Gravio Documentation - The Node Computing Platform

1 — Last update: 5 September 2023

ASTERIA Corporation

Table of Contents

1. Gravio Architecture and Terminology

Gravio can be installed and operated on cloud services such as AWS, Azure, GCP, etc. (hereinafter described as “cloud”) starting from version 5 . Gravio can now be installed and operated on these services as well.

A node on which the Gravio HubKit is installed is called a “Node hub” or “node” starting with Gravio version 5.

By installing the Gravio HubKit package on the cloud service, this server can be used as a Node hub server to aggregate data .
This Node hub server is referred to as the “Main server”.

Note: To operate a Main server, you need a “Main License” granted by your Gravio Business or Bussiness Pro service contract.

Gravio Business/Bussiness Pro customers can also use the Gravio Cloud servers we provide.

Modules running on Gravio Main server

The following modules included in the HubKit package reside on the Gravio Main server as a server.

Module Name	Description
Gravio Coordinator	encompasses the management of the Node hub and the aggregate database of sensor data.
Gravio HubKit	a set of modules for collecting and managing sensor data and other data, including action engines
Gravio Configuration Manager	Provides HubKit settings and initial configuration functions for Gravio coordinators.

modules that work with the Gravio Node Hub

The following modules included in the HubKit package reside as servers in the Gravio Node Hub.

Module Name	Description
Gravio HubKit	A set of modules for collecting and managing sensor data and other data, including action engines
Gravio Configuration Manager	Provides HubKit settings and initial configuration functions for Gravio Coordinators, etc.

HubKit configuration and Action Flow editing tools

In Gravio, “Action Flows” (or “Actions”) You can run a user-created data flow program called (marker-

blue) on HubKit.

A GUI application called “Gravio Studio” is available for Windows and Mac as a tool for editing HubKit settings and action flows.

Module Name	Description
Gravio Studio	HubKit configuration and action flow editing tool

Gravio Studio can be used to manage and configure the following

*Setup of physical sensor devices. Mapping of where they are located within the physical space.
*Management, monitoring and maintenance of sensors.
*Creating and configuring actions triggered by the sensor network.
*Creating and editing action flows.
*Data display of raw sensor data.

Gravio.com

There is a portal site for viewing and managing your Gravio subscription information and license usage. When you log in to Gravio.com from Gravio Studio, you can use various file backup services from Gravio Studio’s configuration panel to gravio.com.

Remarks

Gravio HubKit consists of several server modules.

Server Name	Description
Gravio Control Manager	Server that controls communication with Gravio Studio
Gravio Trigger Manager	A server that receives various sensor data from each device management manager server (Serial Port, Bluetooth, Video, etc.) and saves the sensor data. It also checks incoming data values for conditions specified in the trigger settings and, if applicable, requests the Gravio Action Manager to execute them.
Gravio Action Manager	Server that executes the action and saves the sensor data.
Gravio Serial Port Manager	Responsible for managing the connection with physical sensors via serial ports and sending data acquired from sensors to Gravio Trigger Manager.
Gravio Bluetooth Manager	Manages connections with physical sensors connected via Bluetooth and sends data acquired from sensors to Gravio Trigger Manager.

Gravio Video Service	This service manages the connection to the Onvif Camera, stores images/video from the camera, and sends data acquired from image inference to the Gravio Trigger Manager.
Gravio MQTT Service	Responsible for sending data acquired from MQTT to Gravio Trigger Manager.
Gravio Image Processing Manager	Responsible for sending data acquired from image processing servers such as IVAR to Gravio Trigger Manager.
Gravio App Service	A server with auxiliary functions for executing components

2. Upgrade Instructions for Gravio 5.0 from Gravio 4.x

This section provides instructions for upgrading from Gravio 4.x to Gravio 5.0.

Gravio 5.0 simplifies the process of deploying license files, and now allows users to obtain licenses in the Gravio Configuration Manager.

For this reason, customers with Gravio 4.x licenses will need to change their license format to that for Gravio 5.0.

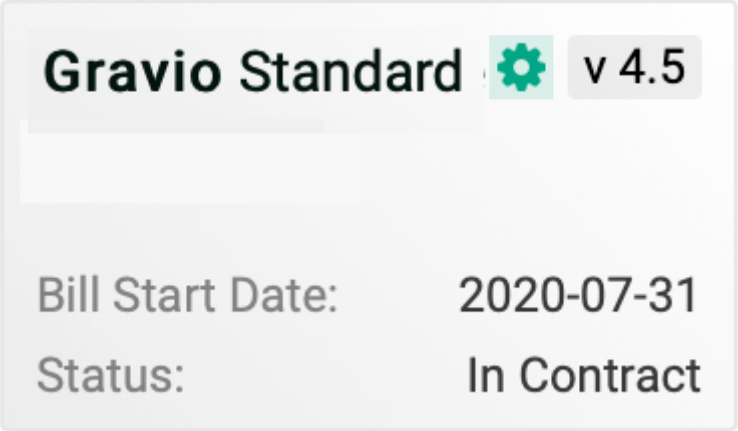
Please login to gc1.gravio.com to change your license format to 5.0 and then install the HubKit package for each platform.

2.1. Set Gravio 4.x license format for 5.0

Before upgrading your HubKit package, please configure your 4.x license format to 5.0. You can do this in your online portal.

Log in to https://gc1.gravio.com/ui/login.

After logging in, in the “Account” tab, find this license tile:



Click on the green cogwheel in order to change the license version. The “Change License Version” dialog box will appear. Press “Set” to change the license version to 5.0.

2.2. Upgrading Gravio on Windows

How to upgrade HubKit

1. Backup your folder `/Program Data/HubKit/`
2. Uninstall HubKit via the Windows uninstaller
3. Download HubKit Windows 5.0 version from https://www.gravio.com/en/download
4. Install the downloaded Gravio HubKit using the installer
5. Follow the steps in Initial Setup

2.3. macOS version

upgrade HubKit.

1. Stop HubKit in the Gravio HubKit Menubar app that appears on the menubar
2. Quit the Gravio HubKit Menubar app.
3. Backup `/Library/Application Support/HubKit`.
4. Move the Gravio HubKit4 in the Applications folder to Trash and empty the trash.
5. Download HubKit Mac version from https://www.gravio.com/en/download.
6. Double-click the downloaded GravioHubKit.dmg file to open it.
7. Drag and drop the Gravio HubKit icon into the Applications folder.
8. Double-click Gravio HubKit in the Applications folder to launch it.
9. Click on the “Gravio Initial Settings” menu from the Gravio HubKit toolbar application menu that appears on the toolbar.
10. Follow the steps in Initial Setup.

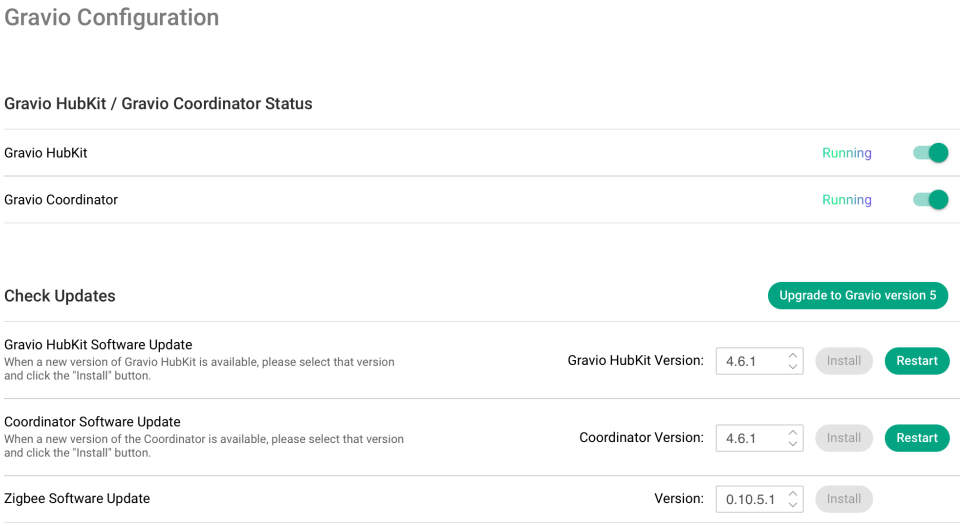
2.4. Gravio Hub 2

Upgrading HubKit

Please upgrade HubKit following the steps below. If the upgrade is successfully completed, you will not use the backed-up data.

1. Log into Update Manager

When the Update Manager is updated to the latest version with the automatic update at midnight, the Upgrade to Gravio Version 5 button is displayed on the Gravio settings tab, so please check.



2. Backup from HubKit / Coordinator Data Backup

Please back up your HubKit data.

3. Upgrade to Gravio Version 5

Please press the Upgrade to Gravio Version 5 button. After the confirmation dialog is displayed, it will automatically restart.
After rebooting, the Update Manager will be displayed as the Configuration Manager.

Please refer to here as the initial setup of Gravio will be done from the Gravio Configuration Manager.

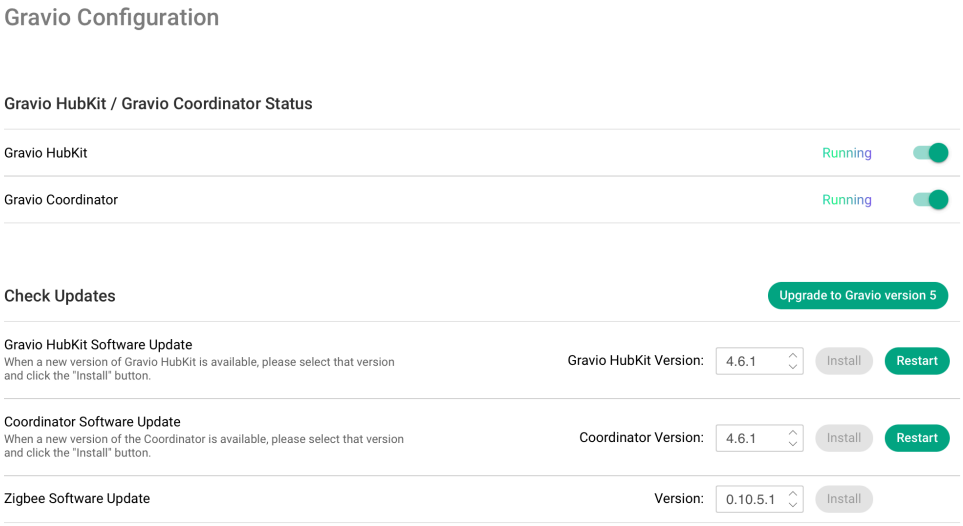
2.5. Linux (Ubuntu AMD64) version

Upgrading HubKit

Please follow the steps below to upgrade HubKit.
If the upgrade is completed successfully, you will not need the data you backed up.

1. Log into Update Manager

When the Update Manager is updated to the latest version with the automatic update overnight, the Gravio settings tab will display a button labeled “Upgrade to Gravio Version 5.” Please check this.



2. Back up from HubKit / Coordinator Data Backup

Please back up your HubKit data.

3. Upgrade to Gravio Version 5

Please press the “Upgrade to Gravio Version 5” button. After the confirmation dialog is displayed, it will automatically restart.
After rebooting, the Update Manager will be displayed as the Configuration Manager.

Please refer to here for the initial setup of Gravio from the Gravio Configuration Manager.

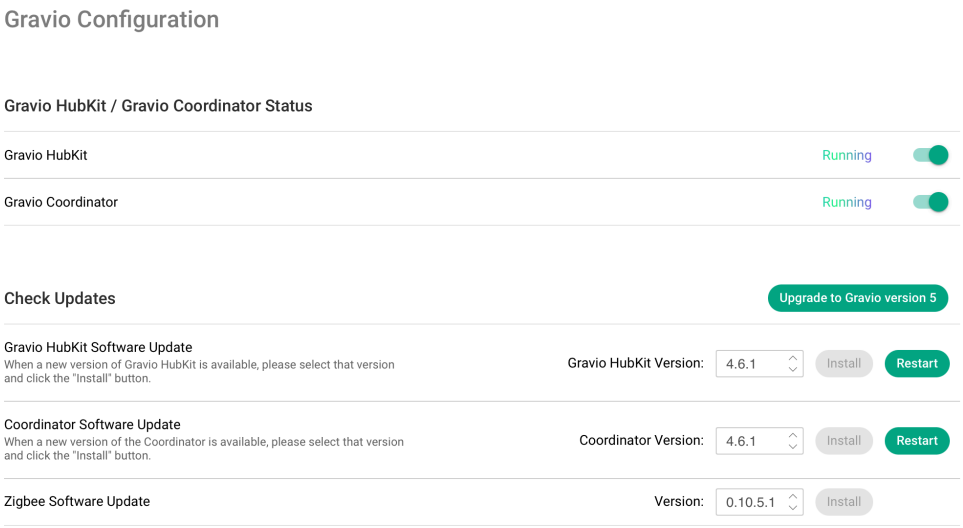
2.6. Linux (Ubuntu ARM64/ Raspberry Pi) Version

Upgrading HubKit

Please follow the steps below to upgrade HubKit.
If the upgrade is completed successfully, you will not need the data you backed up.

1. Log into Update Manager

When the Update Manager is updated to the latest version with the automatic update overnight, the Gravio settings tab will display a button labeled “Upgrade to Gravio Version 5”. Please check this.



2. Backup from HubKit / Coordinator Data Backup

Please back up your HubKit data.

3. Upgrade to Gravio Version 5

Please press the “Upgrade to Gravio Version 5” button. After the confirmation dialog is displayed, it will automatically restart.
After rebooting, the Update Manager will be displayed as the Configuration Manager.

Please refer to here for the initial setup of Gravio from the Gravio Configuration Manager.

3. Installation and Setup

Gravio operates with two pieces of software installed: the server software, the “Gravio HubKit” package , and a client application called “Gravio Studio” that operates the server software. as a client application to operate the server software.

Gravio HubKit is available on Linux, Windows, and Mac. They can be downloaded from https://www.gravio.com/jp/download.
Please download the installer for your operating system.

Gravio Studio is available from the MS Store app for Windows and the App Store app for Mac, so please search for “Gravio Studio” there and install the application.

Note : If you are installing the Windows version of the Gravio HubKit package, be sure to install Gravio Studio from the MS Store first.

3.1. Determining how to operate the Gravio Hub 2 and the steps to set it up

Please decide first on how to operate Gravio Hub 2.

Please decide whether you will operate Gravio Hub 2 as a (marker-blue) “server for node management” or as a (marker-blue) “Node hub”.

The settings are made from a browser on the screen of the Gravio Configuration Manager.

1. Set the WiFi settings for Gravio Hub on the Gravio Hub Settings screen to connect Gravio Hub 2 to the internet.
2. Obtain a Gravio license on the Initial Settings screen

- To make Gravio Hub 2 a server for node management, obtain a MAIN license and perform initial setup of the Gravio coordinator
  Note: Even if you set it as a server for node management, this management server can also operate as one of the nodes, allowing you to acquire sensor data and execute actions.

- If you want to use Gravio Hub 2 as a Node hub, obtain a NODE license and register this Node hub under the control of the Gravio coordinator

With this, the setting of the operation environment is complete.

3.1.1. Gravio Hub 2 Installation

This section describes the installation of Gravio Hub 2.

To use Gravio Hub 2, the following three points are necessary:

1. Connect Gravio Hub to LAN
2. Gravio Hub Settings
3. Initial Settings

Gravio Hub 2 operates on a Linux OS based on Ubuntu 2004. It is designed to allow you to log in to Linux OS as needed. For this reason, in the initial settings of the Gravio Configuration Manager to enhance security, you will be forced to change the password of the login account for logging into Linux. Please prepare the settings for the WiFi network (the network name to be connected and the password) and the new password for the login account.

Note: Please store the login account name and password safely. If you lose the password, you will not be able to log in to the Linux machine again and may have to reset it, which may cause data loss.

The following accessories, including Gravio Hub, are included with Gravio Hub 2 package:

- The Gravio Hub 2 device:

- Gravio USB-C cable
- The power adapter

Gravio Hub 2 has one USB-A connector, USB-C connector, wired LAN connector, and HDMI connector each.

3.1.2. Connecting the Gravio Hub to a LAN

For wired LAN connection

Connect the wired LAN cable to the LAN connector on the Gravio Hub2.

Note: Please use a wired LAN that can provide an IP address via DHCP.

Boot up the Gravio Hub2

Connect the USB-C cable to the USB-C connector on the Gravio Hub2 and start it up.

For wireless LAN connection

Boot up the Gravio Hub2

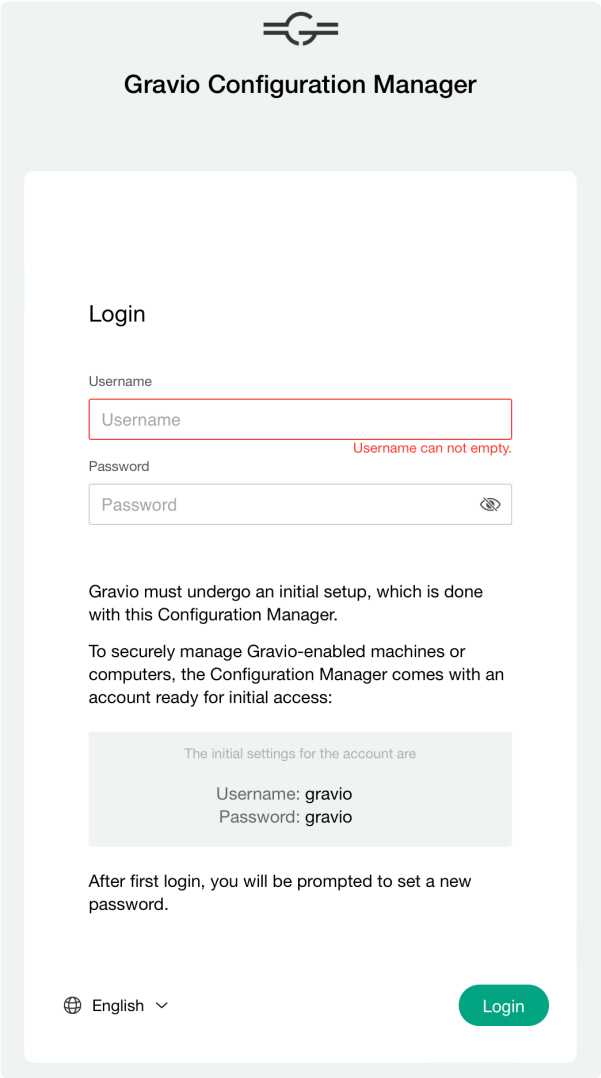
Connect the USB-C cable to the USB-C connector on the Gravio Hub2 to boot up.

Configure Wifi settings on Gravio Hub2

Connect the Gravio Hub to a power source and start it up.

When the light at the front of the Gravio Hub appears blue, connect to the SSID named GravioHubXXXXXXXXXXXX from your PC’s Wifi.

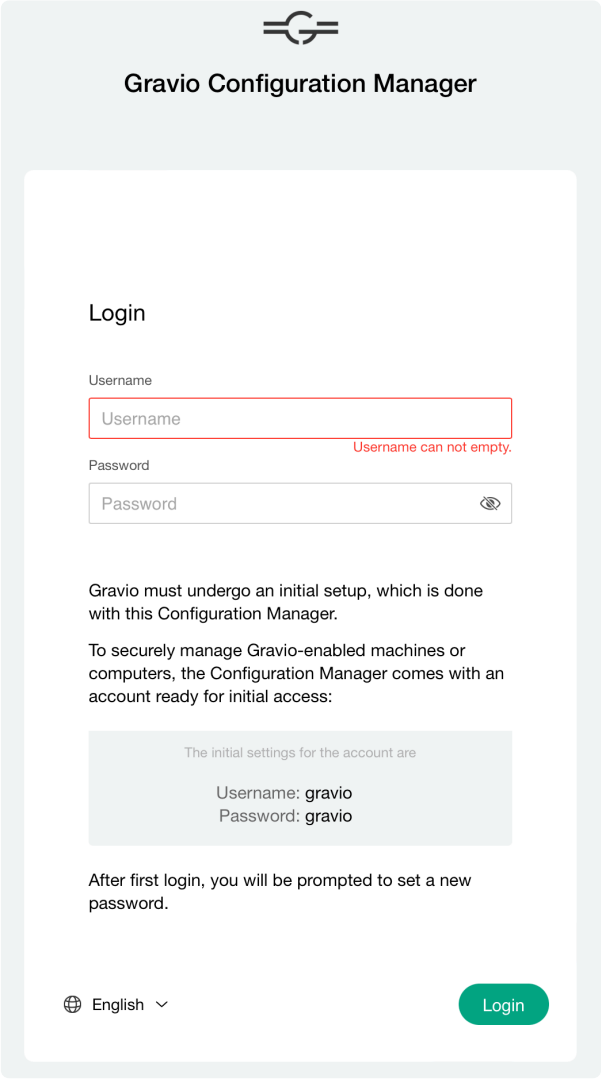
The WiFi password is “graviohub”.



After connecting, open the Gravio Configuration Manager by typing “http://192.168.100.1:8080” in your browser.

Then go to the “Gravio Hub configuration” and proceed to configure the time and date settings, the regional settings, and Gravio Hub Internet settings.

3.1.3. Gravio Hub Settings

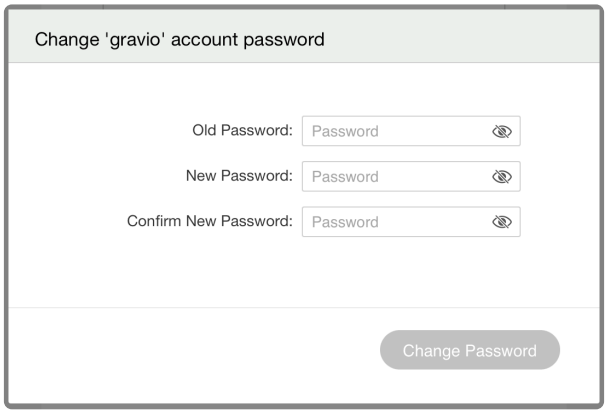


The Gravio Configuration Manager is a web application bundled with HubKit.

Therefore, in order to manage this system securely, the Gravio Configuration Manager has a dedicated account.

Open the Configuration Manager by typing “http://192.168.100.1:8080” in your browser.

Username: “gravio
Password: “gravio

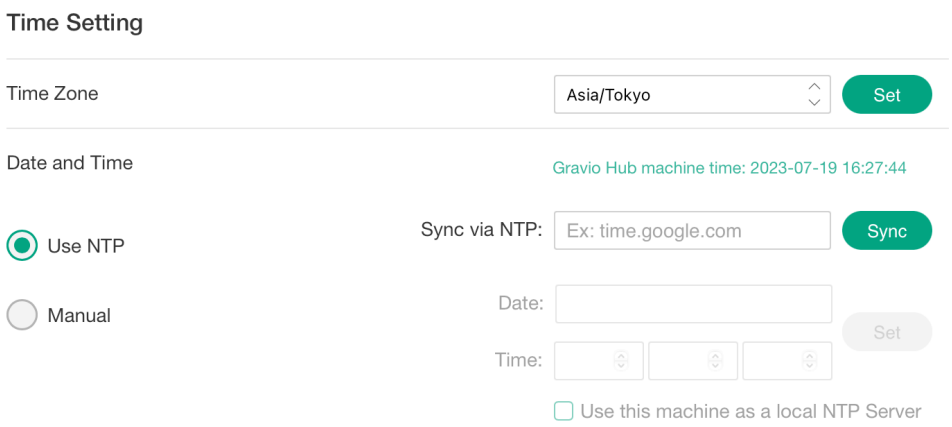


When you log in for the first time, you will see a panel forcing you to change your password. Enter your current and new passwords and press the “Change password” button. The new password will be used from then on.

After logging in, the Gravio Hub Settings screen will open.
This menu will not appear on HubKit Windows/Mac/Linux/RaspberryPI.

time settings

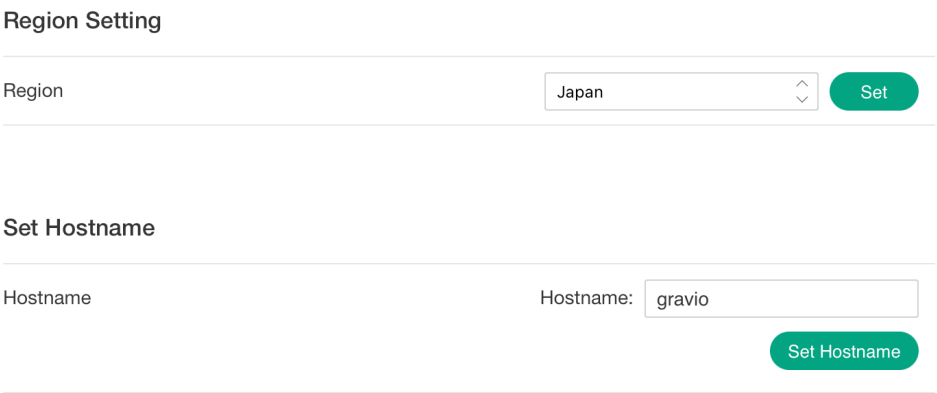
If you are using Gravio Hub on a network connected to the Internet, this time setting is not required, so please skip the time setting procedure below.
If the network on which the Gravio Hub is installed is **not** connected to the Internet, select Manual Settings to set the date and time.



(NOTE : If you are using NTP and only want to change the time zone, please do so after the network configuration is complete.)

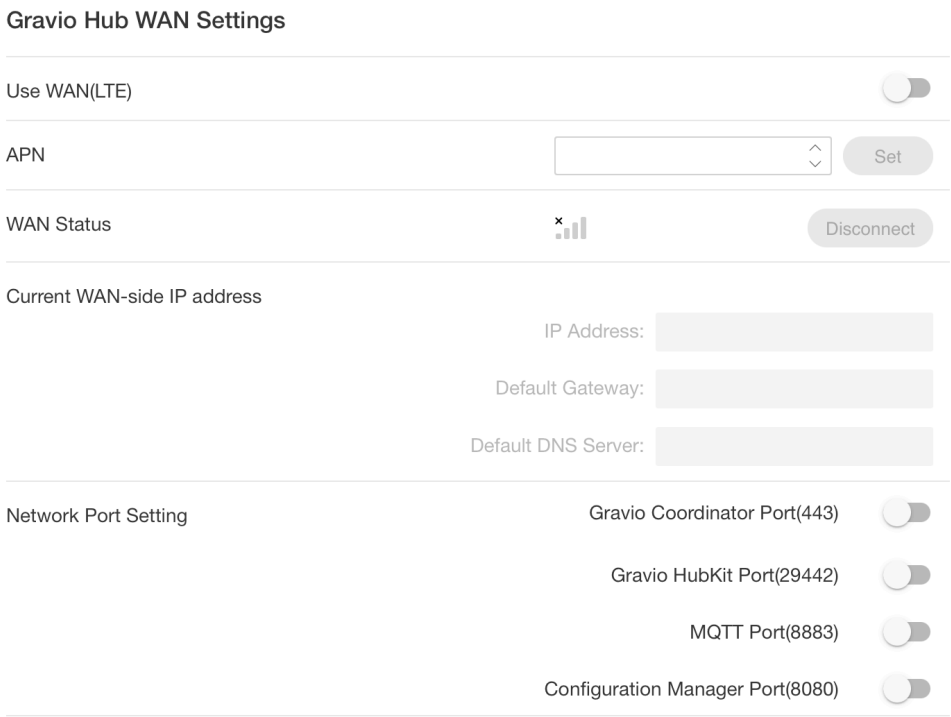
If the Gravio Hub is not connected to the Internet, time synchronization with the outside world will not be performed during manual configuration.

regional settings



Then, select a region and set the hostname in the Host Name setting. The screen will be set to “gravio” so that it can be accessed at “gravio.local”.

