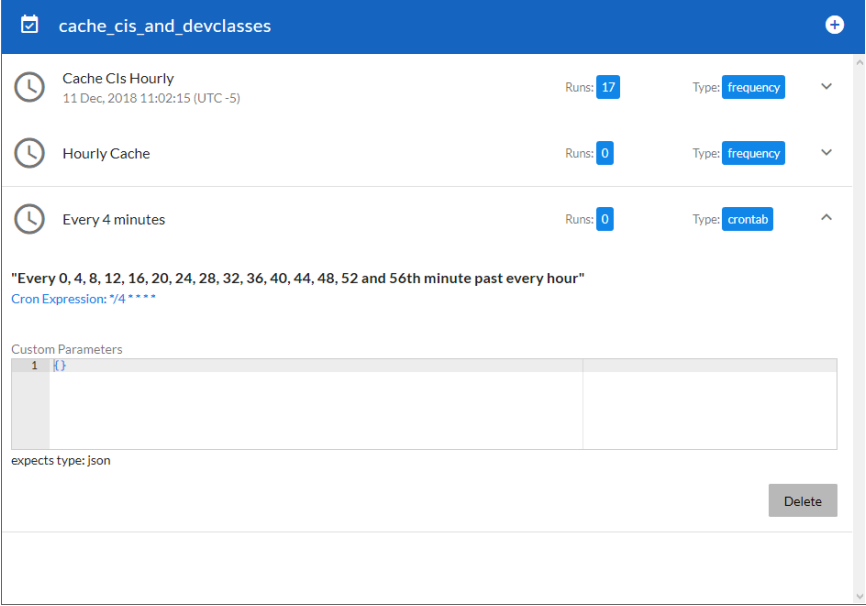
Using the Skylar Automation user interface, you can configure Skylar Automation applications to run on a schedule instead of manually running the applications. As a best practice, if you use any of these applications, ScienceLogic recommends that you schedule those applications, in the following order:

TIP: ScienceLogic recommends that you schedule the "Fetch Skylar One Organizations" application to run at least every 23 hours.

You can create one or more schedules for a single application in the Skylar Automation user interface. When creating each schedule, you can specify the queue and the configuration file for that application.

To schedule an application:

1. On the **Applications** page () , click the **[Schedule]** button for the application you want to schedule. The **Schedule** window appears, displaying any existing schedules for that application:



cache_cis_and_devclasses

Cache CIs Hourly	11 Dec, 2018 11:02:15 (UTC -5)	Runs: 17	Type: frequency
Hourly Cache		Runs: 0	Type: frequency
Every 4 minutes		Runs: 0	Type: crontab

"Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52 and 56th minute past every hour"
Cron Expression: */4 * * * *

Custom Parameters

1	{ }
---	-----

expects type: json

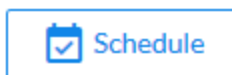
Delete

NOTE: If you set up a schedule using a cron expression, the details of that schedule display in a more readable format in this list. For example, if you set up a cron expression of `*/4 * * * *`, the schedule on this window includes the cron expression along with an explanation of that expression: "Every 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, and 56th minute past every hour".

2. Select a schedule from the list to view the details for that schedule.
3. Click the + icon to create a schedule. A blank **Schedule** window appears:

The screenshot shows a window titled "cache_cis_and_devclasses" with a red close button. Inside is a "Schedule" form. At the top, there's a "Schedule Name" label and a text input field. To the right is a "Switch to Cron Expression" toggle switch. Below this is a "Frequency" label and a text input field with a dropdown menu showing "secs". Underneath is a "Custom Parameters" section with a table. The table has 3 rows and 2 columns. The first row contains "1" and "1". The second row contains "2" and "()". The third row contains "3" and "1". Below the table, it says "expects type: json". At the bottom right is a blue "Save Schedule" button.

4. In the **Schedule** window, complete the following fields:
 - **Schedule Name.** Type a name for the schedule.
 - **Switch to.** Use this toggle to switch between a cron expression and setting the frequency in seconds.
 - **Cron expression.** Select this option to schedule the application using a cron expression. If you select this option, you can create complicated schedules based on minutes, hours, the day of the month, the month, and the day of the week. As you update the cron expression, the **Schedule** window displays the results of the expression in more readable language, such as *Expression: "Every 0 and 30th minute past every hour on the 1 and 31st of every month", based on */30 * */30 * **.
 - **Frequency in seconds.** Type the number of seconds per interval that you want to run the application.
 - **Custom Parameters.** Type any JSON parameters you want to use for this schedule, such as information about a configuration file or mappings. The values in this field will override any application variables that exist in the PowerFlow application you are scheduling.
5. Click **[Save Schedule]**. The schedule is added to the list of schedules on the initial **Schedule** window. Also, on the **Applications** page, the word "Scheduled" appears in the **Scheduled** column for this application, and the **[Schedule]** button contains a check mark:



NOTE: After you create a schedule, it continues to run until you delete it. Also, you cannot edit an existing schedule, but you can delete it and create a similar schedule if needed.

To view or delete an existing schedule:

1. On the **Applications** page, click the **[Schedule]** button for the application that contains a schedule you want to delete. The **Schedule** window appears.
2. Click the down arrow icon (▼) to view the details of an existing schedule:

The screenshot shows a web interface for managing application schedules. The header bar is blue and contains the application name 'cache_cis_and_devclasses' and a plus icon. Below the header, there is a list of three schedules:

- Cache CIs Hourly**: 11 Dec, 2018 11:02:15 (UTC -5), Runs: 17, Type: frequency, with a down arrow icon.
- Hourly Cache**: Runs: 0, Type: frequency, with a down arrow icon.
- Every 4 minutes**: Runs: 0, Type: crontab, with an up arrow icon.

The 'Every 4 minutes' schedule is expanded, showing the following details:

- Cron Expression**: `*/*4 * * * *`
- Custom Parameters**: A table with one row and one column, containing the value `1`.
- expects type**: json
- Delete**: A button to delete the schedule.

3. To delete the selected schedule, click **[Delete]**.

© 2003 - 2025, ScienceLogic, Inc.

All rights reserved.

ScienceLogic™, the ScienceLogic logo, and ScienceLogic's product and service names are trademarks or service marks of ScienceLogic, Inc. and its affiliates. Use of ScienceLogic's trademarks or service marks without permission is prohibited.

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

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



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

International: +1-703-354-1010