WAN settings for Gravio Hub



Gravio Hub2 also supports communication using LTE (LTE mode). You can connect directly to the Internet by inserting a nano SIM of the following carriers into the slot and enabling the WAN settings.

To enable the WAN setting, turn on “Use WAN (LTE)” and reboot.

After rebooting, the Gravio Hub2 network will be “192.168.96.1” with wifi router mode enabled.
The DHCP server will also be enabled, so the IP address assignment by DHCP will also be enabled at “192.168.96.x”.
When you connect to the “GravuiHub-XXXXX” via wifi, an IP address will be assigned automatically.

If you do not use DHCP, please set a fixed IP address at “192.168.96.x/24” for the device connecting to Gravio Hub2.

Then select and set the APN and reboot again.

You will be automatically connected to the WAN after rebooting, but you can also switch manually by pressing the “Disconnect” and “Connect” buttons.
When connected to the WAN, network information will be displayed on the current WAN-side IP address.

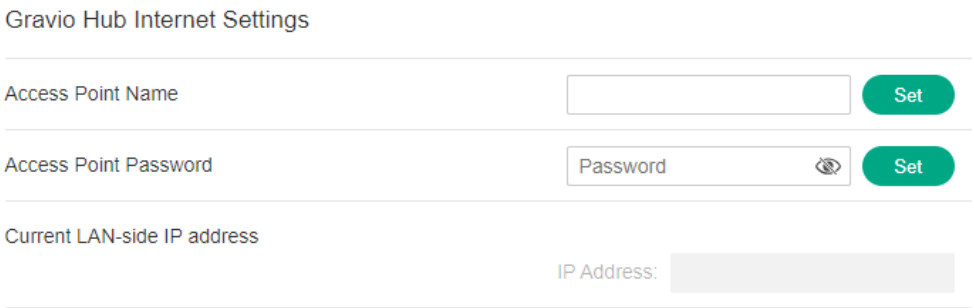
Note: If the Gravio Hub was assigned an IP address as a DHCP client using DHCP in a network environment connected via a wired LAN,
When LTE mode is turned on, the DHCP server on the wired LAN side conflicts with the DHCP server on the Gravio Hub, so please do not connect to the network environment connected by wired LAN.

You can also set the port to allow access to the Gravio Hub from the WAN side in the WAN-side incoming port settings.

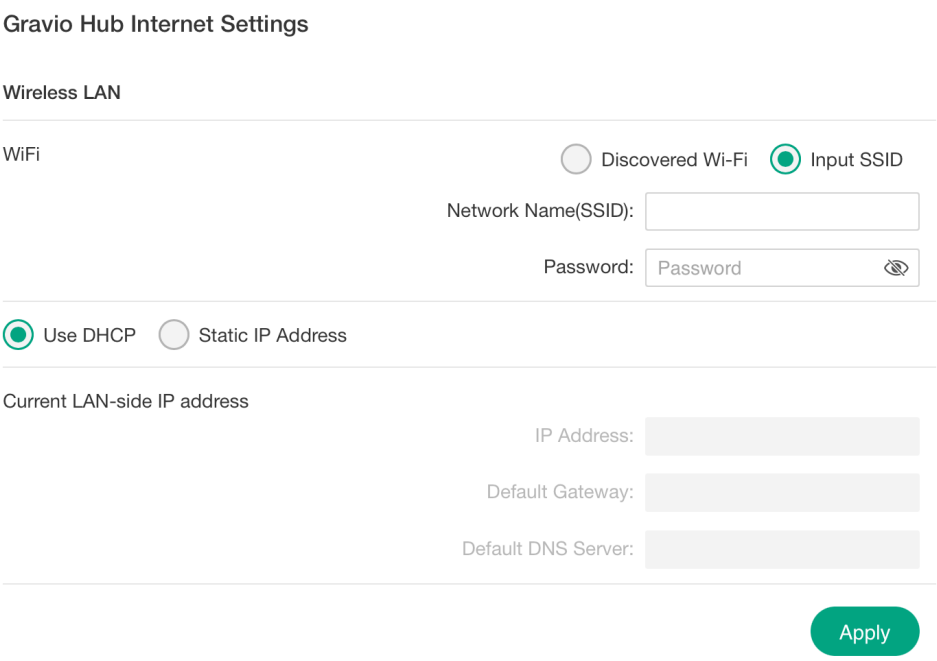
Service Name	Port number
Gravio Coordinator	443
Gravio HubKit Port	29442
MQTT port for node-to-node communication MQTT port for inter-node communication	8883
Configuration Manager port	8080

This completes the WAN configuration.

h3. Internet Settings for Gravio Hub



If LTE mode is on, you can change the WIFI network name (SSID) and WIFI password for the Gravio Hub.
If connected to a LAN, the IP address of the Gravio Hub on the LAN side will be displayed.

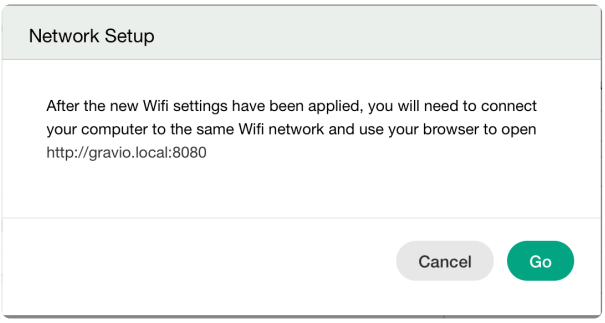


Finally, configure the network to be used. (It is recommended to use a fixed IP address.)
*If a LAN cable is connected to the wired LAN terminal, the settings will be those for wired LAN.
*If a LAN cable is not connected to the wired LAN terminal, wireless LAN settings will be used.
*If a wired LAN cable is connected and a fixed IP is set, the wired LAN will always be used.（Even if a fixed IP is set for WiFi, WiFi will not be available unless the fixed IP setting for the wired LAN is canceled and the DHCP setting is set).

If you wish to use a static IP address, turn off “Use DCHP” and enter the static IP address, default gateway, and DNS server in the Static IP Address field.

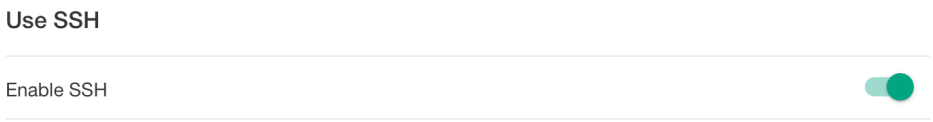
Click the “Configure” button to complete the settings.

Click the “Configure” button.



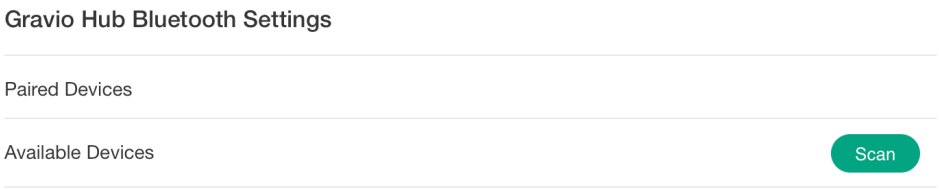
A message dialog box will appear when the wireless and wired LAN settings are complete.

Use of SSH



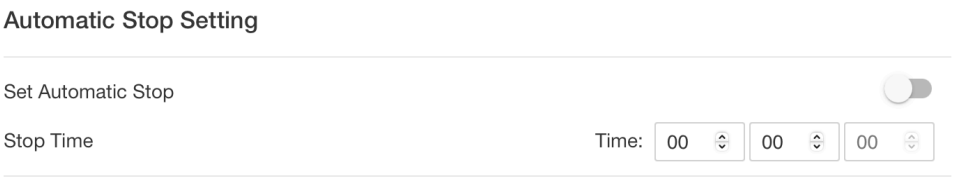
Turn on Use SSH when connecting to Gravio Hub via SSH. (If you turn it on, you need to restart Gravio Hub.)

Bluetooh setting



Connect to the Bluetooth device (e.g. speaker) to be used with the Gravio Hub.
First put the device in pairing mode to pair with a Bluetooth device and press the “Scan” button.
After a short wait, the Bluetooth device will appear in the Available Devices and press the “Pair” button.
After pairing is complete, press the “Connect” button to connect to the device.

Auto stop setting



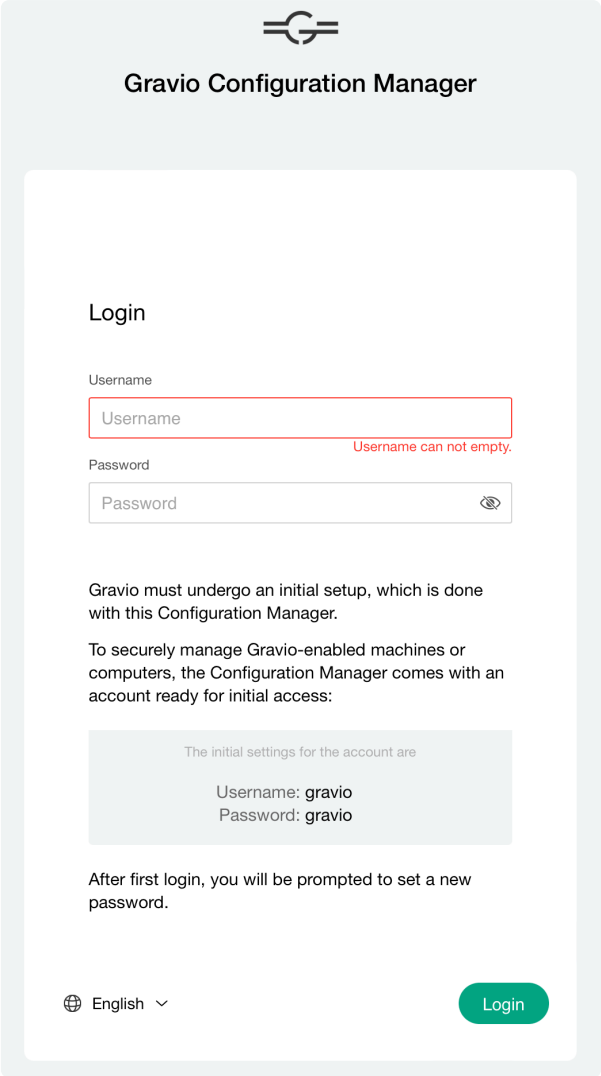
Set the Gravio Hub to automatically shut down and turn off at a set time. Use this setting when the main power is turned off every day in a factory, etc.

If you are using a wireless LAN, set the WiFi connection point again, and then enter “.local” in your browser.
If you are using a wired LAN, enter “.local” in your browser.

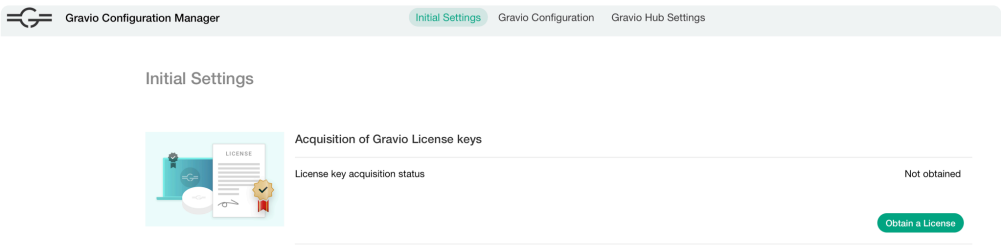
This completes the installation.

3.1.4. Initial Settings (Gravio Configuration Manager)

Open the Configuration Manager using a browser. You will reach the Configuration Manager by entering the URL `http://<IP of your HubKit>:8080` . Note, if you are connecting to your HubKit via the Access Point Mode, the address will be `http://192.1601.100.1:8080`



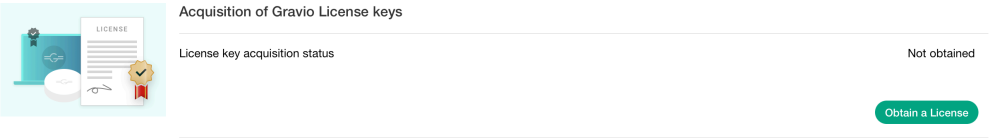
In the browser, enter ".local:8080" configured in the previous chapter.



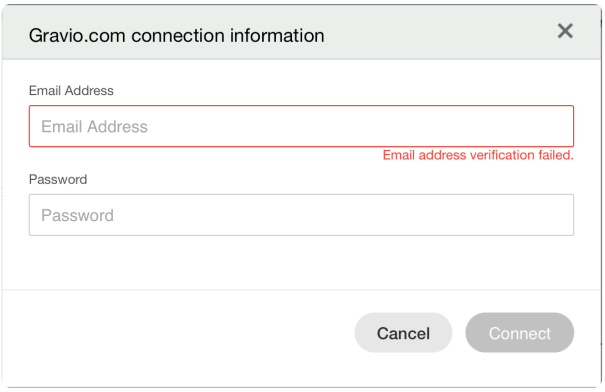
After logging in, the Gravio Hub Settings screen will open and the initial menu will open.

Deploying your License

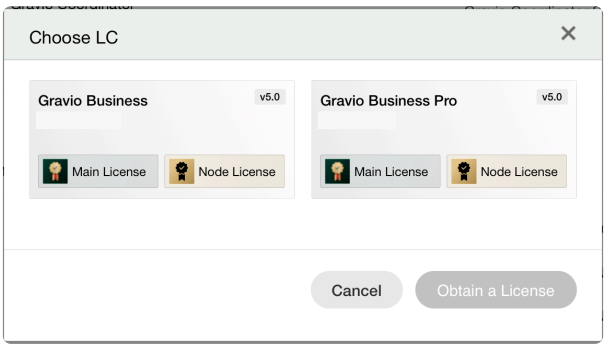
You must first obtain a license to use Gravio. This applies also if you are using the free Gravio version.



Install the license associated with the Gravio user into HubKit.

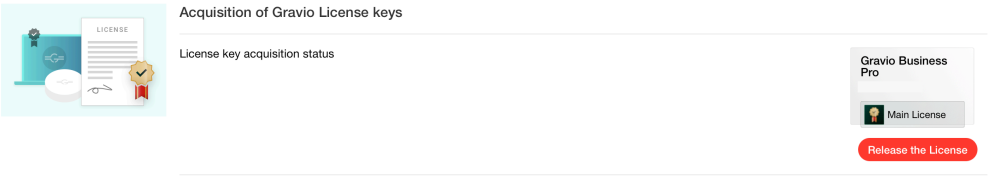


Click the “Obtain License” button to open the connection information panel to Gravio.com and enter your Gravio username and password under which you have registered your online account.



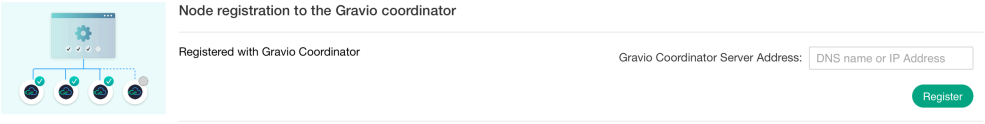
After you have entered your username and password, you will be authenticated and your associated licenses will be displayed. Select the license you like to install on the device you are currently logged in. The license file will be installed onto your device over the internet.

Apart from the different service levels are two types of licenses. “Main Licenses” including the Gravio Coordinator, and “Node Licenses” for HubKits.



The acquired license will be displayed. Select Release License to release it. You might like to do that to install the license on a different device.

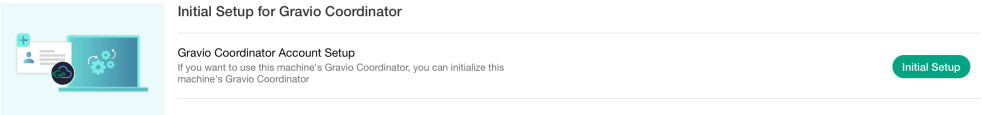
Register your Gravio HubKit Node with the Coordinator



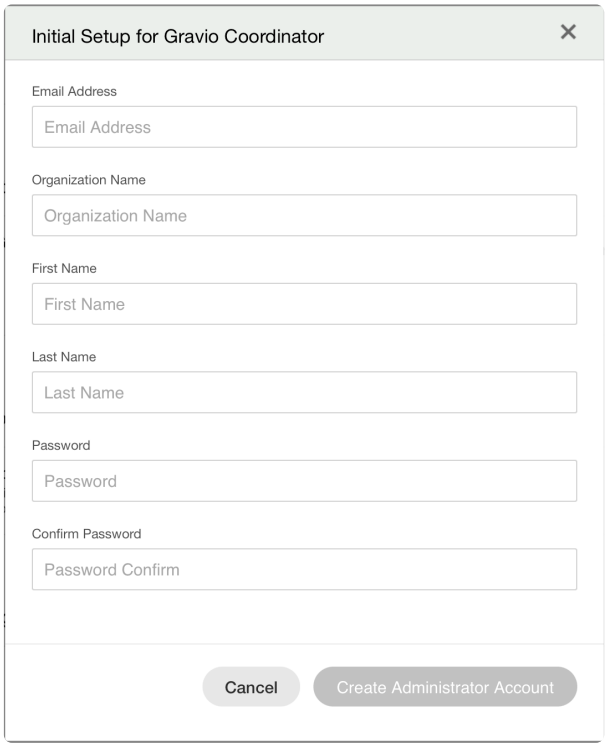
If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

Initialization of the Coordinator



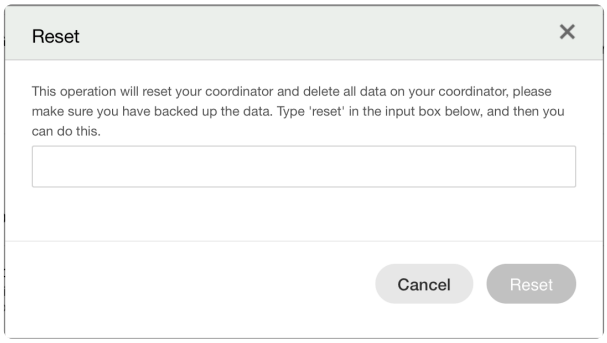
Upon the first launch, you have to create your Gravio Coordinator account. This will be used to manage all your HubKits.



Enter the account information and press the Create Admin Account button to create the Coordinator account.

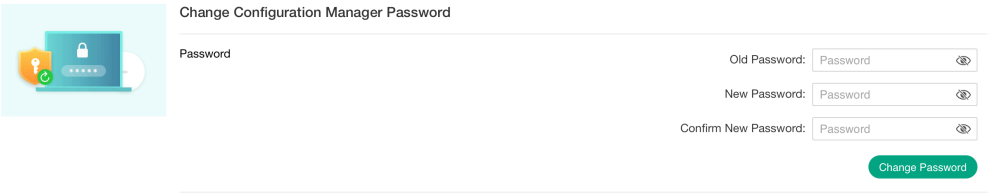
You can use this account to log in to the Coordinator. Please keep this information safe.

To initialize the created account, press the Reset button.



A confirmation panel will appear so you can confirm and initialize.

Change Password for the Gravio Configuration Manager

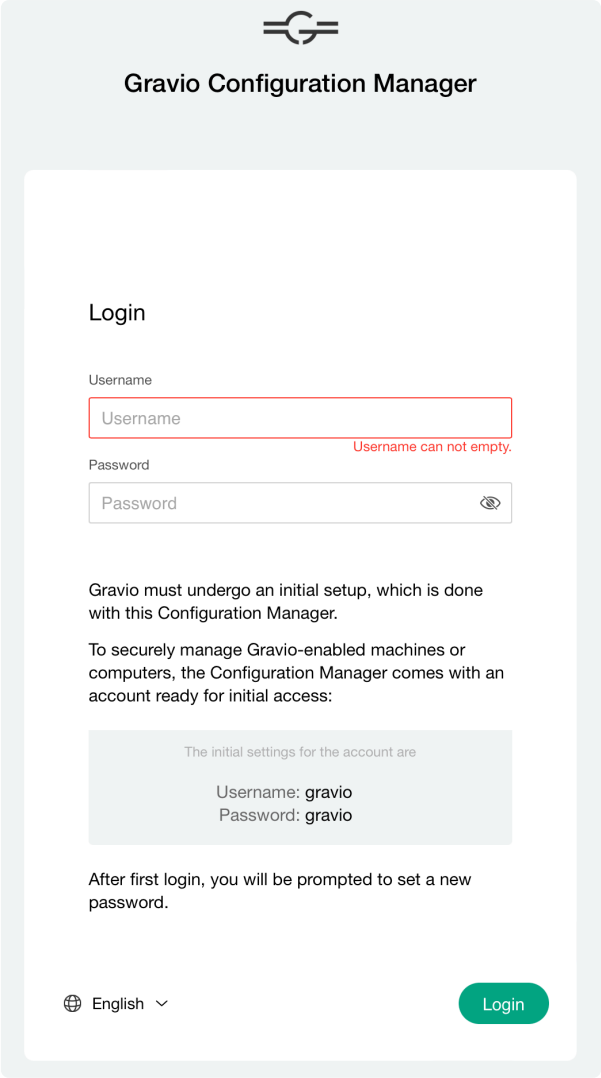


This is used to change the configuration manager password. The change will be effective from the next login.

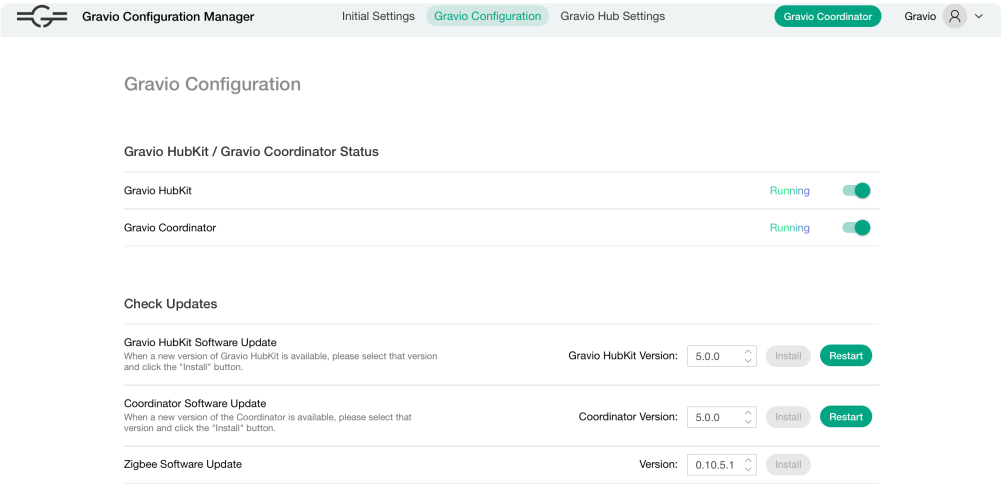
3.1.5. Other Settings

3.1.5.1. Gravio Settings

Open the Configuration Manager using a browser. You will reach the Configuration Manager by entering the URL `http://<IP of your HubKit>:8080` . Note, if you are connecting to your HubKit via the Access Point Mode, the address will be `http://192.1601.100.1:8080`



Or, in the browser, enter “.local:8080” configured in the previous chapter.

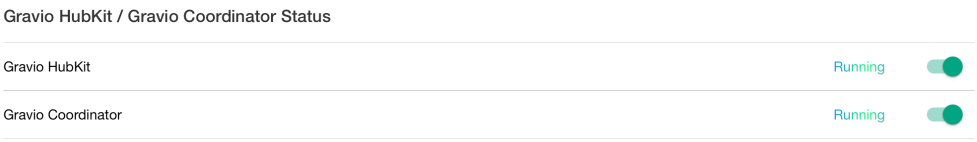


After logging in, the Gravio Hub Settings screen will open, and open the Gravio Settings menu.

Note, the coordinator menu will not appear on HubKit Windows/Mac.

Gravio HubKit / Gravio Coordinator Status

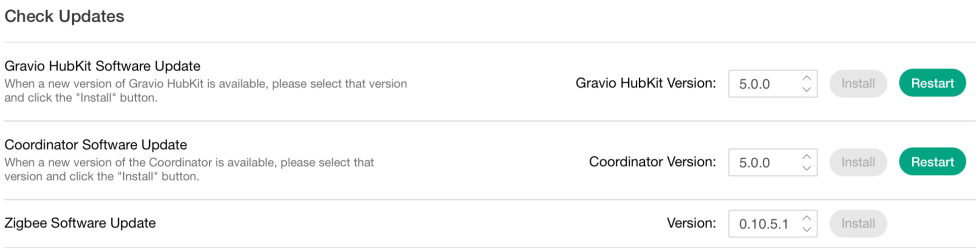
You can check the status of HubKit and Coordinator services.



Use the toggle to start/stop HubKit and Coordinator services respectively.

Software Updates

You can update the software of the coordinator and HubKit over the internet.



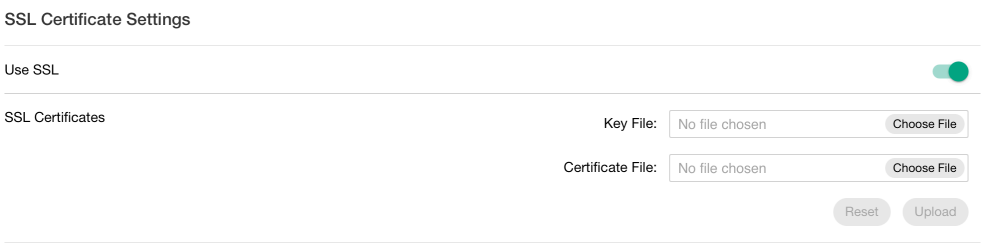
Select the version of the HubKit and Coordinator software to be updated and press the “Install” button. When the installation is complete, click the “Reboot” button.

You can also update the Zigbee firmware. Select the firmware to be updated with and press the “Install” button. Once the installation is complete, you can use the device as is.

h3. Changing the Coordinator's Certificate

The coordinator has a standard self-certificate installed, but you can change it to a certificate for your own domain.

Prepare the key file and the certificate file in advance.



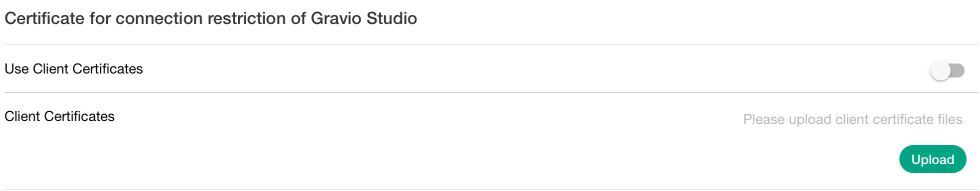
Upload the key file and certificate file to the SSL certificate with "Use SSL" turned on in the certificate settings.

When the upload is complete, the certificate will be renewed.

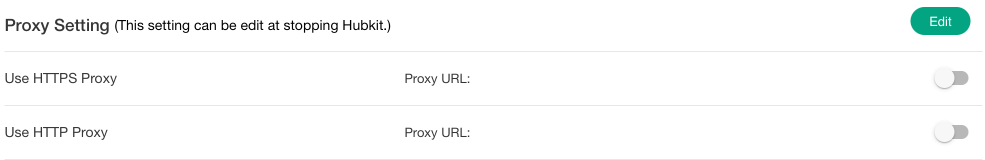
Manage Gravio Studio Connections Using Client Certificates

Gravio communicates securely between Gravio Studio, the HubKits and the Coordinator using certificates. To set up a client certificate for Gravio Studio to connect to HubKit.

Prepare the certificate files in advance, and upload them here.



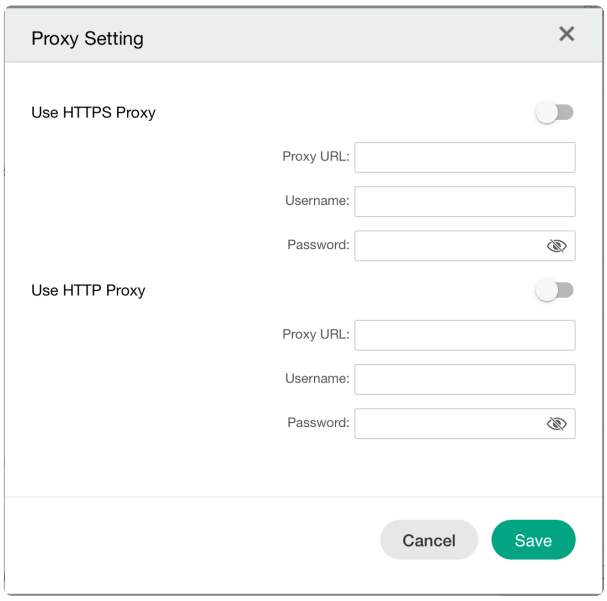
Proxy Settings



Set these credentials if HubKit needs to go through a proxy server when using Internet services.

Click the "Edit" button to display the configuration screen. Enter the account information (proxy URL,

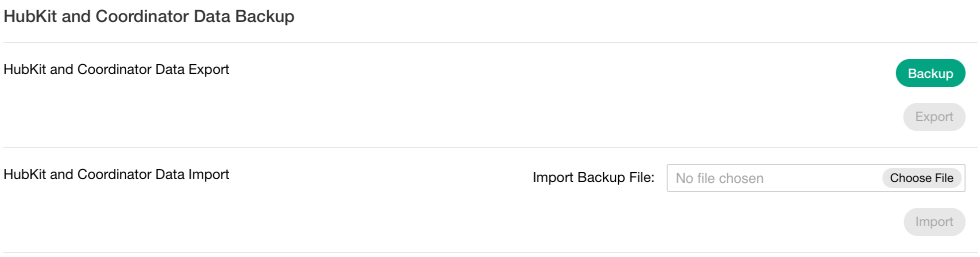
account, and password) for the proxy server to be used for HTTPS and HTTP protocols, respectively.



Note that HubKit must be stopped during this update.

HubKit / Coordinator Data Backup

HubKit / Coordinator data can be backed up to a local PC and restored.



To backup HubKit / Coordinator data, press the Backup button. After a few moments, the backup will be completed and you can download the backup file to your local PC by pressing the “Export” button.

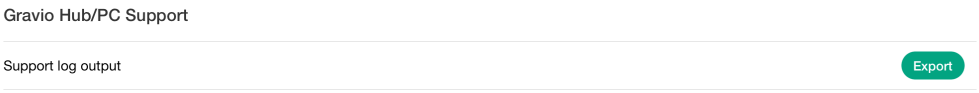
To restore the backed-up data, press the “Select File” button in the Import File section and select the file. After a few moments of pressing the Import button, the file will be uploaded and processed.

Notes:

1. Please ensure that the file you are importing was previously backed up by the Gravio Configuration Manager.
2. Please ensure that the Zigbee firmware version of the Gravio Hub is the same for the backup source and the restore destination.

Gravio HubKit Support Log

You can download the HubKit and syslog log files in a zip file.



Press the “Export” button to download the log files.

Gravio Hub Power Management

You can power off the Gravio Hub here.

Press the “Power Off” button to turn off the Gravio Hub.
Pressing the “Reboot” button will reboot the Gravio Hub.
Pressing the “Initialize” button will return the Gravio Hub to its factory settings and delete all configuration information and data. Please use this function carefully.

3.1.5.2. Performance Tuning

When using the Gravio Hub for inference computer vision detection, please consider the following:

Processing Performance

- When an inference is selected as a “layer as a soft sensor” on a Gravio Hub, only one model should be used per Gravio Hub.
- Depending on the complexity of the model file, the load on memory and CPU may vary.
- For a typical TensorFlow model, we recommend that you use no more than two cameras connected and that you run the model at intervals of about one minute.
- Make sure to run the camera images at least 15-20 seconds apart.
- In the case of the Lite TensorFlow model, it is recommended that no more than two cameras are connected at any given time, and that the recognition runs at intervals of 30 seconds.
- To keep the Gravio Hub at optimum performance, log into the Gravio Hub via SSH and use the `top` command to see the average load. For ideal performance, please keep the load average below 2.

If you plan to operate the HubKit at a rate exceeding this recommendation, please install the HubKit on a dedicated PC with higher processing power.

Disk Space

- The disk capacity of Gravio Hub is 16GB, including the OS.
- The available space for operation is approximately 6GB.
- If you have enabled image output during inference, please be careful not to exceed this capacity.

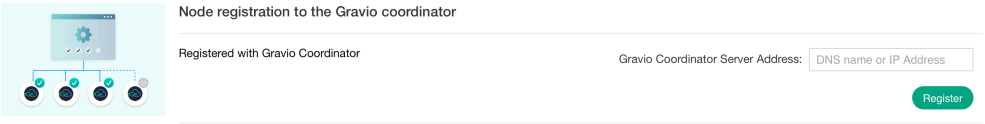
Coral.ai Support

- The Gravio Hub supports the Coral USB Accelerator by Google if you require extra power for your Gravio Hub. (Not included in the package).

3.2. Installation and initial setup of the Windows version

HubKit for Windows can be operated as a Gravio HubKit Node.

- Follow the instructions from the windows installation manual . Note, if you already have Gravio Studio, and just need to install the HubKit software, you can download this from https://www.gravio.com/en/download
- Open the Initial Setup screen to deploy your Gravio license.
- Register your Gravio HubKit Node with the Coordinator



If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

This completes the setup of the operating environment.

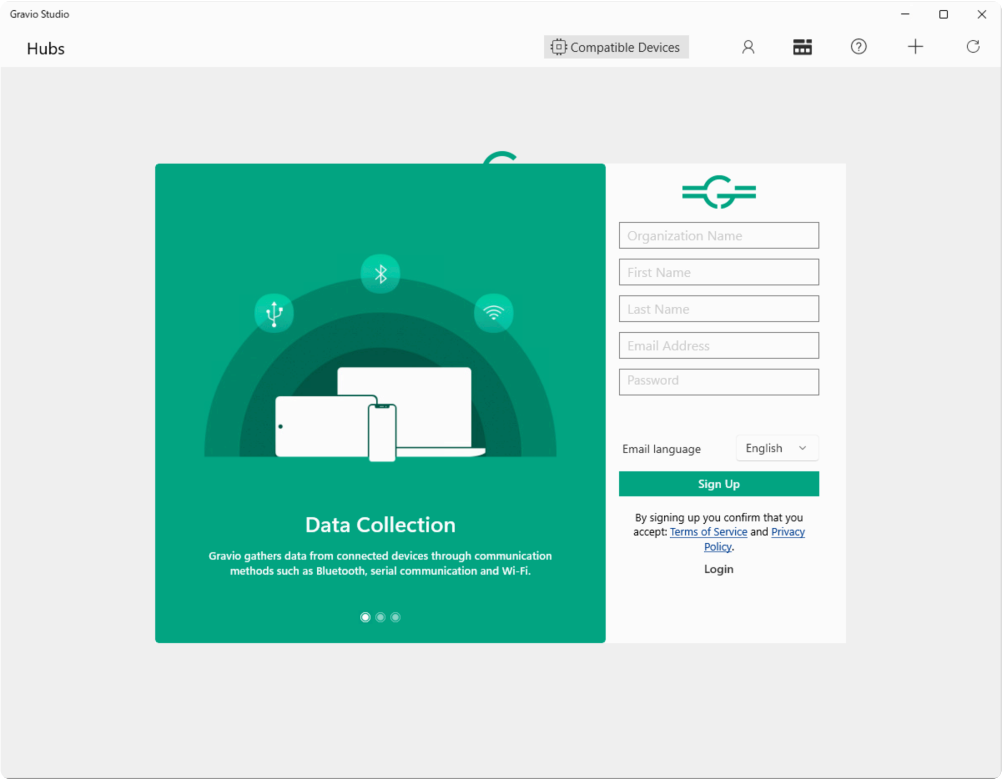
3.2.1. Installing the Windows version

Using Gravio with Windows

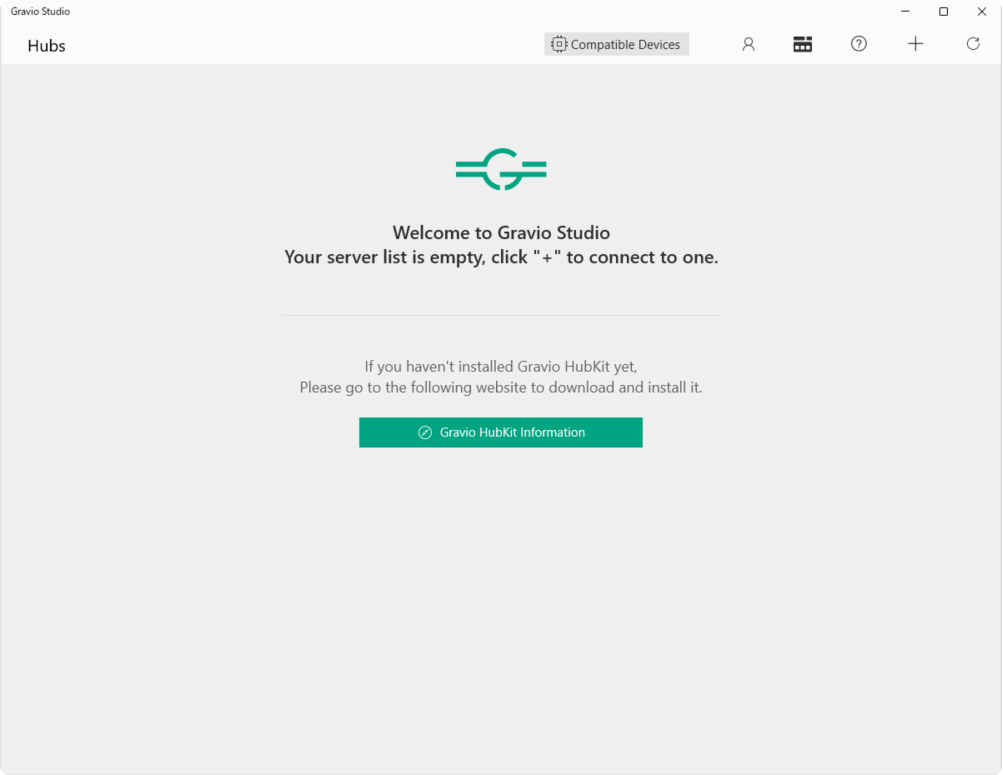
To use Gravio, you will need to install two pieces of software.
Gravio Studio for Windows, the software to manage your infrastructure. You can get Gravio Studio from the App Store. Secondly, you will need Gravio HubKit, the actual node server software that runs as a daemon on your machine.

Installing Gravio Studio for Windows and HubKit

- Install Gravio Studio from the Microsoft Store.
- Start Gravio Studio.
- If you do not already have an account, fill out this form to agree to the Terms of Use and create an account.



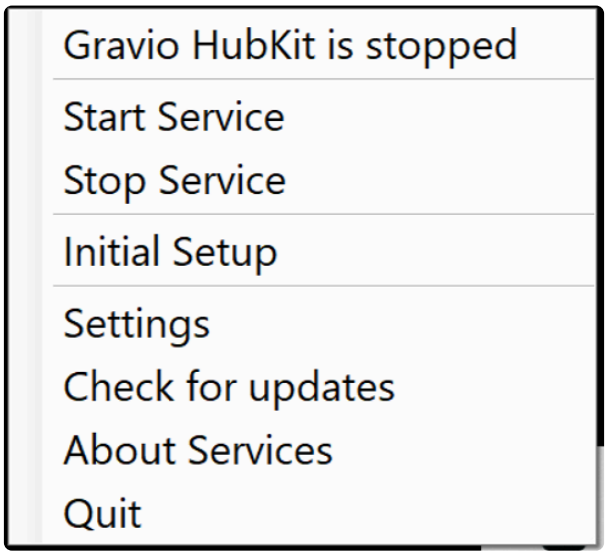
- If you already have an account, please login.
- If you have signed up, you will receive an authorization code by e-mail. Check your e-mail. When you log in, you will be asked for the verification code, please enter it together with the verification code described in the email.
- Initially, if you have not yet installed the Gravio HubKit server, the server list will be empty.



However, if you have Gravio HubKit instances running in the same network they will appear here.

- To install HubKit locally, click the “Download HubKit” button.
- A browser will open and ask for permission to download. Download “GravioHubKit4.msi” and install it.
- After installation is complete, HubKit will start automatically, so start GravioTools from the application.

- When GravioTools starts, GravioTools will be resident in the task tray, so right-click to display the context menu.



Important: After the installation, be sure to select Gravio “Initial Settings” to install the license and perform other initial settings.

The Gravio Configuration Manager is a web application that resides locally on the PC on which the HubKit is installed.
Therefore, a dedicated account related to this node needs to be created for the Gravio Configuration Manager to manage this system. This is due to Gravio's Edge computing aspect.

See here for instructions on configuring Gravio's initial settings.

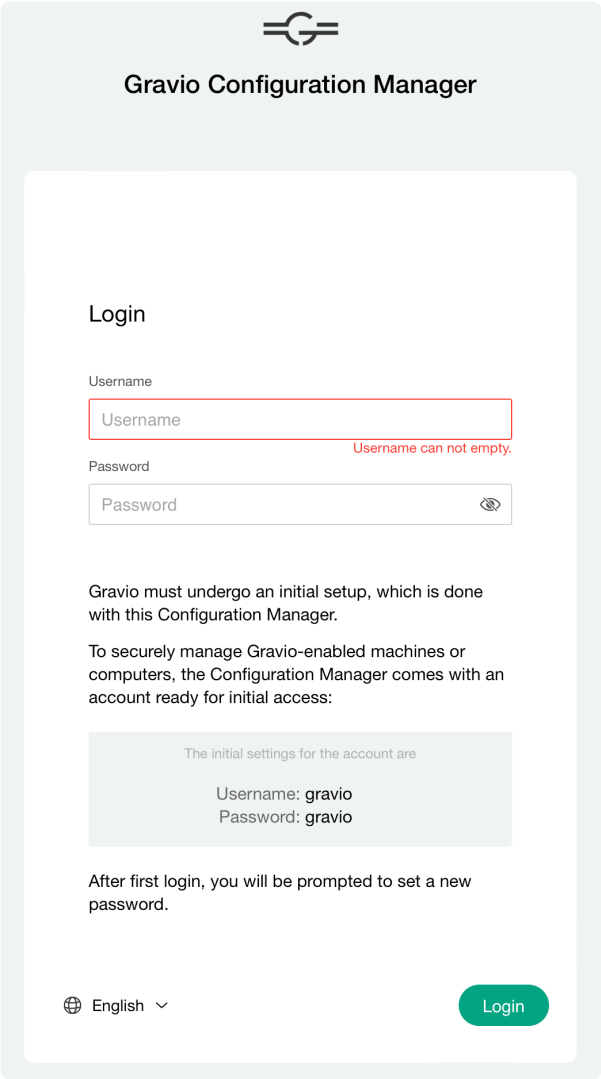
- Once you have completed the HubKit installation, launch Gravio Studio again; you should see the local server on the Hub list.

Note: If your servers are on different network segments, Gravio will not be able to retrieve them automatically. In this case, use the "+" symbol to manually add a new server using its IP address.

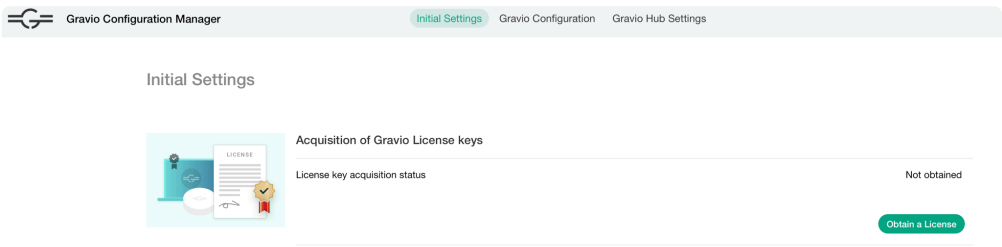
*This completes the installation of Gravio Studio and HubKit.

3.2.2. Initial Settings (Gravio Configuration Manager)

Open the Gravio Configuration Manager screen in your browser using Gravio Tools from Windows Tray. Pick the “Initial Setup” menu option.



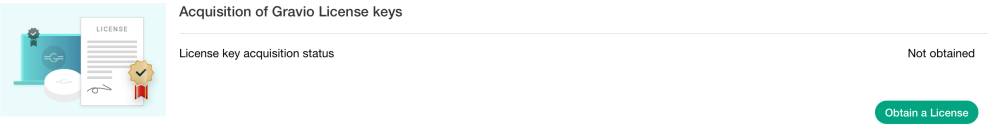
Or, in the browser, enter `<Hostname>.local:8080` configured in the previous chapter.



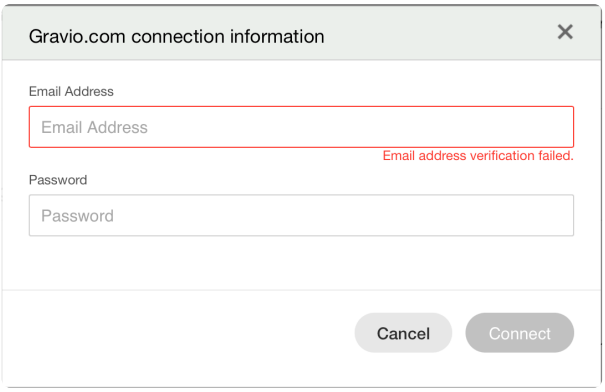
After logging in, the Gravio Hub Settings screen will open and the initial menu will open.

Deploying your License

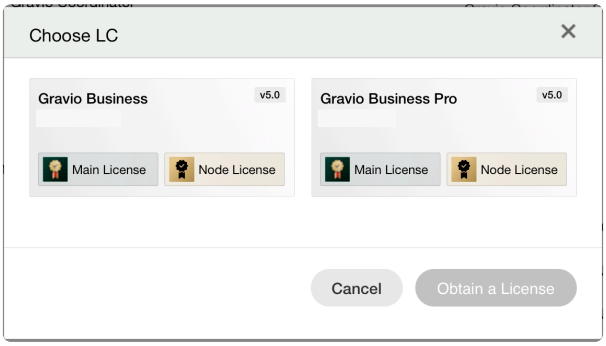
You must first obtain a license to use Gravio. This applies also if you are using the free Gravio version.



Install the license associated with the Gravio user into HubKit.

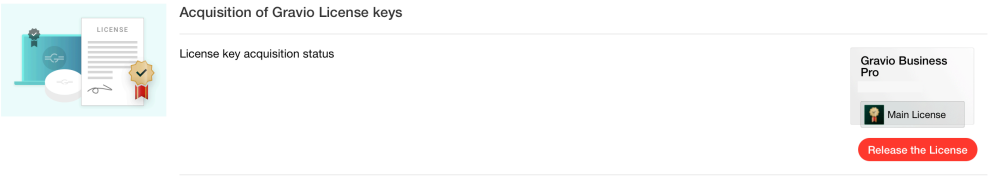


Click the “Obtain License” button to open the connection information panel to Gravio.com and enter your Gravio username and password under which you have registered your online account.



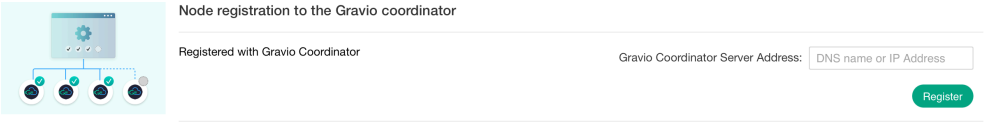
After you have entered your username and password, you will be authenticated and your associated licenses will be displayed. Select the license you like to install on the device you are currently logged in. The license file will be installed onto your device over the internet.

Apart from the different service levels are two types of licenses. “Main Licenses” including the Gravio Coordinator, and “Node Licenses” for HubKits.



The acquired license will be displayed. Select Release License to release it. You might like to do that to install the license on a different device.

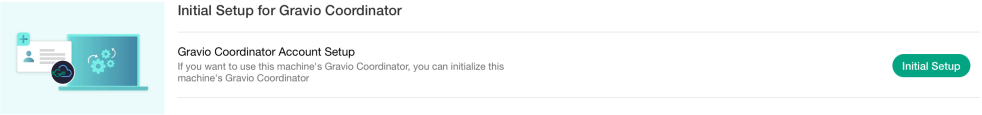
Register your Gravio HubKit Node with the Coordinator



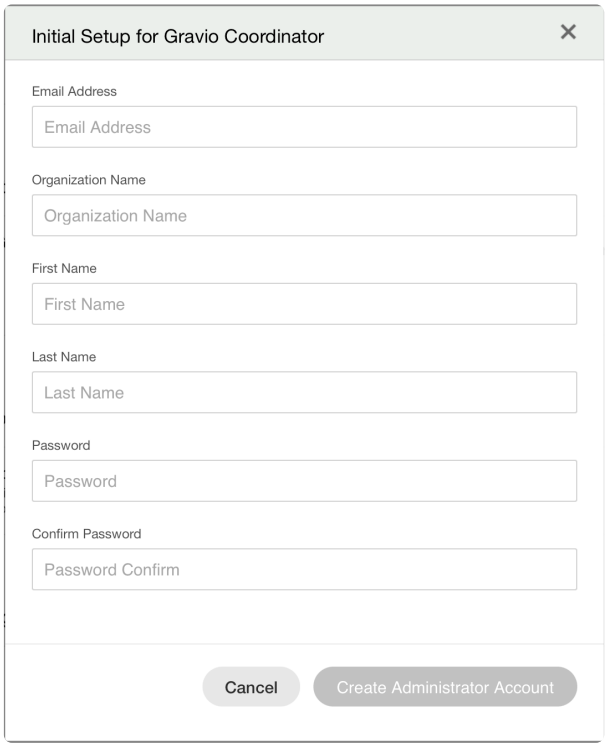
If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

Initialization of the Coordinator



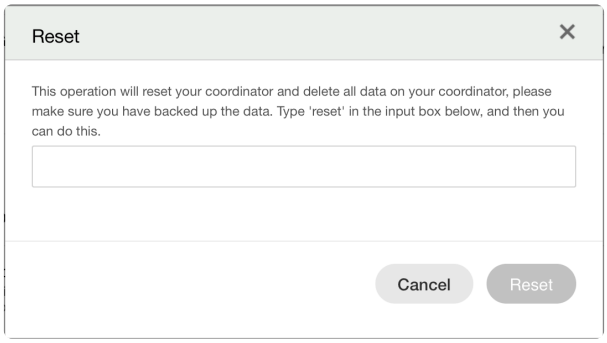
Upon the first launch, you have to create your Gravio Coordinator account. This will be used to manage all your HubKits.



Enter the account information and press the Create Admin Account button to create the Coordinator account.

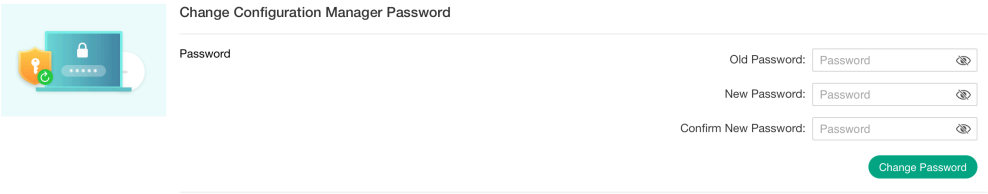
You can use this account to log in to the Coordinator. Please keep this information safe.

To initialize the created account, press the Reset button.



A confirmation panel will appear so you can confirm and initialize.

Change Password for the Gravio Configuration Manager

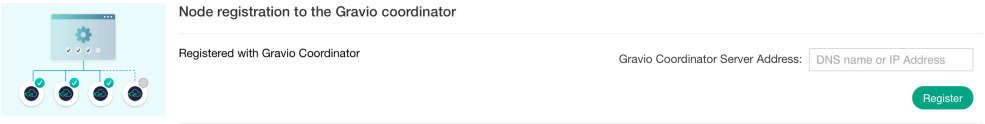


This is used to change the configuration manager password. The change will be effective from the next login.

3.3. Installation and initial setup for macOS version

HubKit Mac version can be operated as a hub in your Node Computing network.

- Follow the instructions from the mac installation manual . Note, if you already have Gravio Studio, and just need to install the HubKit software, you can download this from https://www.gravio.com/en/download
- Open the Initial Setup screen to deploy your Gravio license.
- Register your Gravio HubKit Node with the Coordinator



If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

This completes the setup of the operating environment.

3.3.1. Installing the macOS version

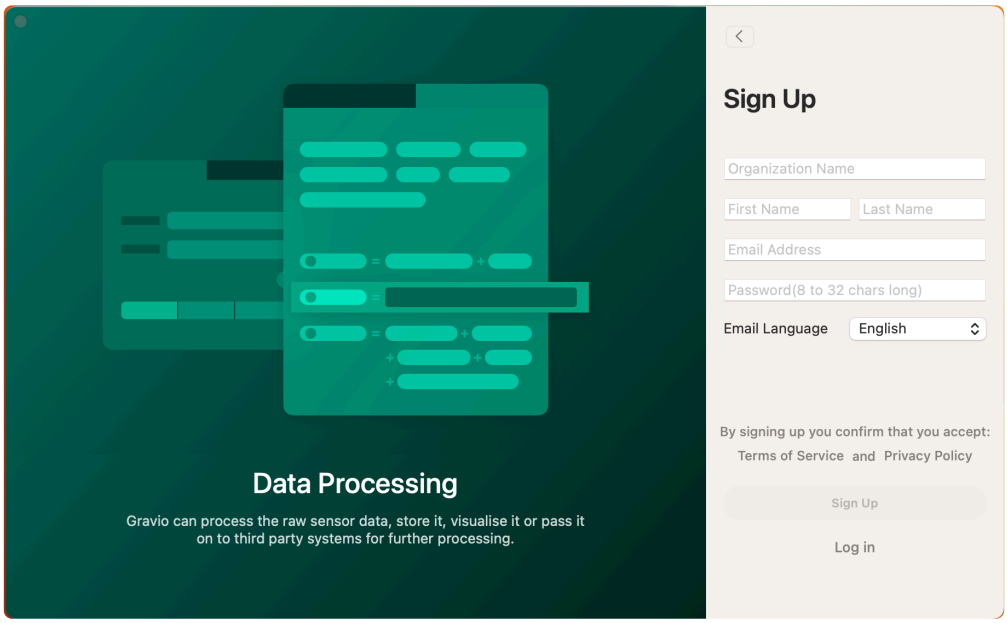
Using Gravio with a Mac

Two software installations are required to use Gravio.

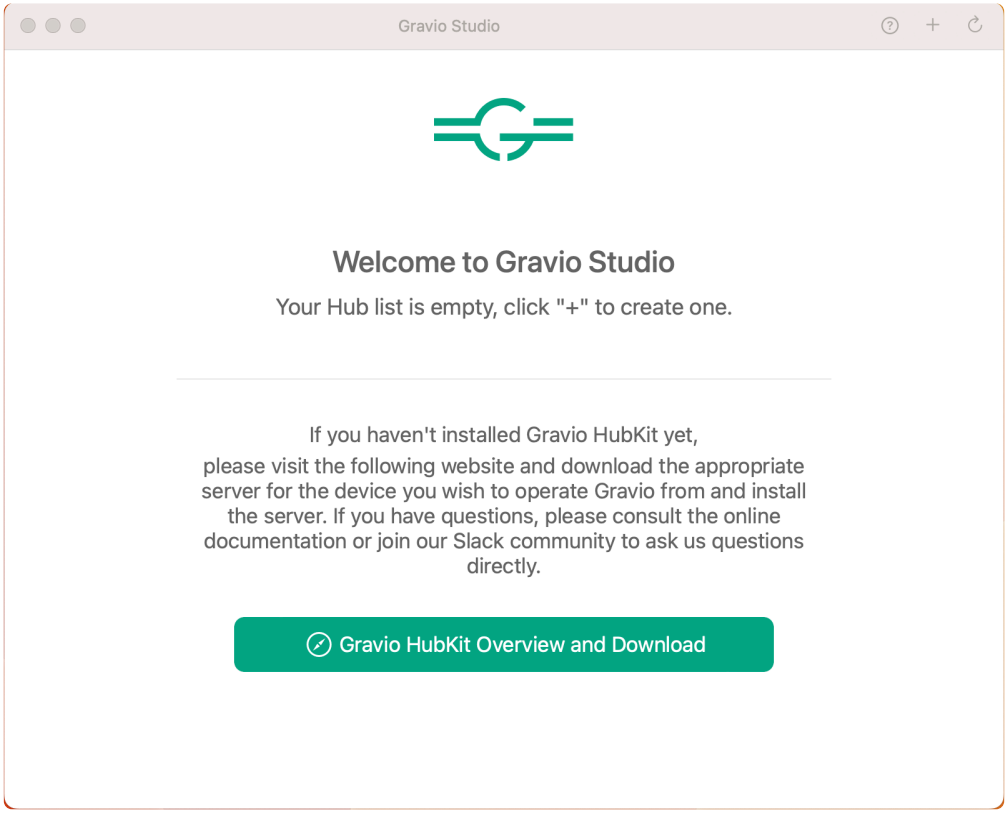
Gravio Studio, which you can get from link.gravio.com/studio for Mac, and HubKit, which you can get from https://www.gravio.com/en/download.

Installing Gravio Studio for Mac and HubKit

- Install Gravio Studio from the App Store: https://link.gravio.com/studio
- Launch Gravio Studio.
- If you do not already have an account, fill out this form to agree to the Terms of Service and create an account.

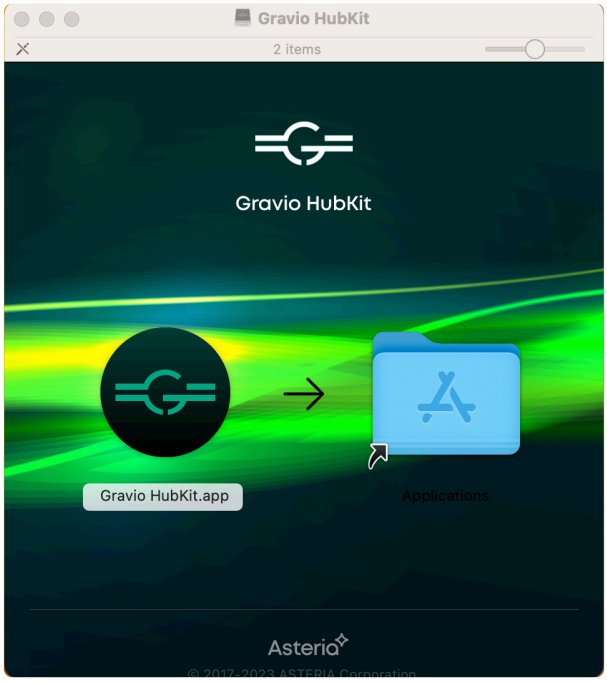


- If you already have an account, please login.
- If you signed up, you will receive an authorization code by e-mail. Check your e-mail. When you log in, you will be asked for the verification code, please enter it together with the verification code described in the email.
- Initially, if you have not yet installed the Gravio server, the server list is empty.

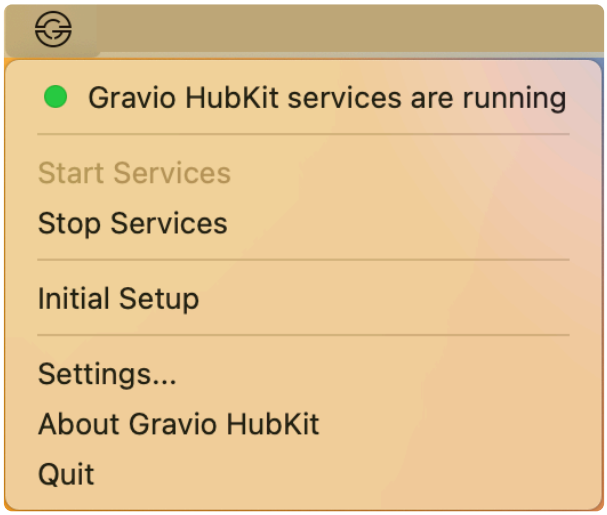


- To install the HubKit locally, click the “About Gravio HubKit” button.
- A browser will open, click on the “Products” menu and click on the Download Gravio HubKit for Mac button at the bottom of the page that appears.

- A screen will open asking for permission to download. Download “GravioHubKit4.dmg” and double-click it to open and install.



- After installation is complete, double-click Gravio HubKit4 in the Applications folder to launch it. You will be prompted to enter the password for administrative privileges.
- When HubKit starts, an icon will appear in the Mac upper menu.



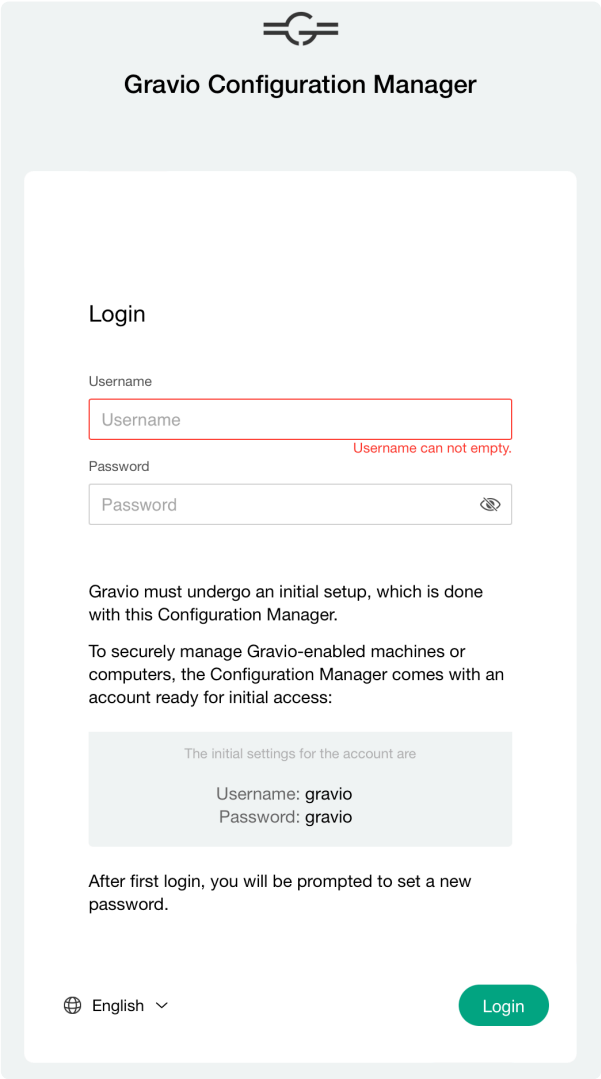
- Once you have completed the HubKit installation, launch Gravio Studio again; you will see your local server on the Hub list.
  Select Gravio Initial Setup and perform the initial configuration, including license installation, etc.
  For more information on Gravio Initial Setup, see here.

Note: If your servers are on different network segments, Gravio will not be able to retrieve them automatically. In this case, use the “+” symbol to manually add a new server using its IP address.

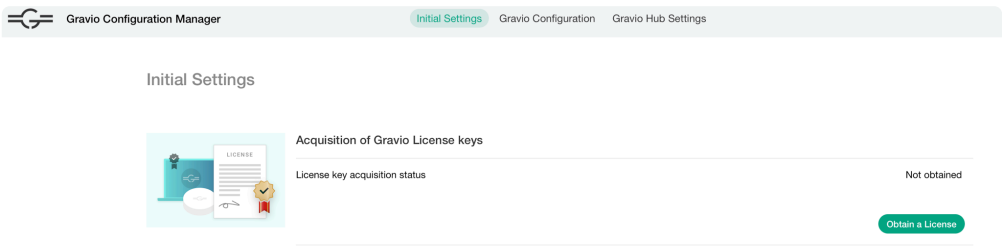
- This completes the installation of Gravio Studio and HubKit.

3.3.2. Initial Settings (Gravio Configuration Manager)

Open the Gravio Configuration Manager screen in your browser using Gravio Tools from Windows Tray. Pick the “Initial Setup” menu option.



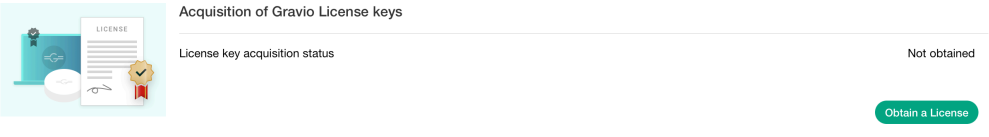
Or, in the browser, enter `<Hostname>.local:8080` configured in the previous chapter.



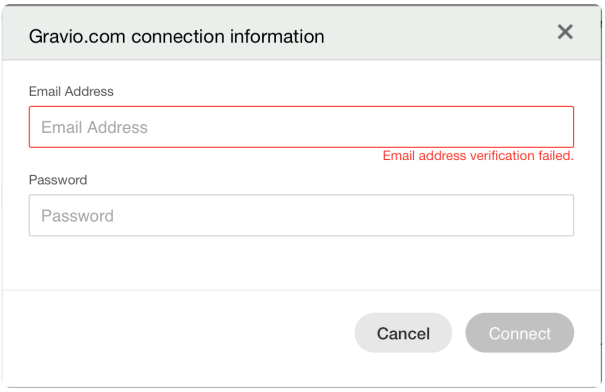
After logging in, the Gravio Hub Settings screen will open and the initial menu will open.

Deploying your License

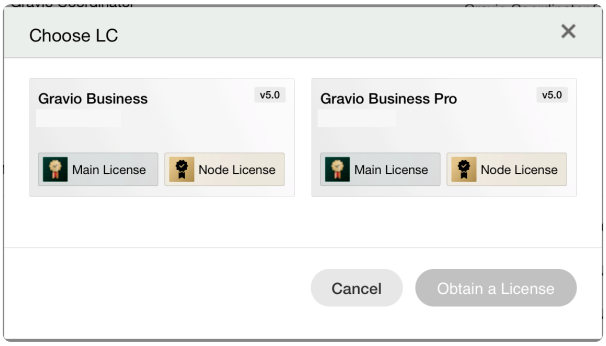
You must first obtain a license to use Gravio. This applies also if you are using the free Gravio version.



Install the license associated with the Gravio user into HubKit.

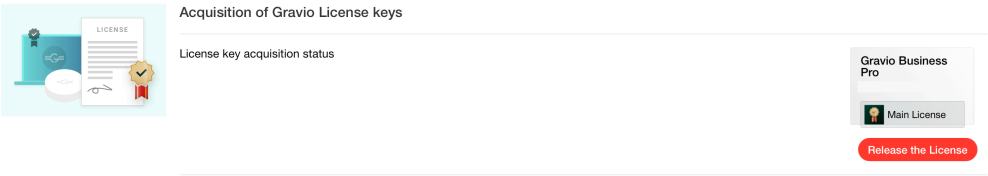


Click the “Obtain License” button to open the connection information panel to Gravio.com and enter your Gravio username and password under which you have registered your online account.



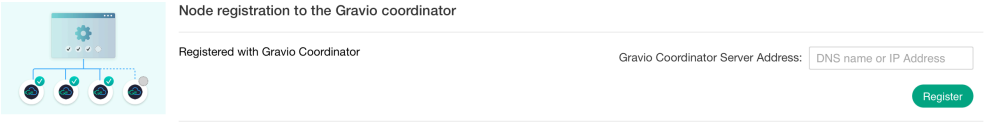
After you have entered your username and password, you will be authenticated and your associated licenses will be displayed. Select the license you like to install on the device you are currently logged in. The license file will be installed onto your device over the internet.

Apart from the different service levels are two types of licenses. “Main Licenses” including the Gravio Coordinator, and “Node Licenses” for HubKits.



The acquired license will be displayed. Select Release License to release it. You might like to do that to install the license on a different device.

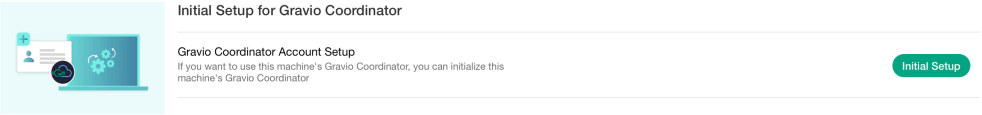
Register your Gravio HubKit Node with the Coordinator



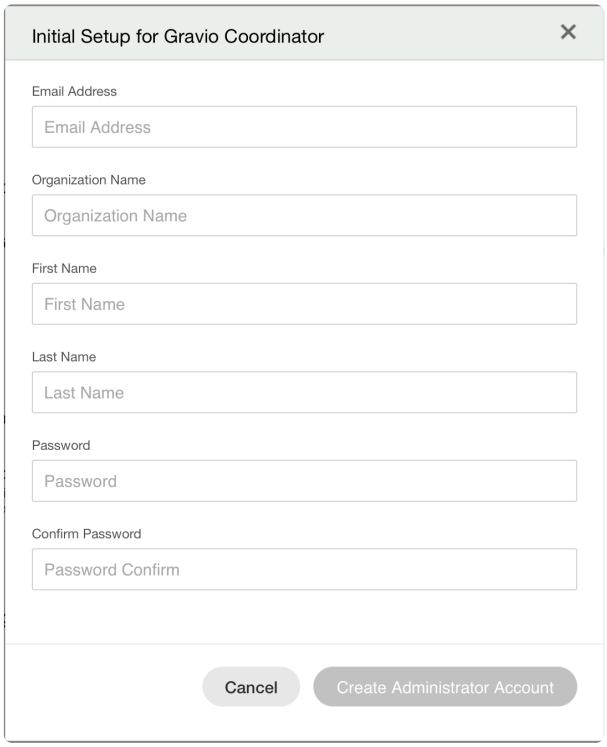
If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

Initialization of the Coordinator



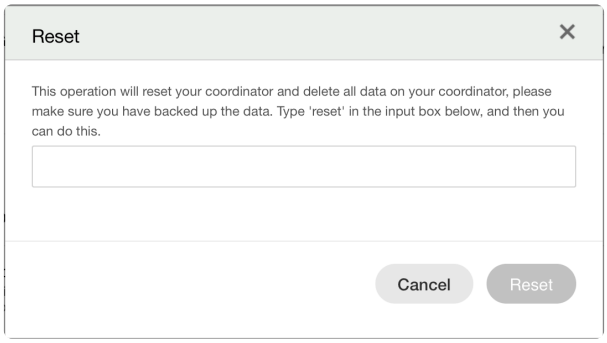
Upon the first launch, you have to create your Gravio Coordinator account. This will be used to manage all your HubKits.



Enter the account information and press the Create Admin Account button to create the Coordinator account.

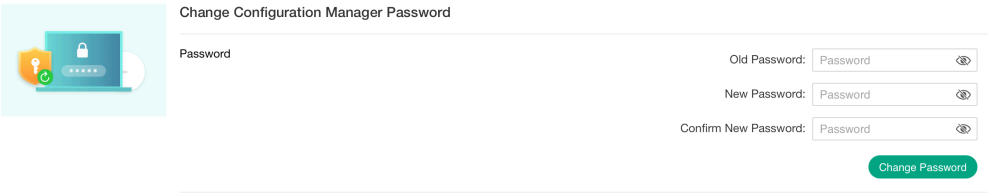
You can use this account to log in to the Coordinator. Please keep this information safe.

To initialize the created account, press the Reset button.



A confirmation panel will appear so you can confirm and initialize.

Change Password for the Gravio Configuration Manager



This is used to change the configuration manager password. The change will be effective from the next login.

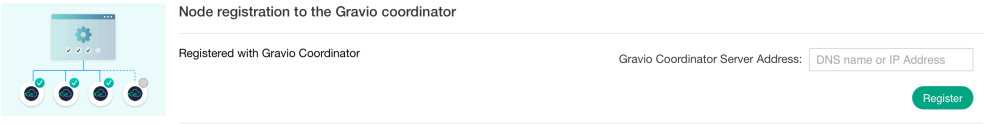
3.4. Installation and initial configuration of Linux (Ubuntu AMD64) version

HubKit for Linux can be operated as a Gravio HubKit Node.

- Follow the instructions from the Linux installation manual. Note, if you already have Gravio Studio, and just need to install the HubKit software, you can download this from https://www.gravio.com/en/download

- Open the Initial Setup screen to deploy your Gravio license.

- Register your Gravio HubKit Node with the Coordinator



If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

This completes the setup of the operating environment.

3.4.1. Installing Ubuntu (AMD64) version

Download the installer for your operating system from https://www.gravio.com/en/download.

- Connect to Ubuntu 20.04 Server LTS using SSH. (Desktop and Core are not supported, use the Server version)
- Transfer the downloaded deb package (setup_-XXXX_amd64.deb for AMD64) to Ubuntu.
- Execute the following Command from Console.

```
sudo apt update
sudo apt upgrade
```

Next, install the deb package that you downloaded.

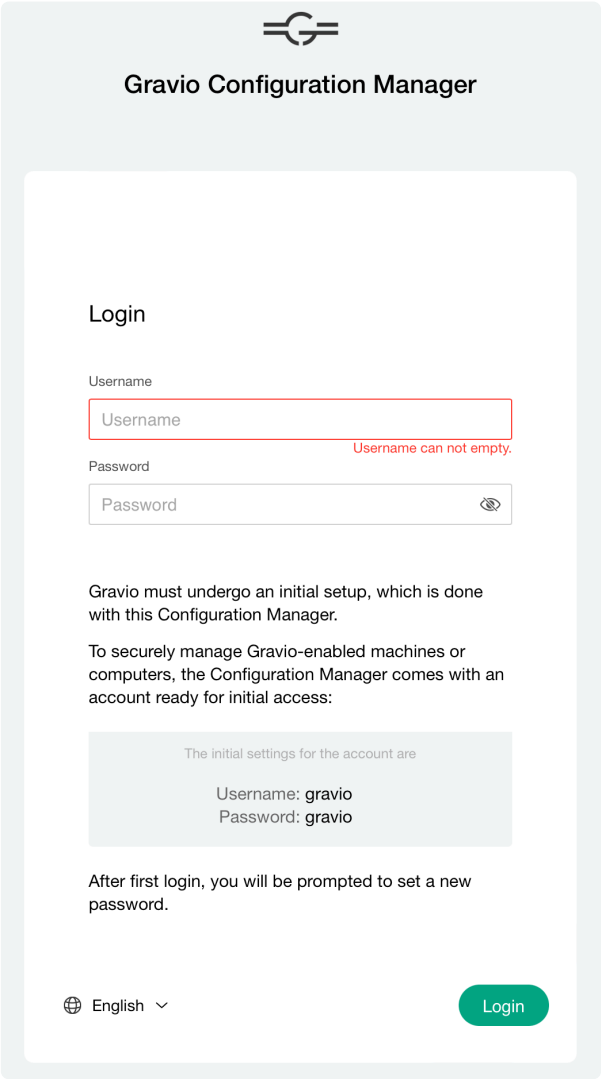
```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

After installing this deb package, the following Linux account will be created

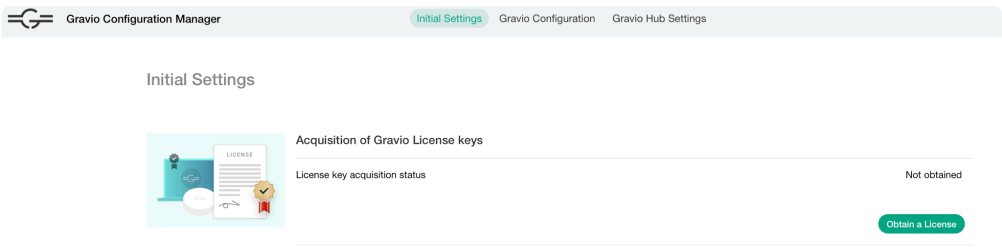
username : gravio
password : gravio

3.4.2. Initial Settings (Gravio Configuration Manager)

Open the Gravio Configuration Manager screen in your browser using Gravio Tools from Windows Tray. Pick the “Initial Setup” menu option.



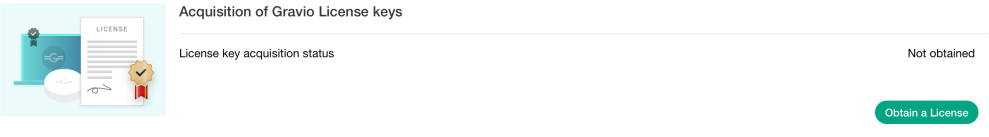
Or, in the browser, enter `<Hostname>.local:8080` configured in the previous chapter.



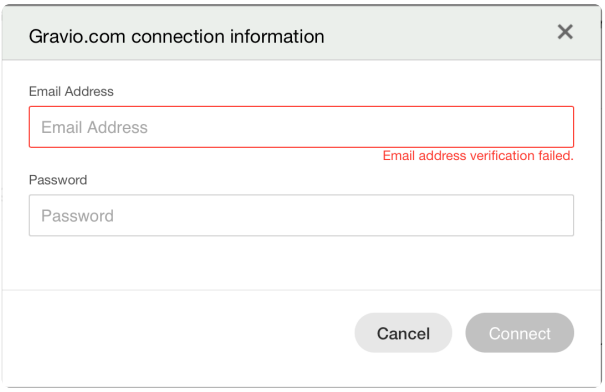
After logging in, the Gravio Hub Settings screen will open and the initial menu will open.

Deploying your License

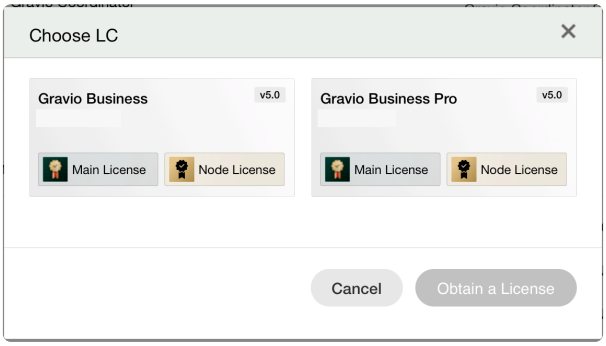
You must first obtain a license to use Gravio. This applies also if you are using the free Gravio version.



Install the license associated with the Gravio user into HubKit.

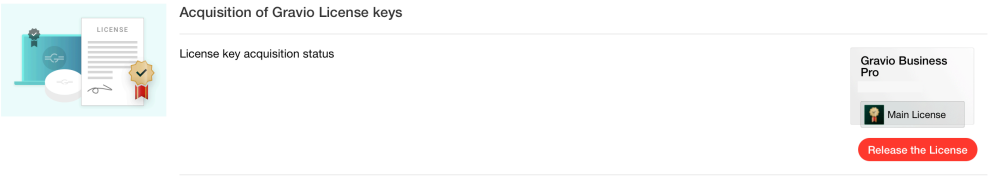


Click the “Obtain License” button to open the connection information panel to Gravio.com and enter your Gravio username and password under which you have registered your online account.



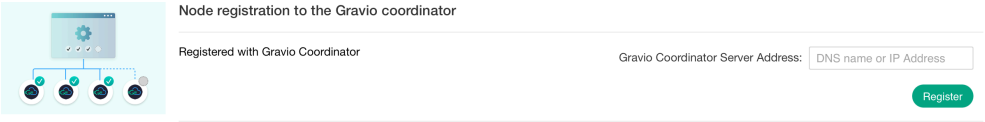
After you have entered your username and password, you will be authenticated and your associated licenses will be displayed. Select the license you like to install on the device you are currently logged in. The license file will be installed onto your device over the internet.

Apart from the different service levels are two types of licenses. “Main Licenses” including the Gravio Coordinator, and “Node Licenses” for HubKits.



The acquired license will be displayed. Select Release License to release it. You might like to do that to install the license on a different device.

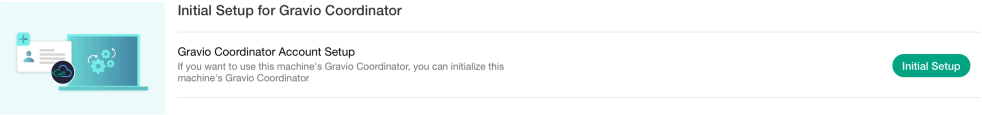
Register your Gravio HubKit Node with the Coordinator



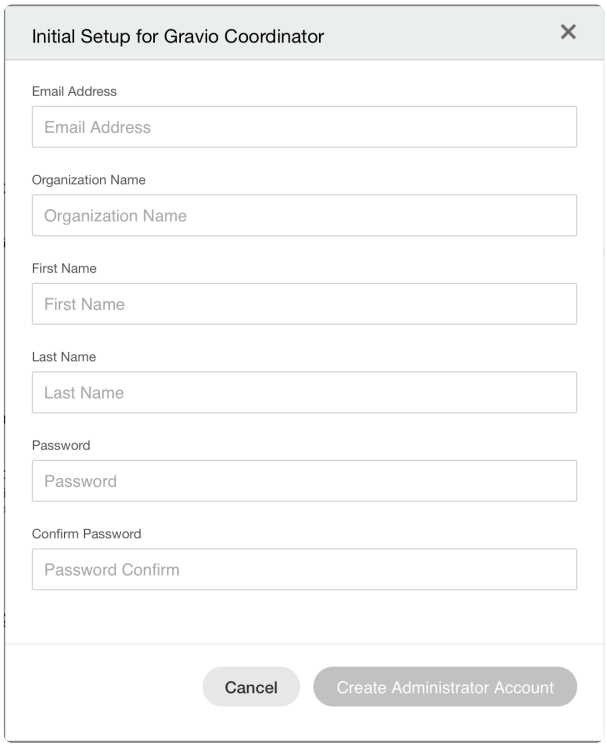
If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

Initialization of the Coordinator



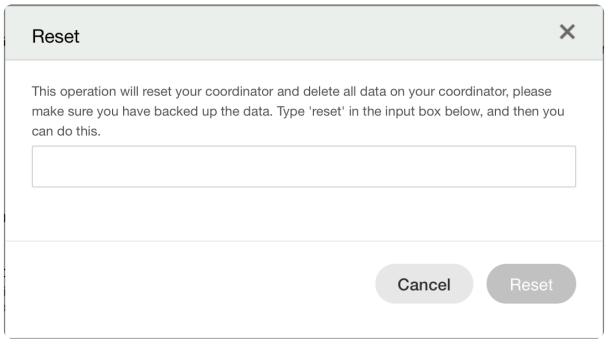
Upon the first launch, you have to create your Gravio Coordinator account. This will be used to manage all your HubKits.



Enter the account information and press the Create Admin Account button to create the Coordinator account.

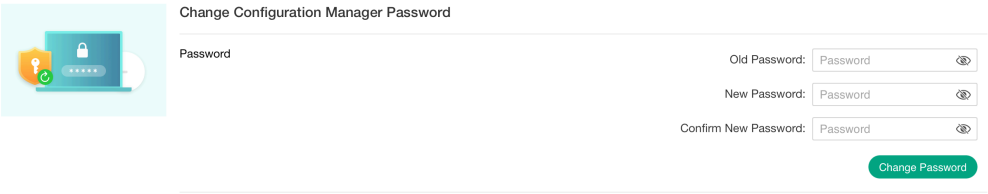
You can use this account to log in to the Coordinator. Please keep this information safe.

To initialize the created account, press the Reset button.



A confirmation panel will appear so you can confirm and initialize.

Change Password for the Gravio Configuration Manager



This is used to change the configuration manager password. The change will be effective from the next login.

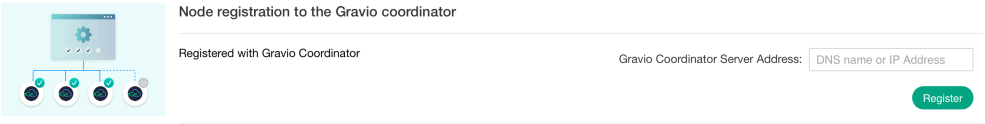
3.5. Installation and initial configuration of Linux (Ubuntu ARM64/Raspberry Pi) version

HubKit for Linux can be operated as a Gravio HubKit Node on a Raspberry Pi.

- Follow the instructions from the Linux ARM64 installation manual. Note, if you already have Gravio Studio, and just need to install the HubKit software, you can download this from https://www.gravio.com/en/download

- Open the Initial Setup screen to deploy your Gravio license.

- Register your Gravio HubKit Node with the Coordinator



If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

This completes the setup of the operating environment.

3.5.1. Installation of Ubuntu (ARM64) version

Download the installer for your operating system from https://www.gravio.com/en/download.

- Connect to Ubuntu 20.04 Server LTS using SSH. (Desktop and Core are not supported, use the Server version)
- Transfer the downloaded deb package (setup_-XXXX_arm64.deb for ARM64) to Ubuntu.
- Execute the following Command from Console.

```
sudo apt update
sudo apt upgrade
```

Next, install the deb package that you downloaded.

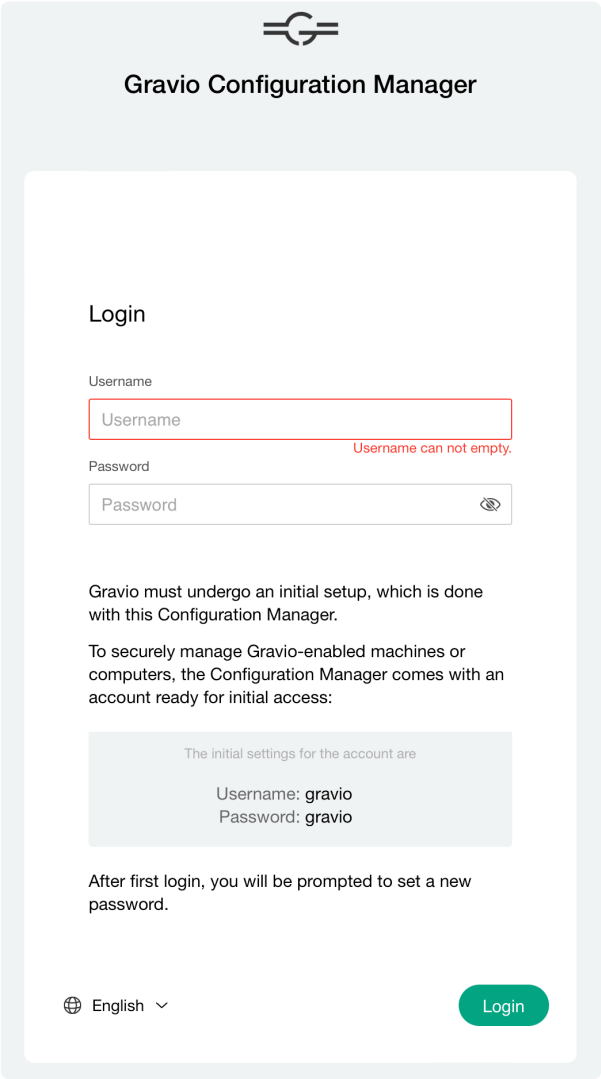
```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

After installing this deb package, the following Linux account will be created

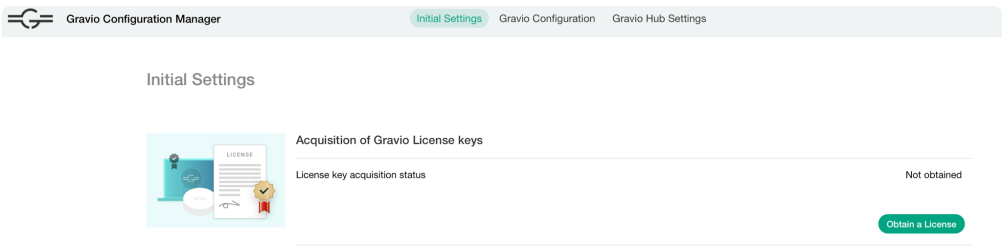
username : gravio
password : gravio

3.5.2. Initial Settings (Gravio Configuration Manager)

Open the Gravio Configuration Manager screen in your browser using Gravio Tools from Windows Tray. Pick the “Initial Setup” menu option.



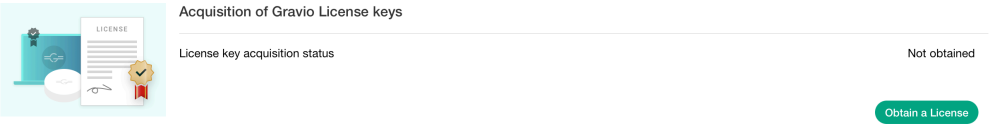
Or, in the browser, enter `<Hostname>.local:8080` configured in the previous chapter.



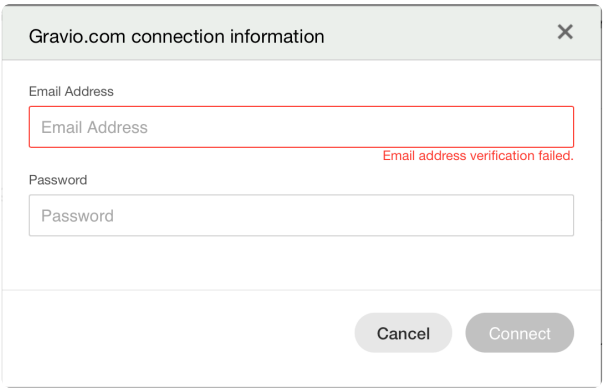
After logging in, the Gravio Hub Settings screen will open and the initial menu will open.

Deploying your License

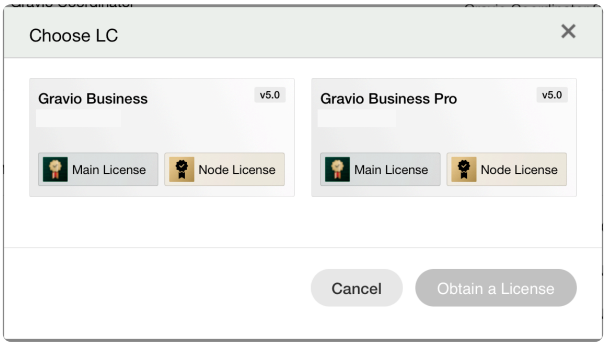
You must first obtain a license to use Gravio. This applies also if you are using the free Gravio version.



Install the license associated with the Gravio user into HubKit.

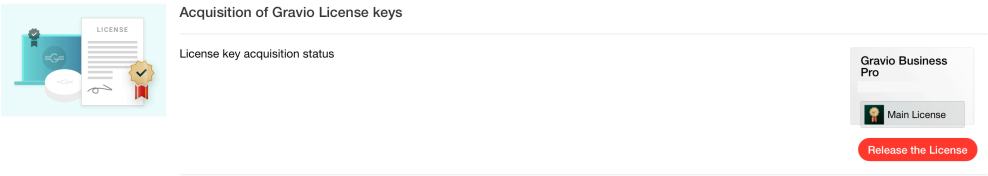


Click the “Obtain License” button to open the connection information panel to Gravio.com and enter your Gravio username and password under which you have registered your online account.



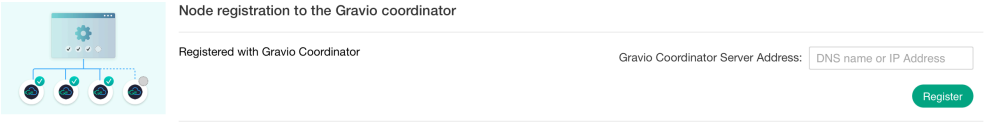
After you have entered your username and password, you will be authenticated and your associated licenses will be displayed. Select the license you like to install on the device you are currently logged in. The license file will be installed onto your device over the internet.

Apart from the different service levels are two types of licenses. “Main Licenses” including the Gravio Coordinator, and “Node Licenses” for HubKits.



The acquired license will be displayed. Select Release License to release it. You might like to do that to install the license on a different device.

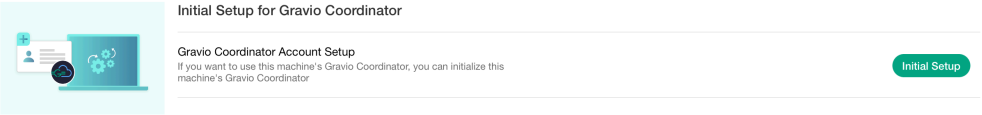
Register your Gravio HubKit Node with the Coordinator



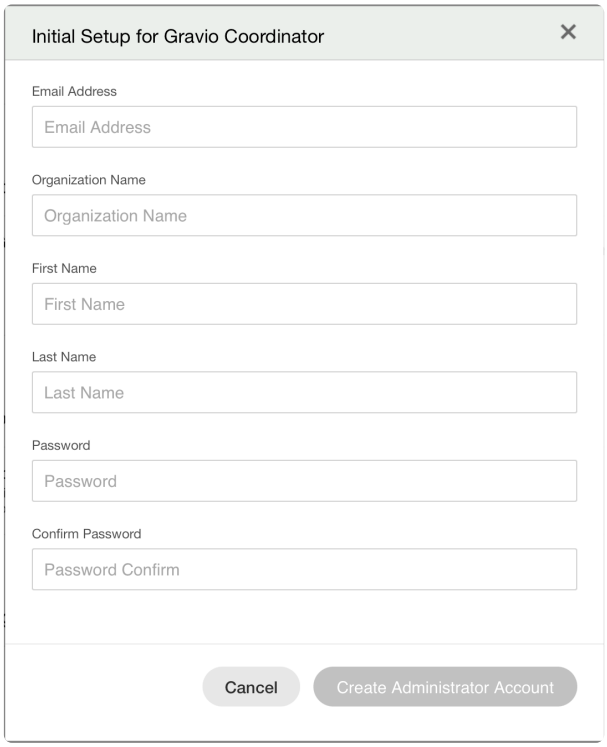
If you have a Gravio coordinator, you can register your HubKit on the Coordinator by entering the server address of the Coordinator and clicking the Register button.

To complete the registration of HubKit in the Coordinator, approval by the Coordinator is required.

Initialization of the Coordinator



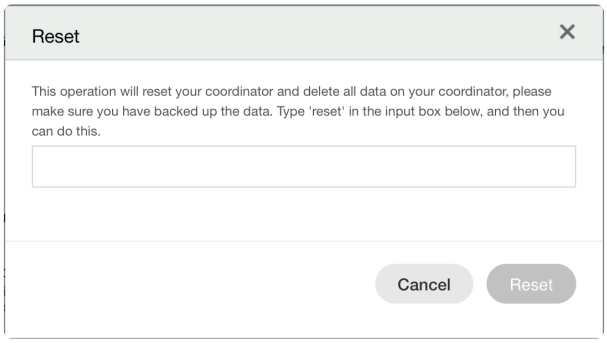
Upon the first launch, you have to create your Gravio Coordinator account. This will be used to manage all your HubKits.



Enter the account information and press the Create Admin Account button to create the Coordinator account.

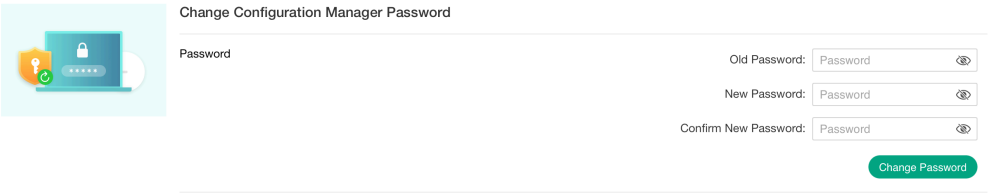
You can use this account to log in to the Coordinator. Please keep this information safe.

To initialize the created account, press the Reset button.



A confirmation panel will appear so you can confirm and initialize.

Change Password for the Gravio Configuration Manager



This is used to change the configuration manager password. The change will be effective from the next login.

3.6. Past HubKit installation and upgrade procedures

3.6.1. Upgrading to HubKit 4.3 on Linux / Raspberry Pi

To upgrade HubKit or Coordinator, please follow the procedure below. In a nutshell, we have migrated from a docker-based installation to a `.deb` package.

This document guides you through the process of upgrading HubKit 4.2 version to 4.3 on Linux / Raspberry Pi.

Update installation of HubKit 4.3 series and Coordinator 2.6 series

Download the server for the appropriate platform from the download site.

- Use SSH to connect to your Ubuntu 18.04/20.04 LTS machine.
- Transfer the downloaded deb package (setup_XXXX.deb) your machine.
- Execute the following Commands in the console:

Ensure you have the latest packages:

```
sudo apt update
sudo apt upgrade
```

Stop your current HubKit that is running”

```
cd <home directory which you run Gravio>
sudo docker-compose down
```

Delete the `gravio` user directory that has been created.

```
sudo rm -rf /home/<youraccoutforgravio>/
```

Install Update Manager, Coordinator, and HubKit

```
sudo apt install -y ./setup_XXXX.deb
```

Check Update Manager’s running.

Open a browser and access port 8080 of the IP address of Ubuntu 18.04/20.04 LTS.

https://(ip address):8008/

You will see a holding screen while the Update Manager processes the Coordinator and HubKit. Please wait until the login screen appears to proceed with the next step.

Migrating the data of the running Coordinator and HubKit.

(The directory of the newly installed HubKit 4.3 is backed up to hubkitrepo4_bak just in case.)

```
cd /home/gravio
sudo mv hubkitrepo4 hubkitrepo4_bak
sudo mv /home/(Username where HubKit 4.2 is installed)/hubkit/hubkitrepo4 .
sudo chown -R gravio:gravio hubkitrepo4
```

Now that the configuration is complete, reboot Ubuntu.

```
sudo reboot
```

Logging in to Update Manager

After rebooting Ubuntu, open a browser and access port 8080 of the Ubuntu 18.04/20.04 LTS IP address.

```
https://(ip address):8080/
```

For instructions on how to use Update Manager, please refer to here.

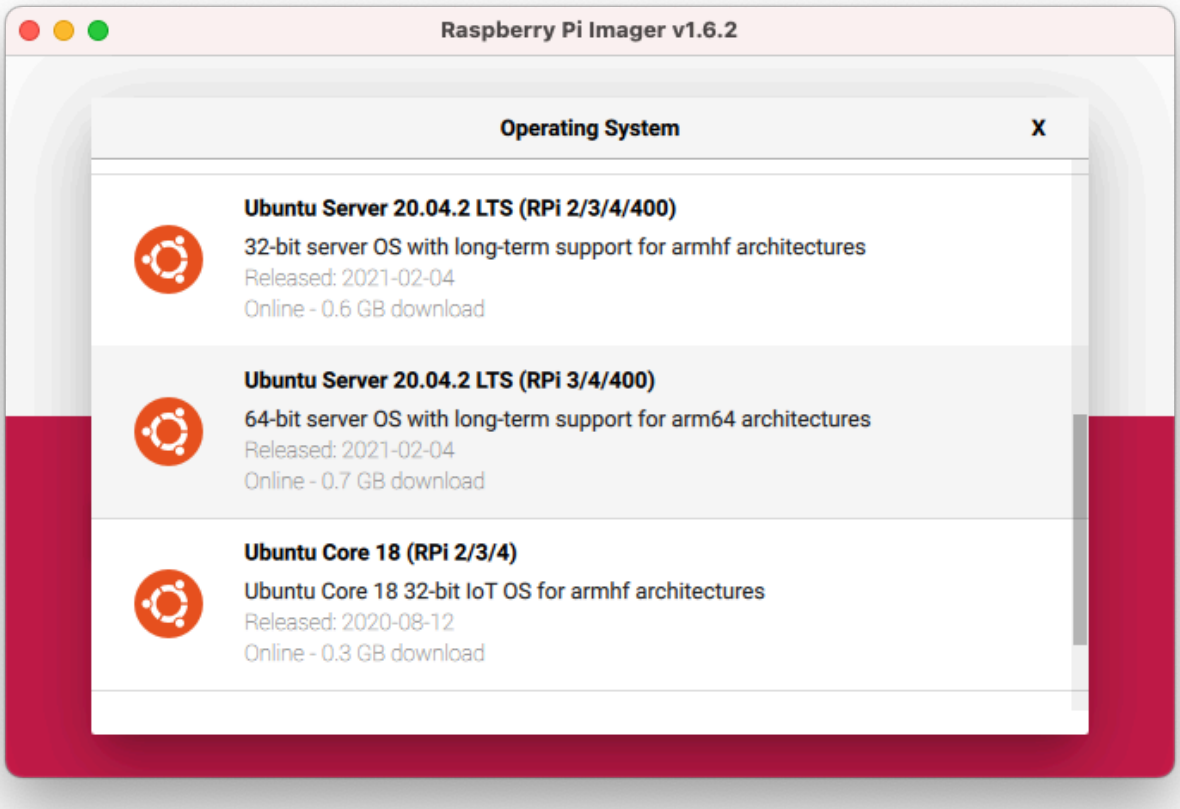
3.6.2. Setup HubKit 4.3 on Linux / Raspberry Pi

Linux / RaspberryPI

Installation of HubKit 4.3 and Coordinator 2.6

We have created a .deb package for the installation. This makes the installation of HubKit 4.3 very easy on Linux.

Please note that if you use a Raspberry Pi, you must have a 64bit Raspberry Pi. This means it has to be a Raspberry Pi 4. We suggest 4 GB RAM or more. The Raspberry Pi 4 uses the ARM64 processor version. We currently do not support 32bit OSes such as Raspbian. To install on a Raspberry Pi, we recommend the Raspberry Pi Imager and install the Ubuntu Server 20.04.2 LTS on your SD Card.



Once you have Ubuntu installed and connected, download the Gravio HubKit server for the appropriate platform from the download site.

- Use SSH to connect to Ubuntu 18.04 or 20.04 LTS. (Gravio requires Ubuntu **LTS** only)
- Transfer the downloaded deb package (`setup_-XXXX_amd64.deb` for AMD64, `setup_-XXXX_arm64.deb` for ARM64) to Ubuntu.

- Execute the following commands in your console:

Ensure your apt package manager is up to date:

```
sudo apt update
sudo apt upgrade
```

Install Update Manager, Coordinator, and HubKit. Very important: After the installation **do not reboot**. Do the next step first. Why not reboot? After the installation, Gravio might be downloading some updates from the internet. If you reboot during that process, the installation may break. Install by entering:

```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

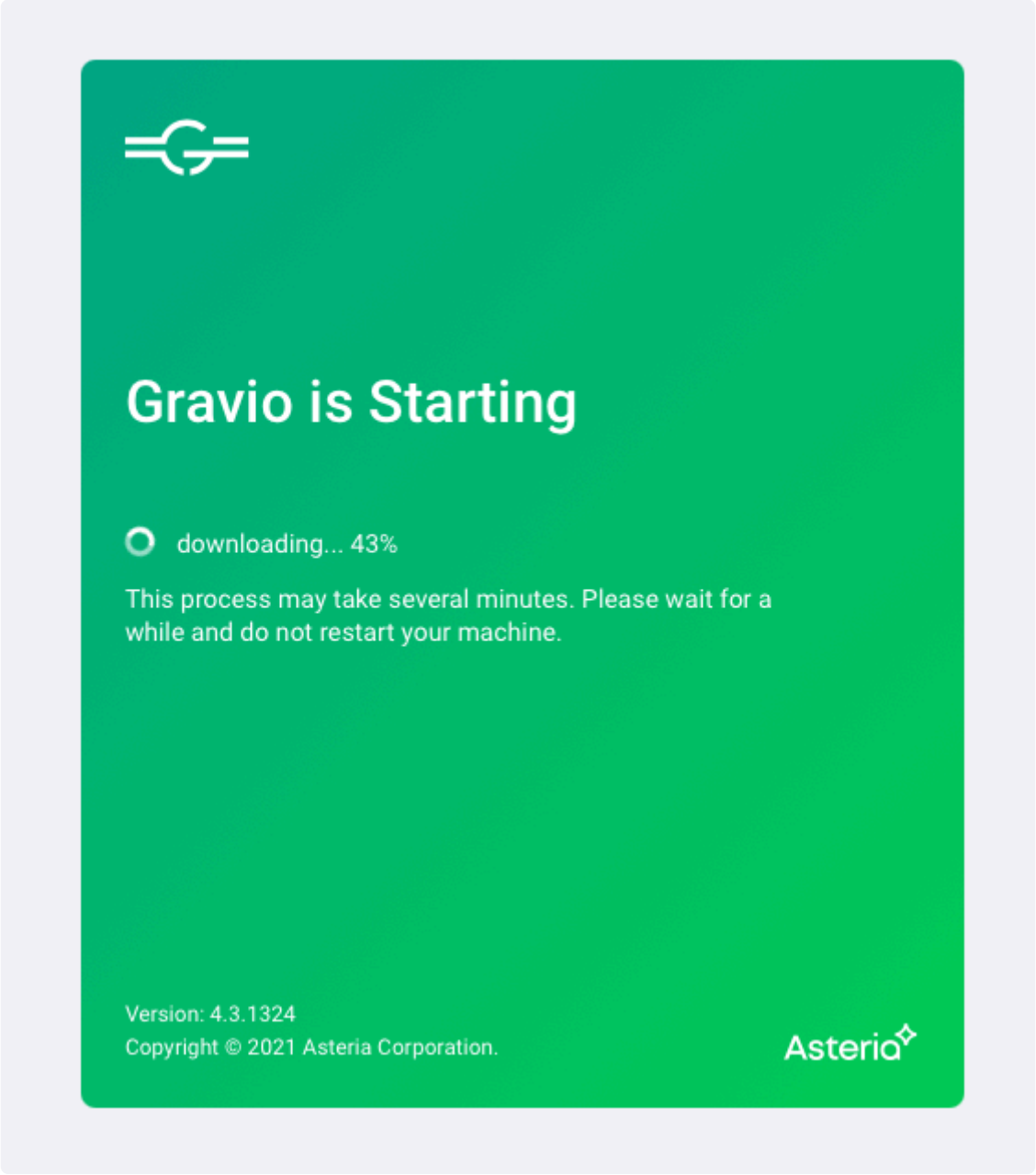
If you happened to reboot after the installation, and or your installation is corrupted in any way, please use this command to re-install: `sudo apt --reinstall install -y ./setup_-XXXX_XXX64.deb`

Login to Update Manager

Open a browser and access port 8080 of the IP address of your Ubuntu 18.04/20.04 LTS machine.

`http://(ip address):8080/`

You will see a holding screen while the Update Manager processes the Coordinator and HubKit.



Once the Coordinator and HubKit are set up, you will see a login screen where you can log in. Please refer to here to learn how to use the Update Manager.

3.6.3. License Management in Gravio 4.5

From Gravio 4.5, a license management feature has been added for customers with a Standard contract or higher. If you are using Gravio 4.5 or higher, please follow the steps below to register your license.

*Please note that the contract content of Gravio Enterprise AI Edition Platinum is used as an example on the screen. Depending on your contract, there may be differences in Gorilla licenses and other restrictions.

License Management Process

From Gravio 4.5, you register the Hub ID on Gravio.com and manage the license.

- If you are using the Coordinator in an environment connected to the internet, please follow the steps: 1. Register Hub on Gravio.com → 3. Manage Hub License with Gravio Coordinator

- If you are using the Coordinator in an offline environment, please refer to here and use Gravio's license file.

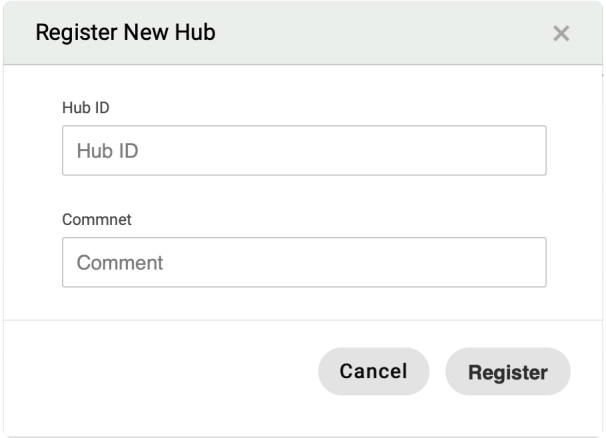
- If you are not using the Coordinator and the Hub is running in an environment connected to the internet, follow the steps: 1. Register Hub on Gravio.com → 4. Manage Hub License with Gravio Studio

- If you are not using the Coordinator and the Hub is running in an offline environment, follow the steps: 1. Register Hub on Gravio.com → 2. Download the Hub license file → 4. Manage Hub License with Gravio Studio

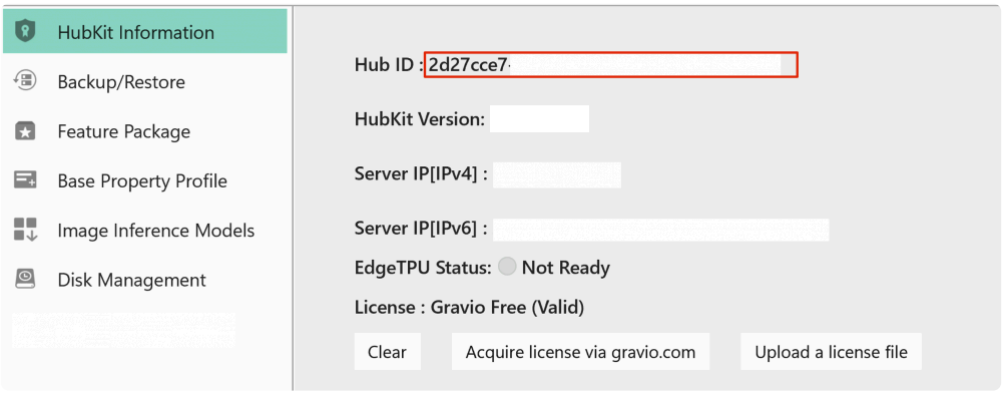
1. Registering Hub on Gravio.com

Log in to Gravio.com and open the Hub license.

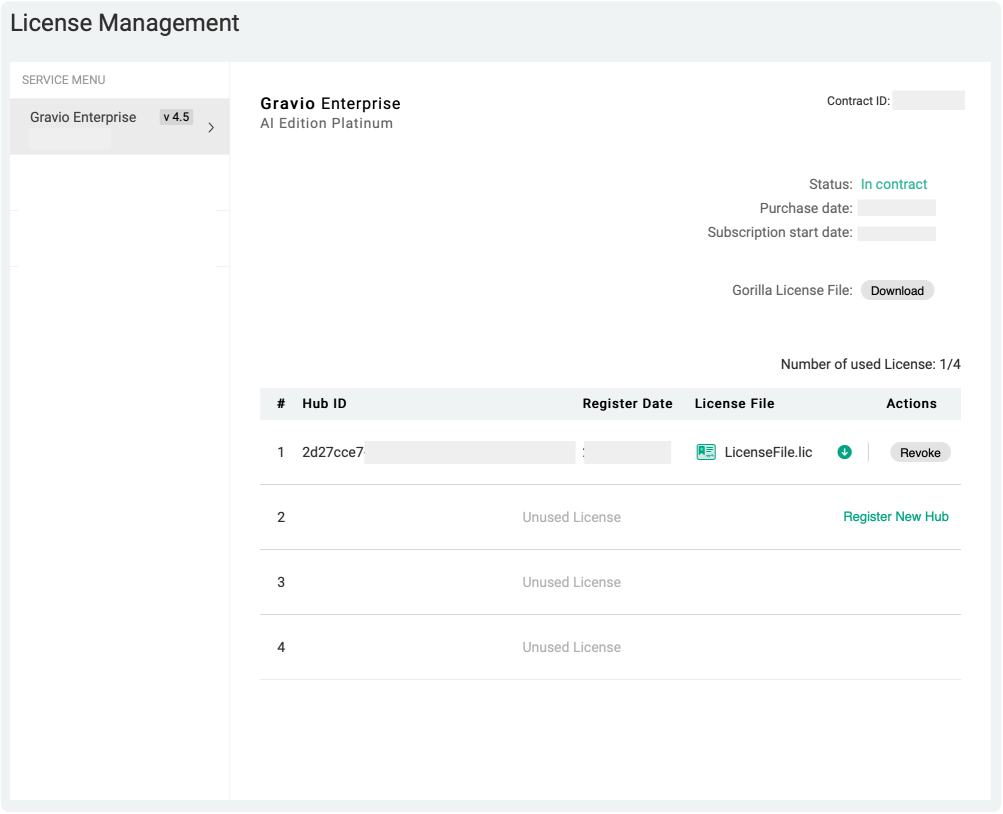
Press the Register Hub ID button and register the Hub ID of the Hub for which you want to register the license.



The Hub ID is displayed in the HubKit Information on the Settings panel when you open the Hub from Gravio Studio, so please copy and register it.



The registered Hub ID will be displayed in the list.



To delete a registered Hub ID, press the Invalidate button.

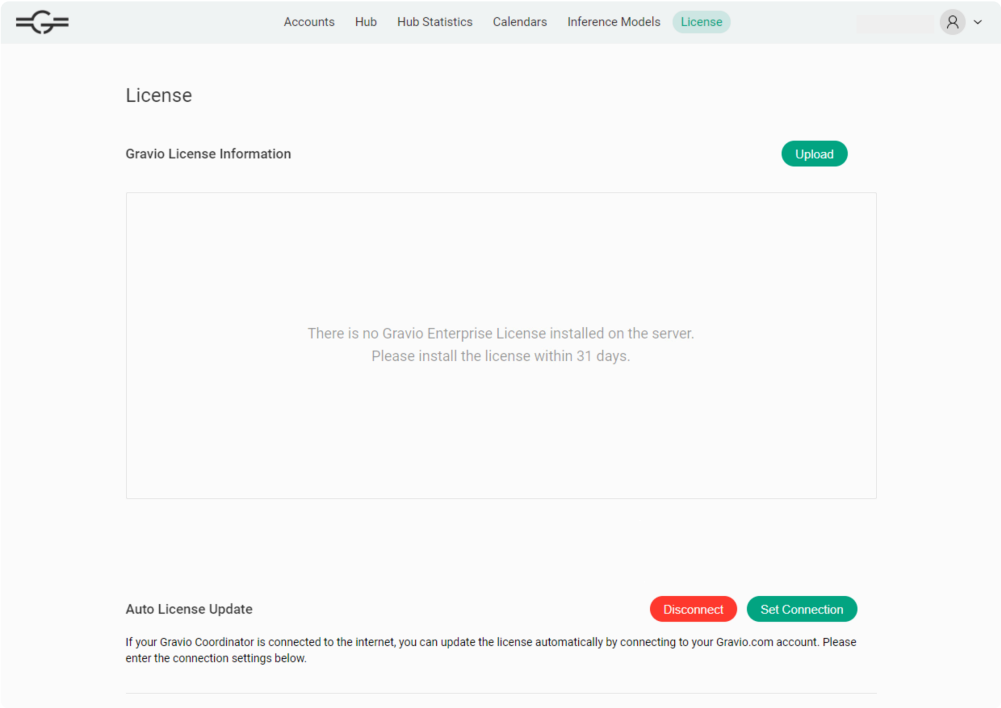
2. Downloading the Hub License File

When the Hub is connected to the internet, the license is regularly checked on Gravio.com. However, if the Hub is offline, you cannot check automatically, so you need to register the license file to the Hub.

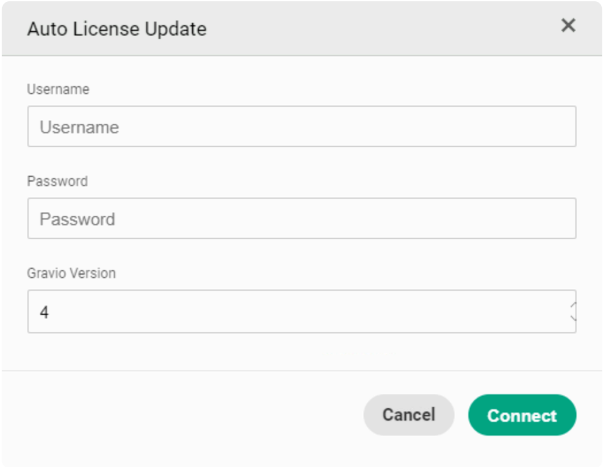
If you are using the Hub offline, click on the LicenseFile.lic of the corresponding Hub ID to download the license file.

3. Managing Hub License with Gravio Coordinator

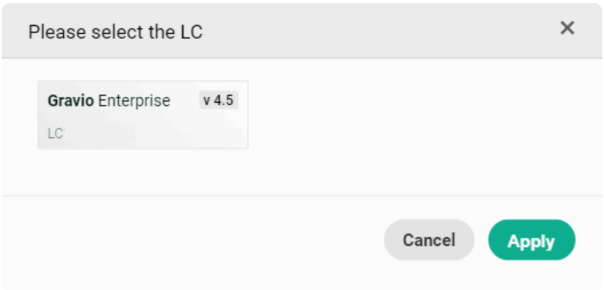
If you are using Gravio Coordinator and connected to the internet, you can update the license file from Gravio.com.



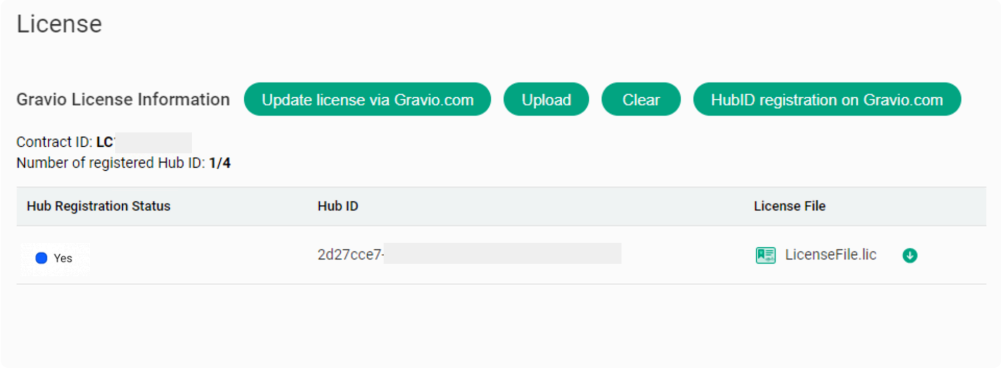
First, press the Settings button in Gravio.com connection information to register your account ID and password.



When the connection status of Gravio.com connection information becomes Connection OK, press the Update license from Gravio.com button displayed in Gravio license information. The contracted license is displayed, so select it and press the Apply button to update.



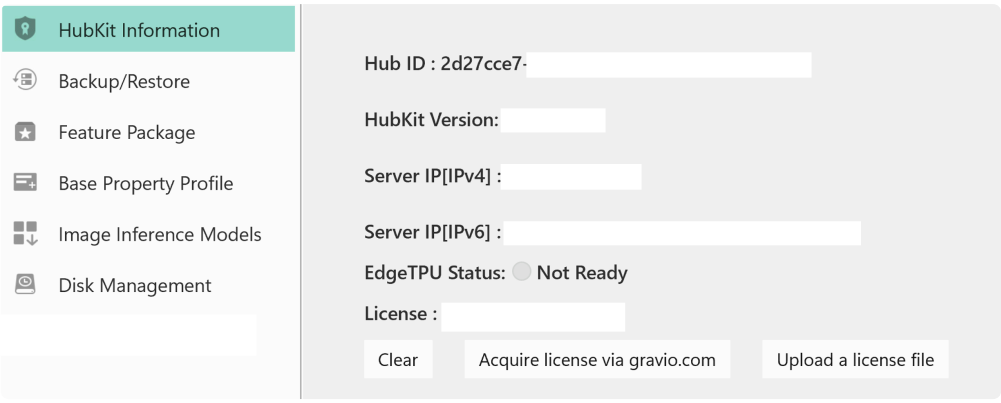
The license is updated from Gravio.com and displayed.



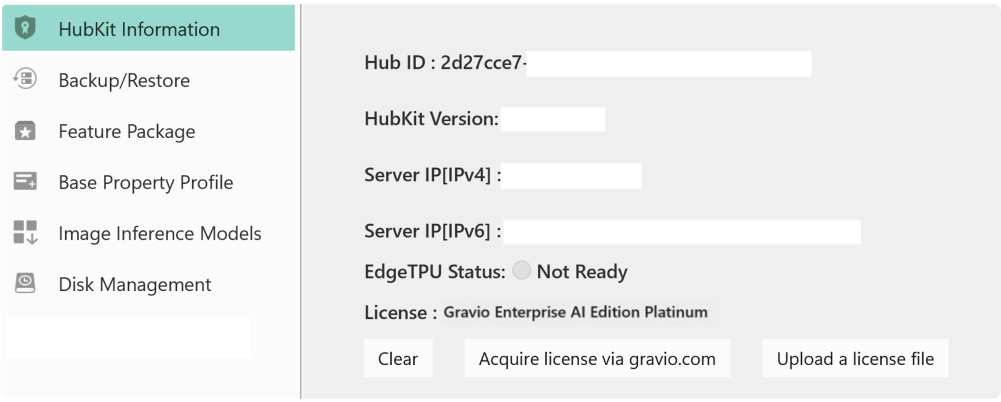
In case you are using Gravio Coordinator in an offline environment not connected to the internet, please refer to here and update the Gravio license file from the upload.

4. Managing Hub License with Gravio Studio

If the Hub is connected to the internet, you can update the license file from Gravio.com.



When you press the Update license from gravio.com button, the license file is obtained and displayed.



If you are using the Hub in an offline environment, specify the license file you downloaded in 2. Download the Hub License File by pressing the Upload License File button. The license file is loaded and displayed.

4. Using Gravio Cloud

Gravio Business and Gravio Business Pro customers have access to Gravio Cloud servers.

The Gravio Cloud servers are AWS-based cloud servers that Asteria provides for each MAIN server license for one hosting server by using the MAIN license included in your Gravio contract. The MAIN server license is a one MAIN server license per MAIN server license.

Please follow the instructions in the “Server Build Request” section below to create and use a server.

4.1. Request to establish Gravio Cloud Server

How to use Gravio Cloud

With Gravio 5.0, Business and Business Pro license holders have a MAIN license to use Gravio in the cloud.

Gravio Cloud allows you to run Gravio 5.0 on your own dedicated server.

If you are already using a MAIN license in your local environment, you can release the license to use the Cloud service.

For Gravio Free licensees

Gravio Free licensees can apply for a functional evaluation license, which will allow them to use Gravio Cloud for a 3-week trial.
For more information on the components that can be run with a Business Pro license, please see here.
For more information on the components that can be run with a Business Pro license, please refer to here.

Building a Gravio Cloud server

For information on how to build a Gravio Cloud server, please see here if you have a Business or Business Pro license, or here if you have a Gravio Free license. “:#gravio-free-gravio-cloud.

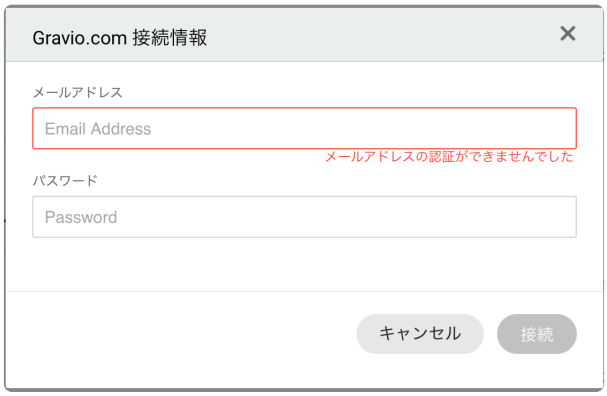
4.1.1. For use by Business and Business Pro purchasers

obtain a license

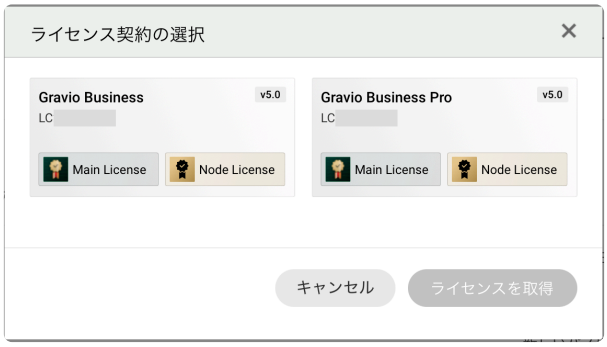
Follow the steps below to obtain a Business or Business Pro license.

For Gravio Hub2, open the Configuration Manager and go to the Preferences tab to obtain a license.
For HubKit, open the configuration menu for the respective platform and acquire the license.

For details on the initial settings of Gravio Hub2 configuration manager, please refer to here.
*For the initial settings of HubKit's configuration manager, please refer to Windows macOS Linux LINK+gravio-amd64-init} Linux



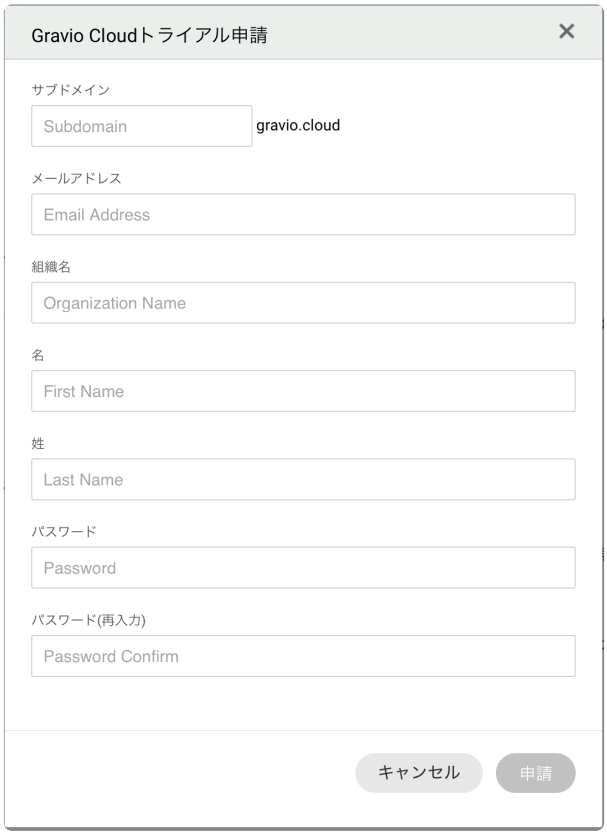
Enter the account information you signed up with on Gravio.com.



Select and obtain the Node License among the licenses displayed.

Request Gravio Cloud server installation!

Once you get the license, Reload your browser screen once. Then you will see a button to request installation of a Gravio Cloud server in the section for node registration with the Gravio coordinator.



Clicking on the Request button will display a panel where you can enter the necessary information for the request.

Once your request is accepted, an email will be sent to your registered email address.

Start using Gravio Cloud server

Once the server construction is complete, an email will be sent to you with the server information. Once you have received the email confirming the completion of the construction, you will be able to access the input panel for information necessary for your request with the host name you have registered.

Connection from Gravio Studio

When connecting to the Gravio Cloud server from Gravio Studio, be sure to connect using Login to Gravio Coordinator at Gravio Studio Login UI.

Start Gravio Studio and Login to Gravio Coordinator to display the login screen,
Enter the Coordinator's URL, email address and password to login.

The URL of the Coordinator is the host name of the Gravio Cloud server. (https://{Gravio Cloud server host name})
Please use this email address and password when accessing the Gravio Coordinator WebUI.

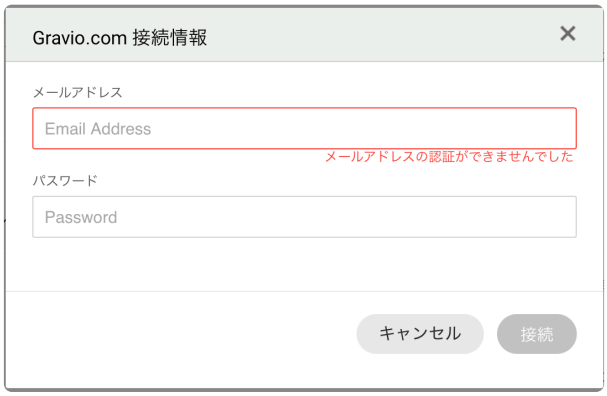
4.1.2. Gravio Free users use the Gravio Cloud server for evaluation purpose

Obtain a license

Follow the steps below to obtain a license for Gravio Free.

For Gravio Hub2, open the Configuration Manager and go to the Preferences tab to obtain a license.
For HubKit, please open the configuration menu for each platform and acquire the license.

For details on the initial settings of Gravio Hub2 configuration manager, please refer to here.
*For the initial settings of the configuration manager of HubKit, see Windows macOS Linux LINK+gravio-amd64-init} Linux



Enter the account information you signed up with on Gravio.com.

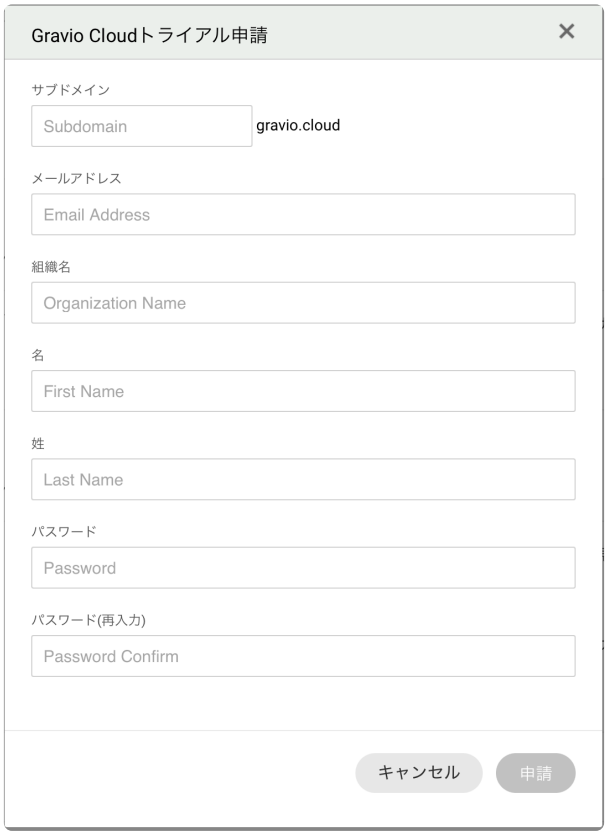
Select the Node License to obtain the license displayed.

Request Gravio Cloud server installation

After obtaining the license, %(marker-red)reload your browser screen once. Then you will see a button to apply for a functional evaluation license.

The evaluation license allows you to evaluate the functionality of Business Pro for a period of 3 weeks.

After you have completed your request for a functional evaluation license, %(marker-red)reload your browser screen again. A button to request installation of a Gravio Cloud server will appear in the %Register a node with the Gravio Coordinator section.



Clicking on the “Request” button will display a panel where you can enter the necessary information for the request.

When your request is accepted, an email will be sent to your registered email address.

Start using Gravio Cloud servers

Once the server build process is complete, an email will be sent to you regarding your server information. Once you have received the email about the completion of the build, you can access the input panel for the information necessary for your request with the host name you registered.

Connecting from Gravio Studio

When connecting to the Gravio Cloud server from Gravio Studio, be sure to connect using Login to Gravio Coordinator at Gravio Studio Login UI.

Start Gravio Studio and Login to Gravio Coordinator to display the login screen,
Enter the Coordinator’s URL, email address and password to login.

The URL of the Coordinator is the host name of the Gravio Cloud server. (https://{Gravio Cloud server host name})
Please use this email address and password when accessing the Gravio Coordinator WebUI.

5. Account types for Gravio operations

Several accounts are required to use Gravio.

Depending on the functionality of the Gravio system, you may need to create several accounts to access different features.

1. Gravio User Account

This account is created by signing up in Gravio Studio or by creating an account on Gravio.com at Create Account.

Once you are a registered Gravio User, you will be able to configure your Gravio Hub and Hub PC in Gravio Studio.
The purchased license will be associated with and managed by this Gravio User.

h3.2 Gravio Configuration Manager Account

When you install the HubKit package, a dedicated account called Gravio Configuration Manager is automatically created after installation to configure the HubKit environment for your PC.
The user name is fixed and will be named “gravio” .

The default password is “gravio” . The first time you log in, you will be asked to change the initial password.

This Gravio Configuration Manager account will be the Linux account “gravio” for the Linux version and the GravioHub for the installed OS. h3. 3.

h3.3 Gravio Coordinator Account

The Gravio Coordinator is used to manage users who have access to the NodeHub.
The first Gravio Coordinator account with Admin rights can be created in the Gravio Configuration Manager under Initial Settings.

Gravio coordinators can create accounts with the following user privileges in the “Accounts” administration: {#404.

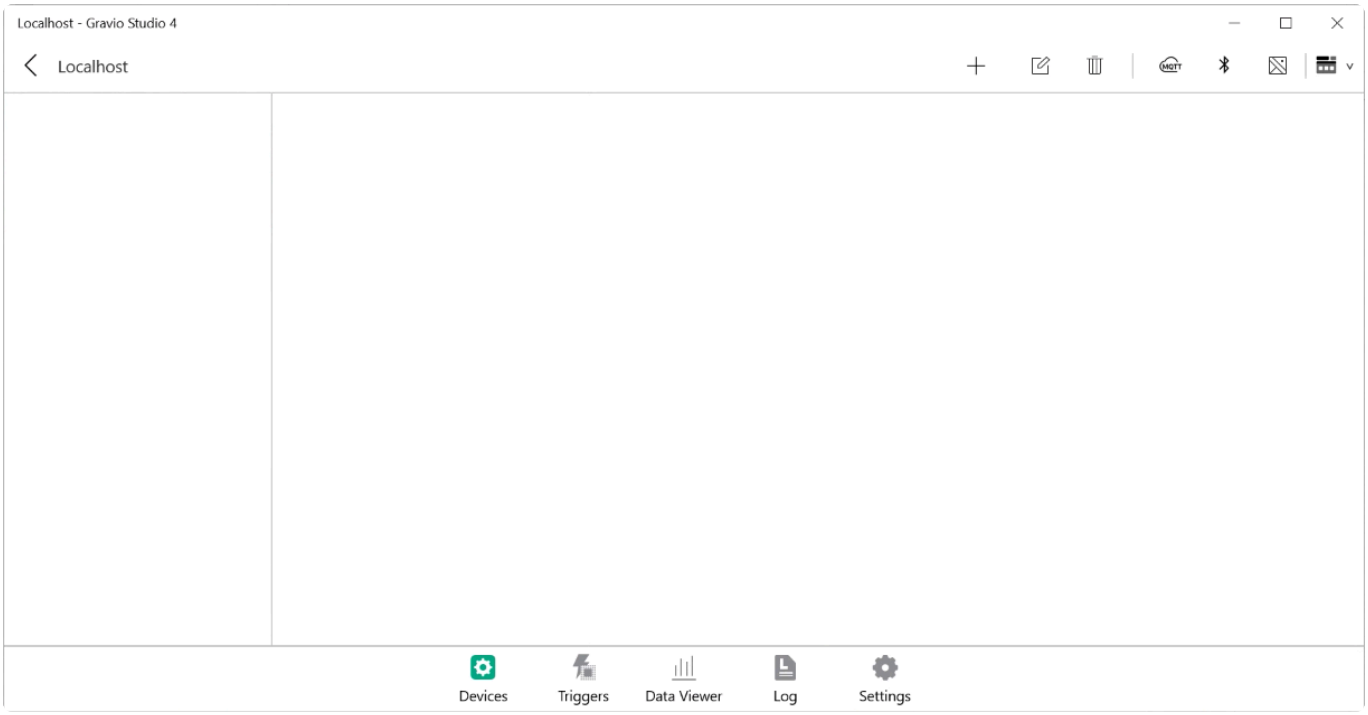
Available Screens	Admin	Developer	Maintainer	AppUser
Account	○	△ (only myself)	△ (only myself)	△ (only myself)
HubManagement	○	○	○	X
Statistics	○	○	○	△ (view only)

Calendar	○	○	○	X
Inference Model	○	○	○	X
カレンダー	○	○	○	X

6. Using Gravio Studio

6.1. Hub Configuration View

If you double click on the upper half of the Server card, you will get to the Hub Configuration view:



At the top you can find a number of buttons such as:

- A plus sign button to create new physical Areas to group your Layers
- An edit button to edit an Area or Layer
- A trash bin to delete Areas or Layers

Separated from that, you find:

- A MQTT Cloud to connect to an MQTT Broker. This will behave like an input device.
- A Bluetooth icon to connect new Bluetooth devices
- A xyz icon to connect to hardware sensors or serial port dongles

- In the far right, you see the Actions Editor icon, where you can jump straight to one of your actions.

At the bottom you find a tab bar, containing 5 tabs:

- Devices — where you set up your sensors and place them on areas and layers.
- Triggers — where you set triggers to execute Actions (see Actions).
- Data Viewer — where you can find current sensor data. The sensor values are visible in this tab.
- Log Viewer — where you can find the Gravio Server logs for debugging and optimisations.
- Settings — where you can find server-related settings and perform backups.

The big middle section is split into the left side, for the Areas and the right side, for the Layers and devices.

6.1.1. Device Tab

In the Devices tab, you manage the devices connected to a HubKit by adding, removing and configuring them, including their physical location.

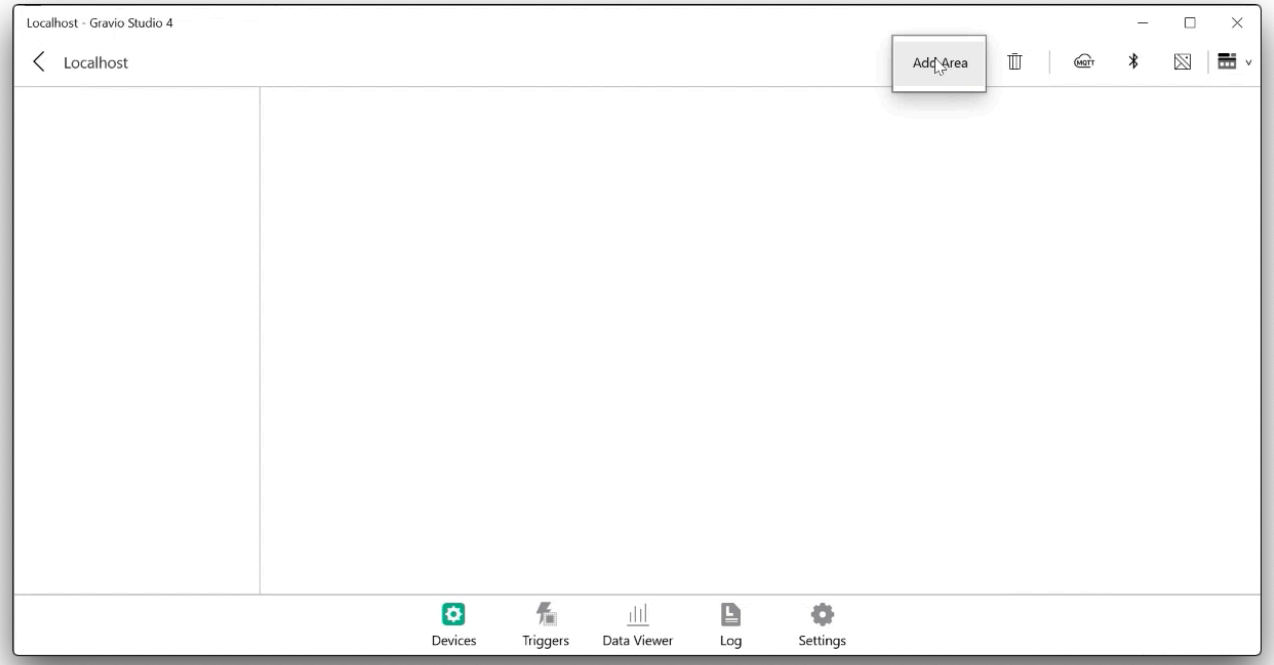
In order to keep the data stores organised and efficient, Gravio features two dimensions how to place and assign sensors:

1. The Areas
2. The Layers

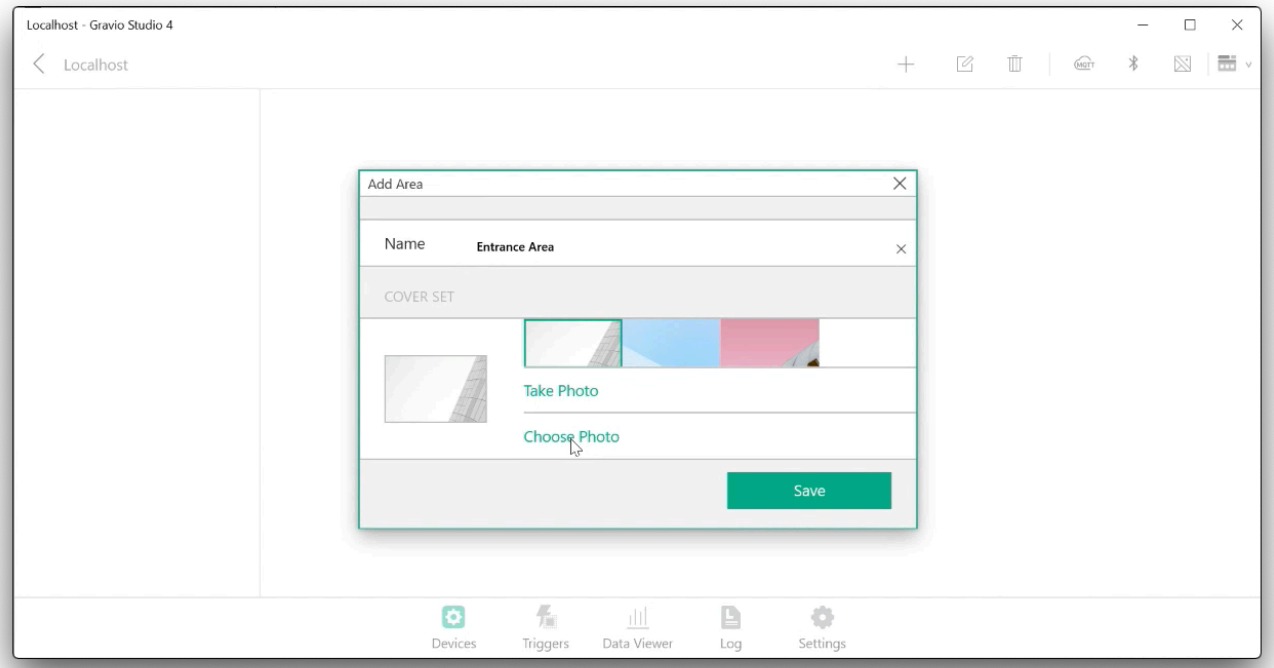
Adding Areas to the Server

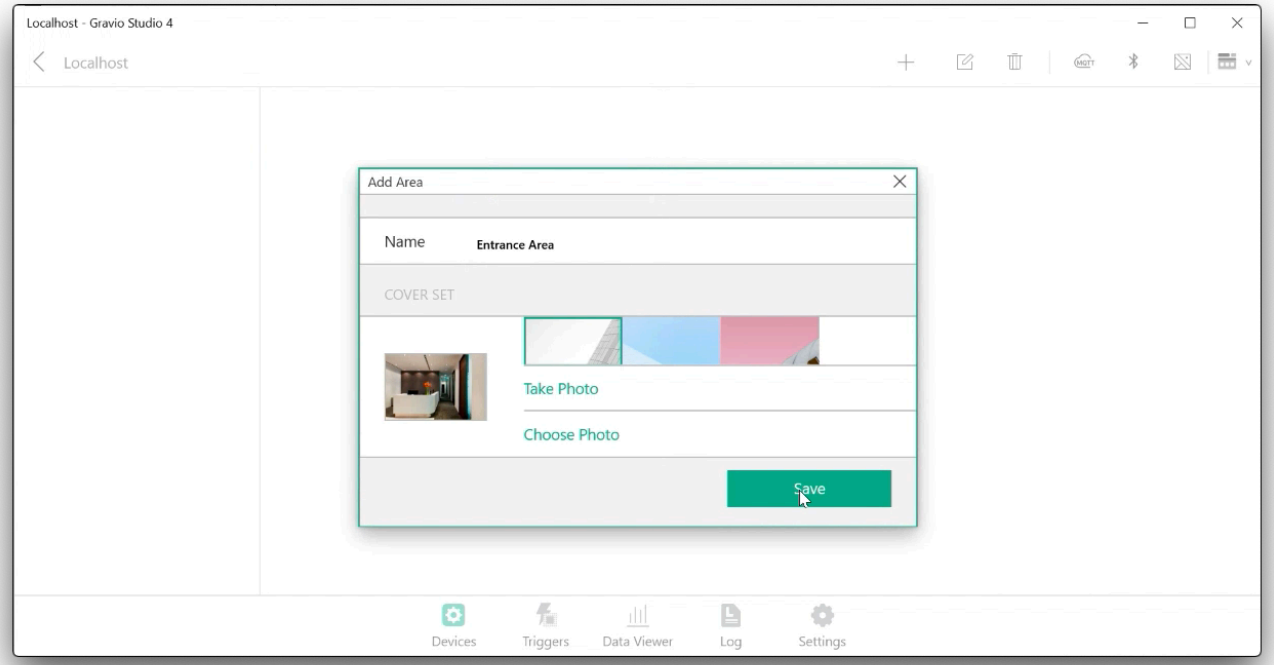
Areas define a physical location, such as for example “3rd Floor West”, “Seminar Room”, “Entrance Area” etc.

Areas can have multiple **Layers**, which are used to group sensor information according to their functionalities, e.g. “Temperature Layer” or the “Camera Devices Layer”. All layers must contain the same sensor device types.



Adding an Area is easy, simply click on the “+” mark at top right and you can add a new Area.

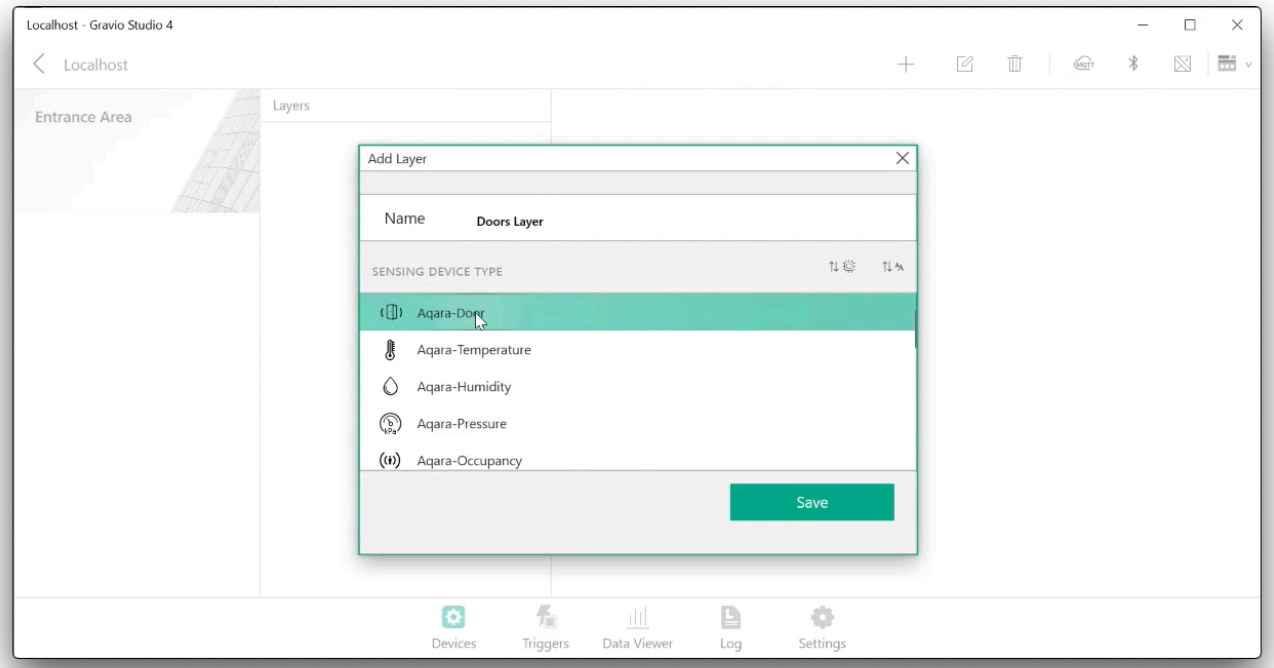




We suggest you add a photo to identify the Layer easily in the future.

Adding Layers to the Areas

Next, let's add a few layers to an area. Give the Layer a meaningful name …



… and choose the type of sensors the Layer should incorporate.

Depending on what device you choose, you have different options. Gravio supports:

- Physical Sensor Devices
- Software Sensors
- MQTT Subscriptions

6.1.1.1. Adding Physical Sensor Devices

Gravio is built in a way that allows us to easily extend new sensor families. On the website https://www.gravio.com/product#Gravio-Rent you can see a range of hardware sensors Gravio supports off the shelf. If you require other sensors to be connected, please do not hesitate to get in touch via Slack or e-mail under support@gravio.com

As of today, we support two types of widely known sensor standards:

- Zigbee devices (Zigbee specification)
- Bluetooth devices
- Energy harvesting EnOcean devices (EnOcean specification)

Using Zigbee

In order to use the Zigbee devices, you need to either use the Gravio Hub or use the below depicted Gravio Zigbee USB receiver dongle with your PC/Mac that will run the Edge Server:

The Bluetooth devices need to be supported by Gravio. Please ask us in the Slack channel if a device is supported.

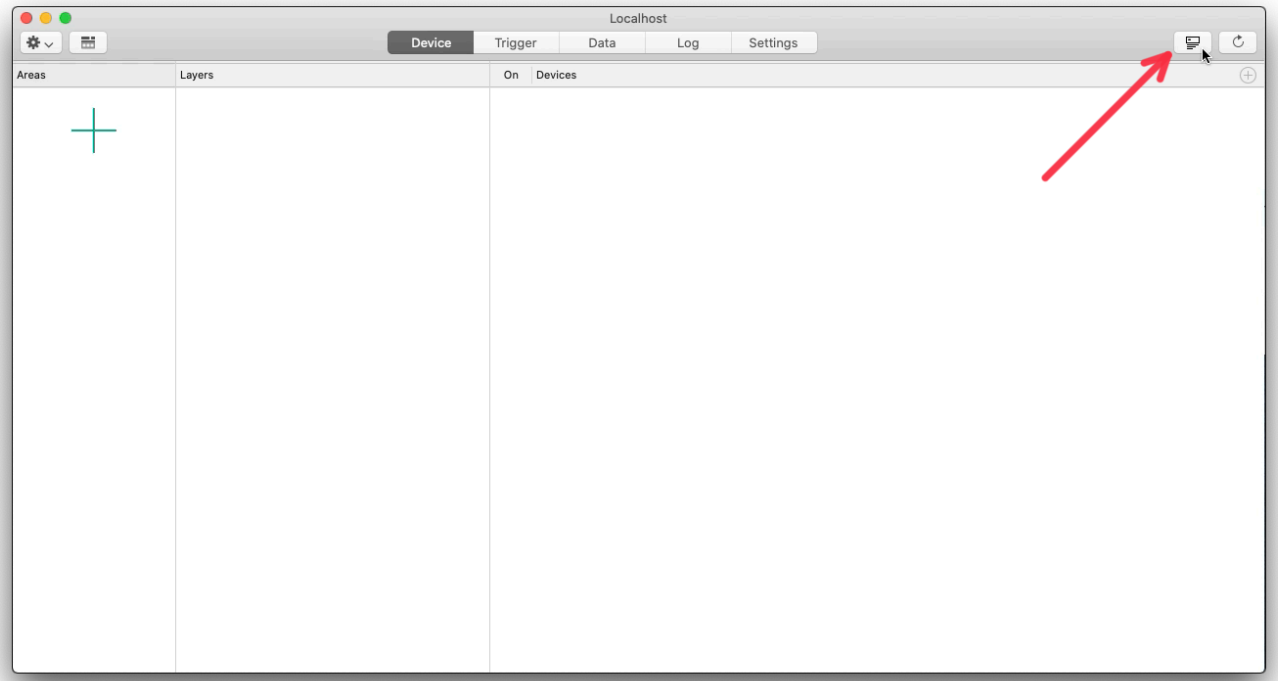
The EnOcean sensors do not require any additional driver as the EnOcean driver is built into the Gravio HubKit application. If you have Gravio HubKit and the EnOcean dongle connected, you are ready to connect to your sensors out of the box. For more details how to use EnOcean, please refer to Adding EnOcean Devices and Sensors

Adding Devices to the Layers

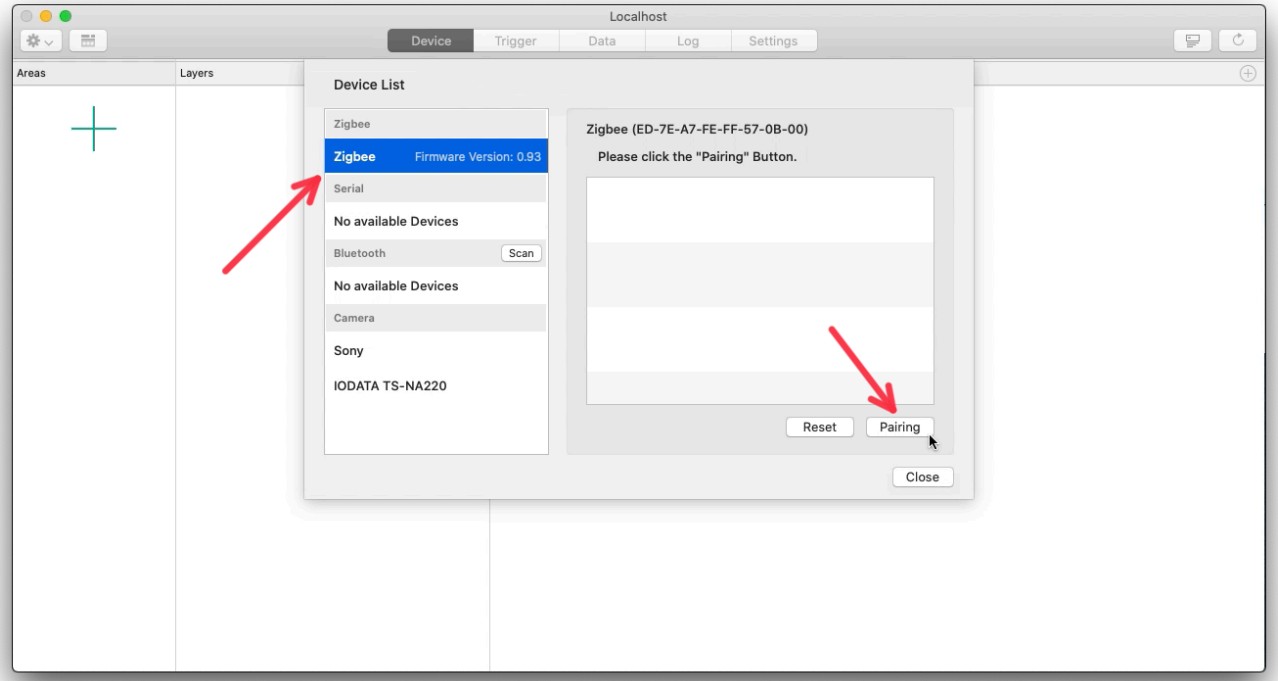
Now that you are ready to connect to the sensors, you need to bind the receiver(s) to the layers you want to receive the data to. In order to do this, start Gravio Studio for Mac or PC, connect to your Gravio HubKit and start adding your sensor devices. In this example, we are adding a Zigbee Single Button

sensor as illustration:

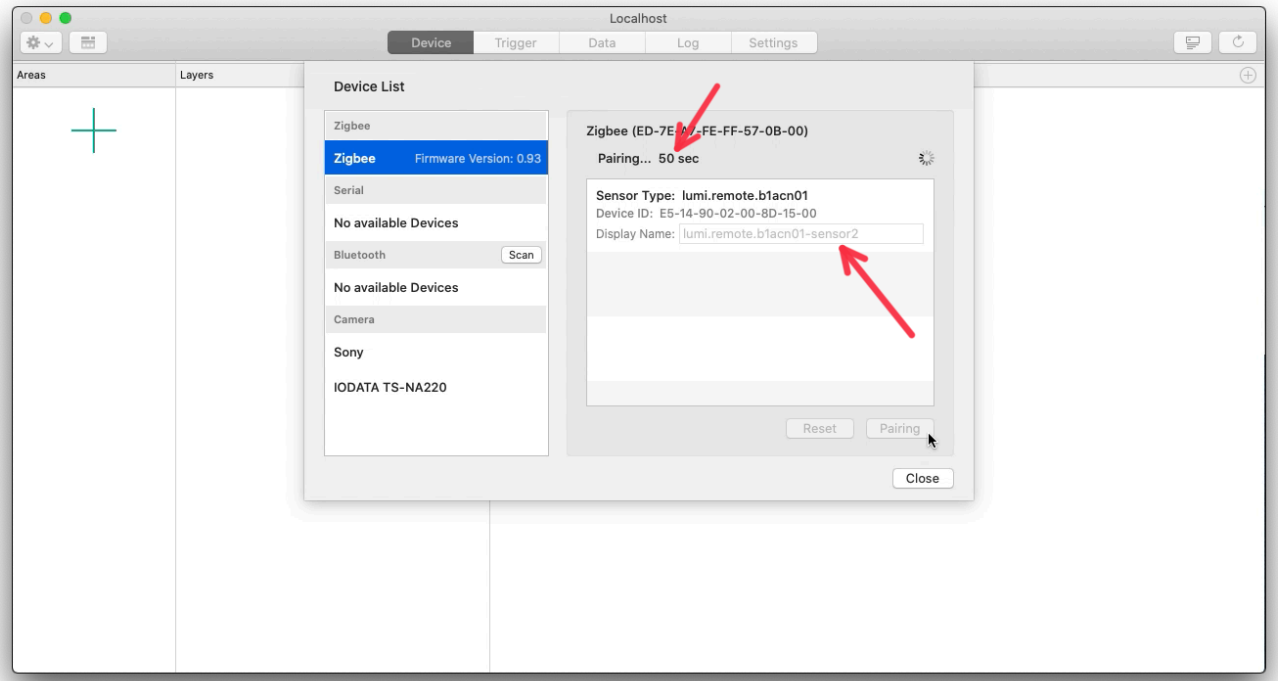
Click the devices icon on the top right:



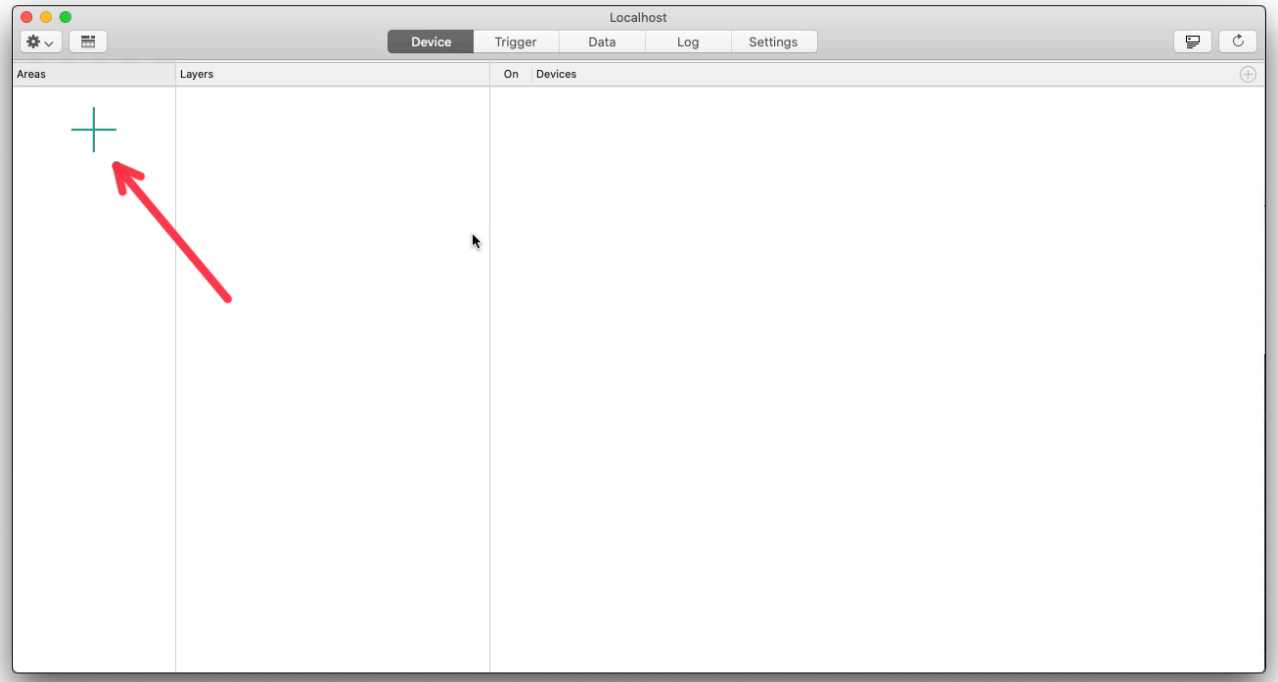
Select the Zigbee section on the devices list, and click the Pairing button:



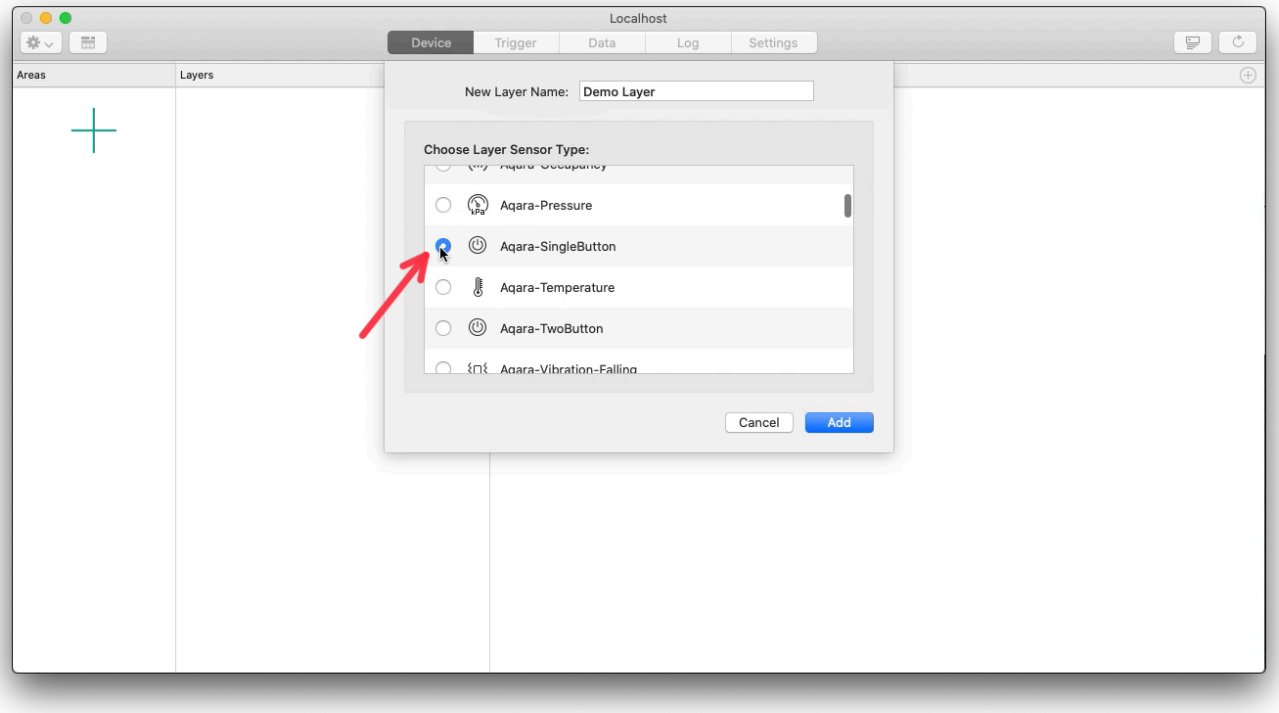
A countdown will start in which you can pair your sensors. Click the pairing buttons on the sensor a few times until it appears in the list:



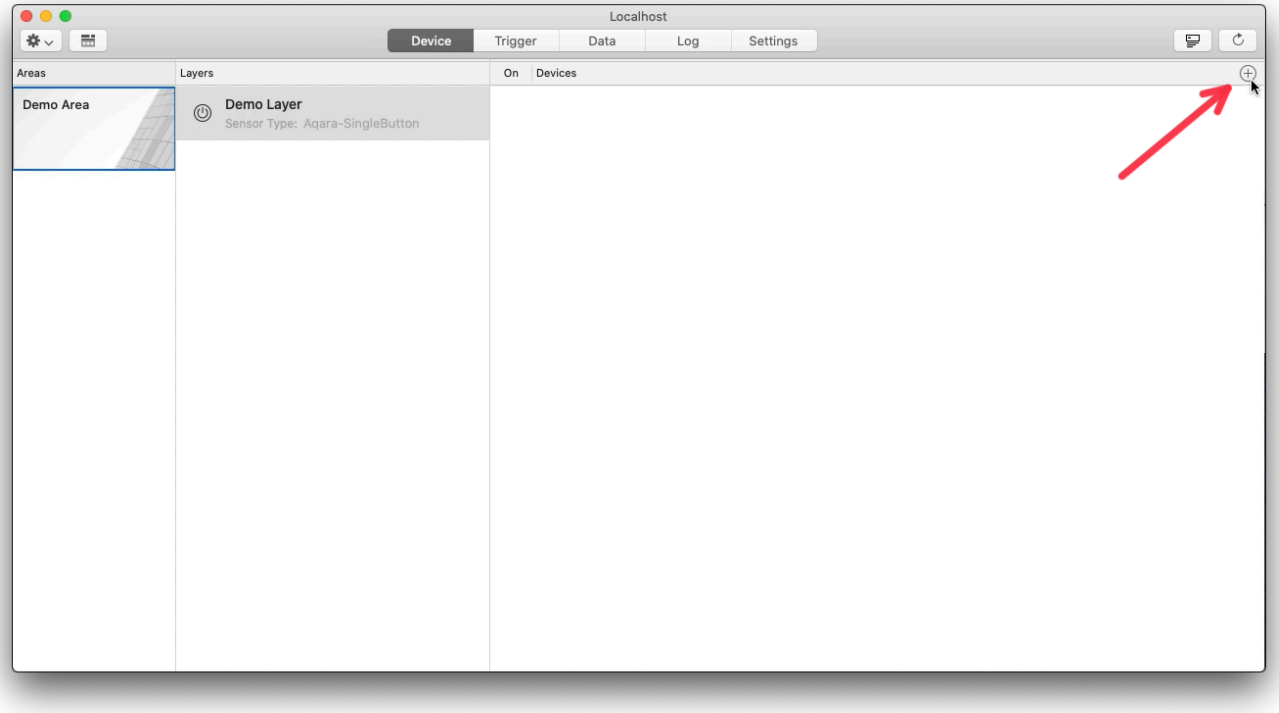
Once you have added all your sensors, you can close this view and add an Area where you want to add the sensors:



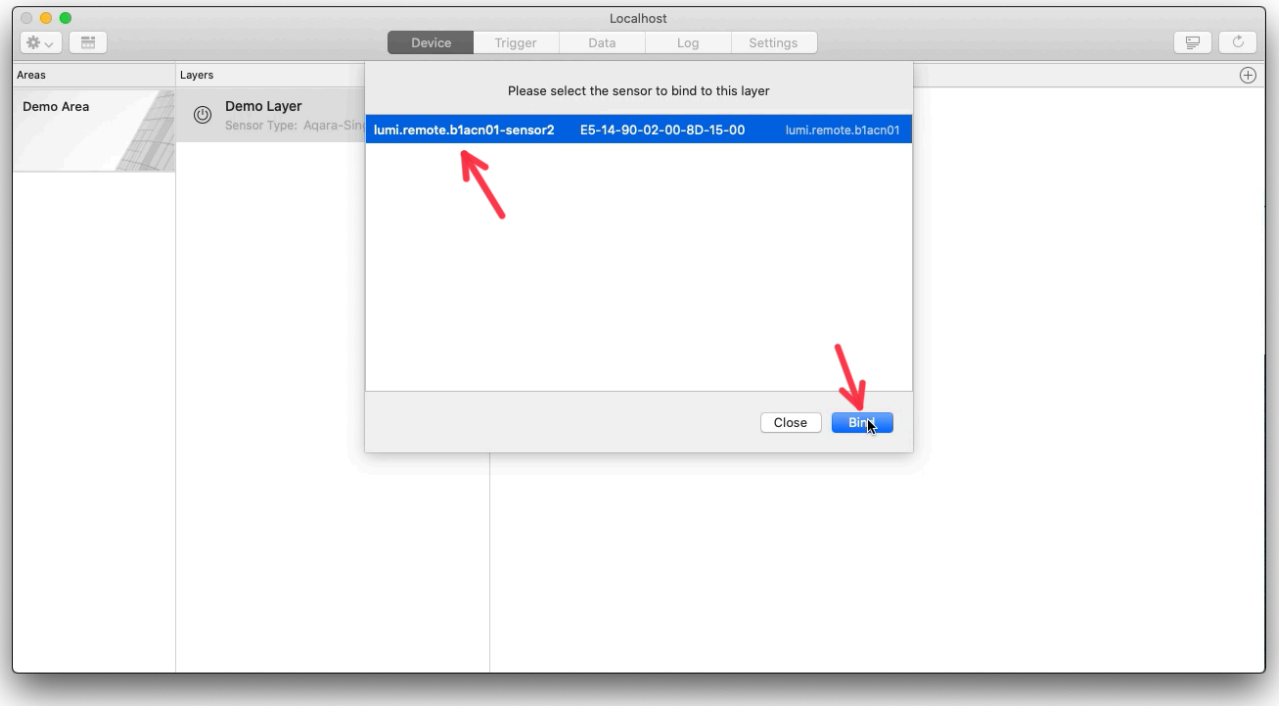
All data types are in a specific data layer. These layers have to match the type of sensors you are adding. In this case, we are adding a button, therefore you have to add a button layer. All sensors in this layer have to be buttons of the same type because the layer defines the character of the incoming data.



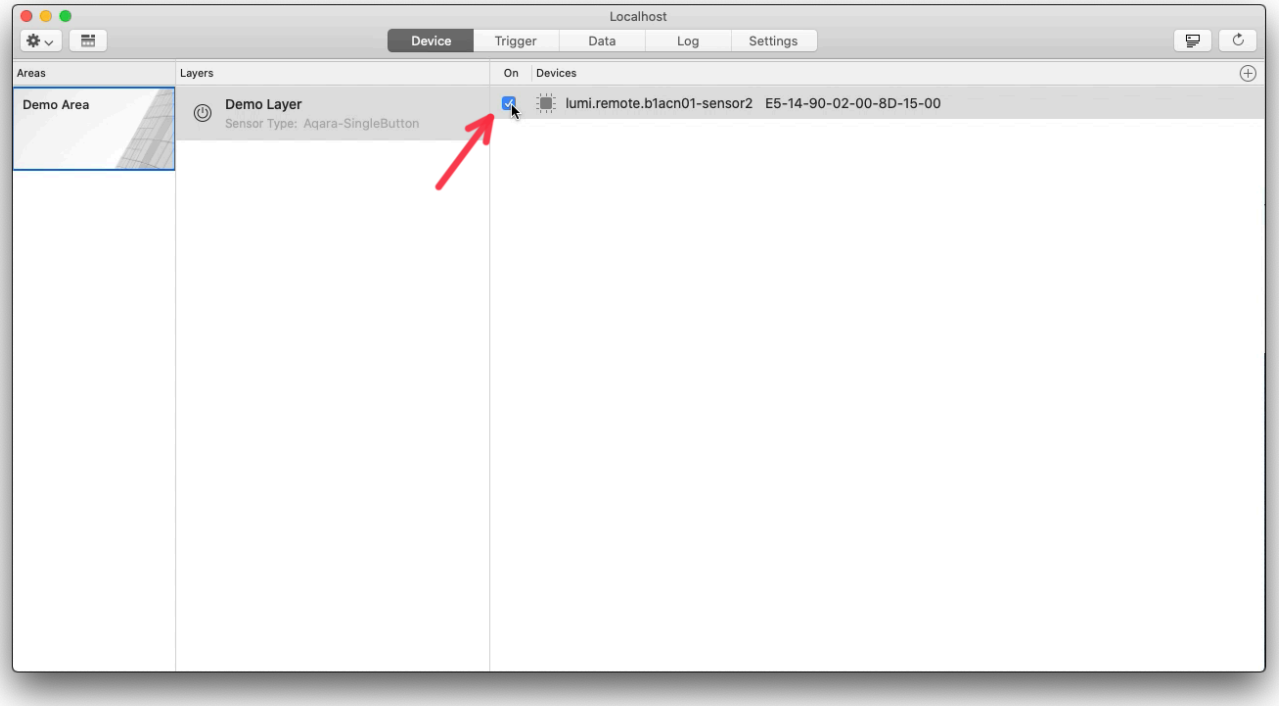
Now add the sensors you have previously paired to this layer:



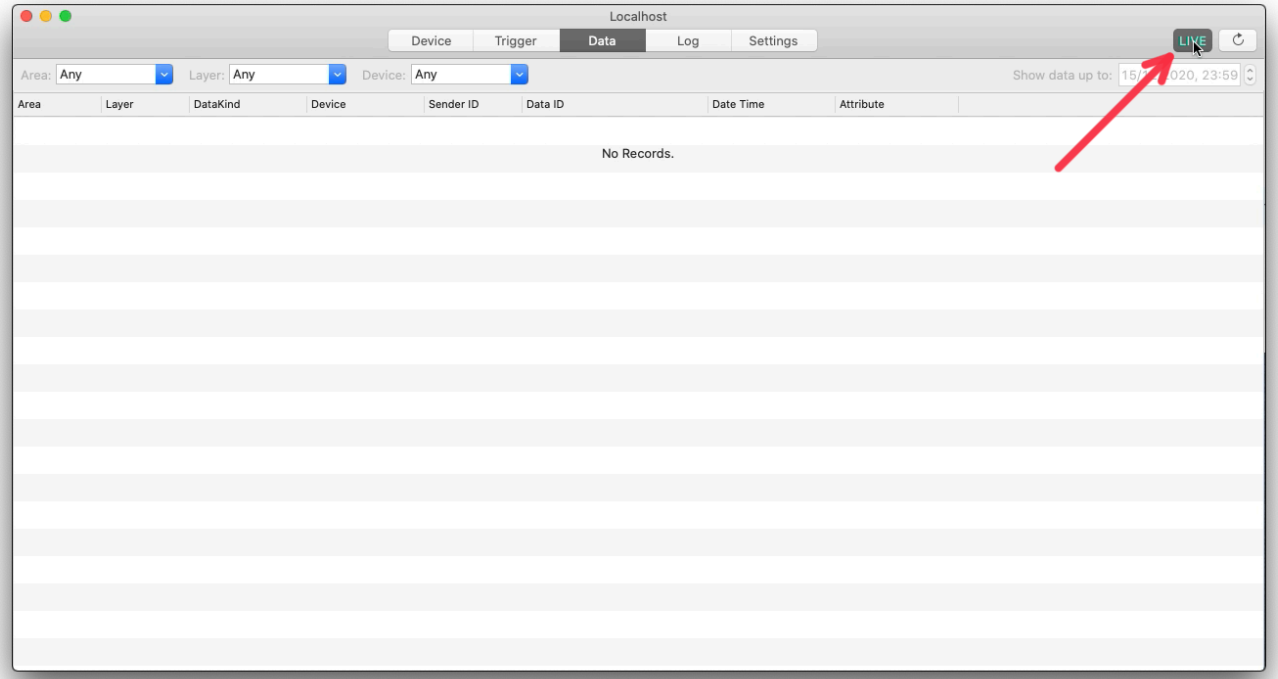
Ensure you are adding the right type to the right layer:



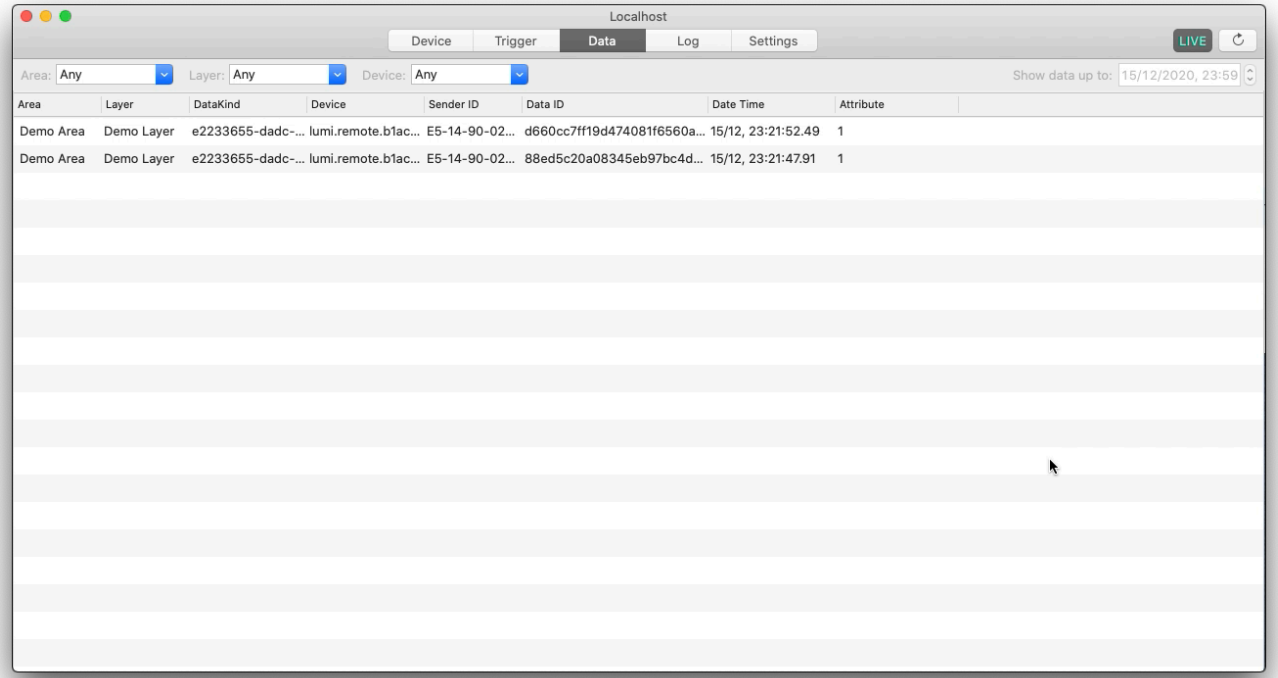
Now enable the sensor device:



At this point the data is coming in and being processed by Gravio. You can check that by opening the Data tab and hitting the “live” button to see the incoming data in real-time:



Incoming sensor data will create new lines in this view:



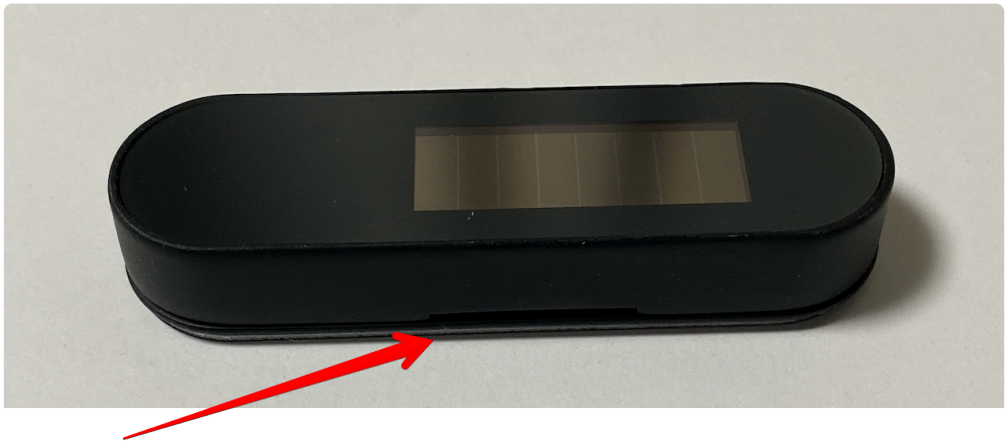
Congratulations, you have now connected sensors to Gravio and the incoming data is ready to be processed in triggers and actions.

6.1.1.1.1. Adding an EnOcean Sensors

We also rent out EnOcean energy harvesting sensors, that are getting powered by the energy surrounding them, such as solar panels, or by kinetic energy, e.g. when buttons are pushed.

We have two types of sensors:

1. The temperature sensors, identifiable by the small slot on the side:

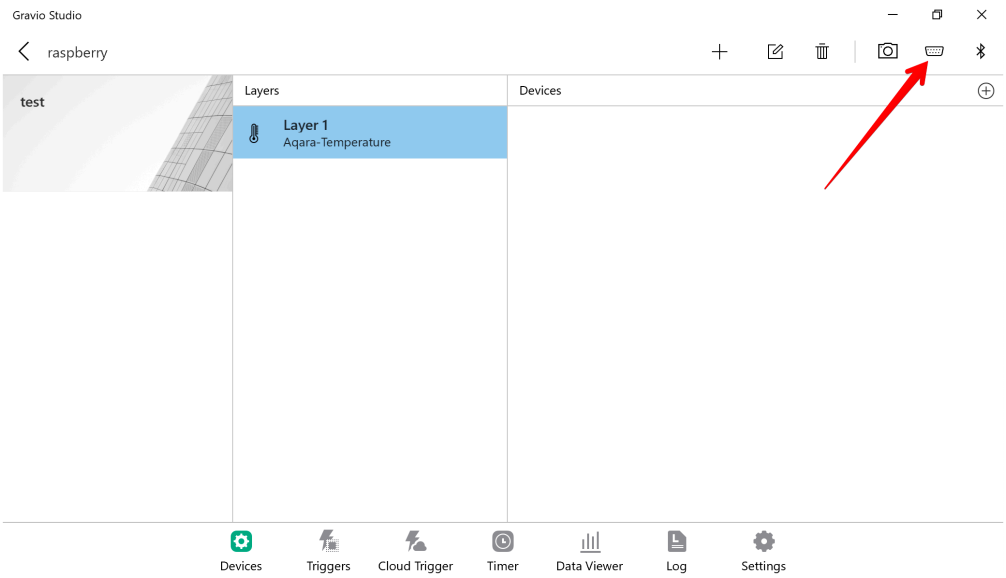


2. The door/window open/close sensor that comes with a magnetic counterpart:

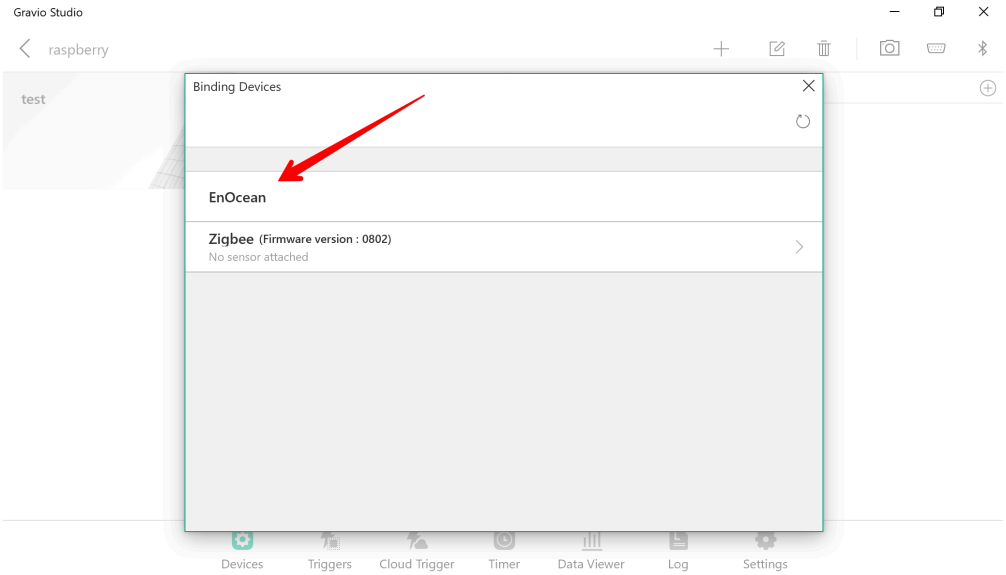
Note that due to frequency regulations, these sensors come in localised variants.

When using these Gravio sensors, please choose the layer type `Gravio-Door` or `Gravio-Temperature`. If you are using the devices in Japan, please use the same layer types but suffixed with `-JP`.

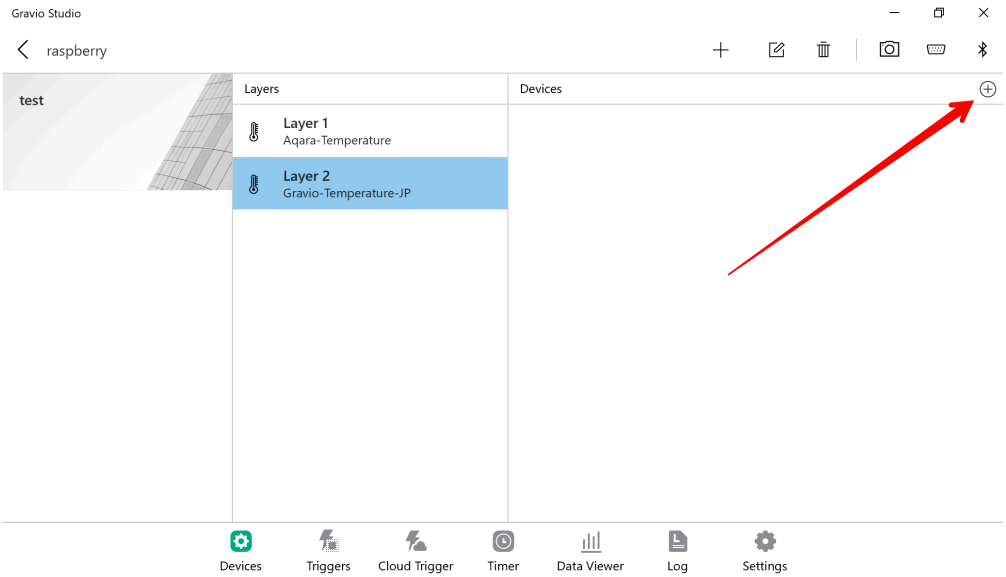
1. Once you have added the layers, click on the **Serial Interface Settings** button to open the dialogue and connect and pair the EnOcean USB Dongle with the sensors:



2. If your EnOcean Dongle is connected correctly, you will see it in the following list:

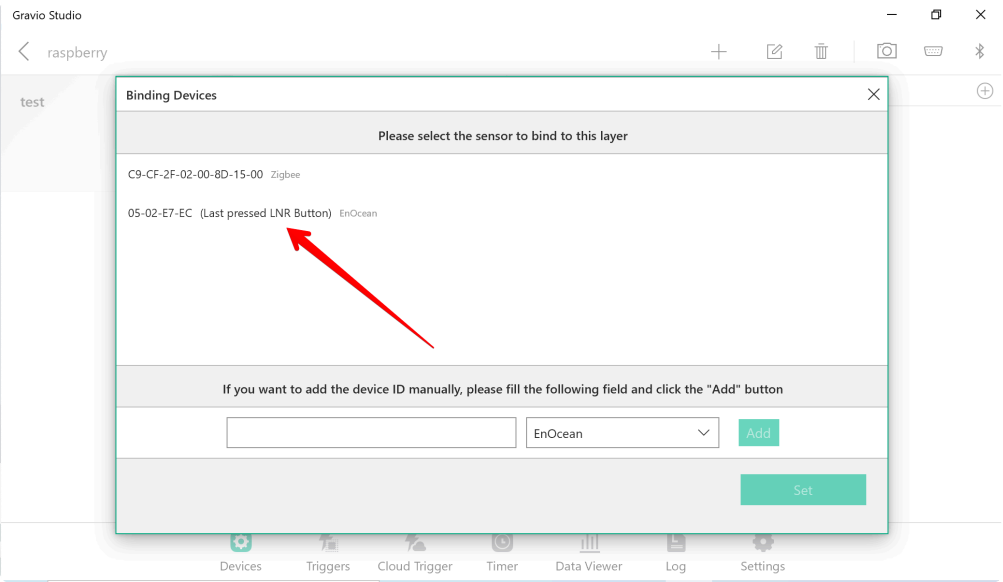


3. Once you have confirmed the receiver is correctly connected, close the window and add the sensors to the layer:

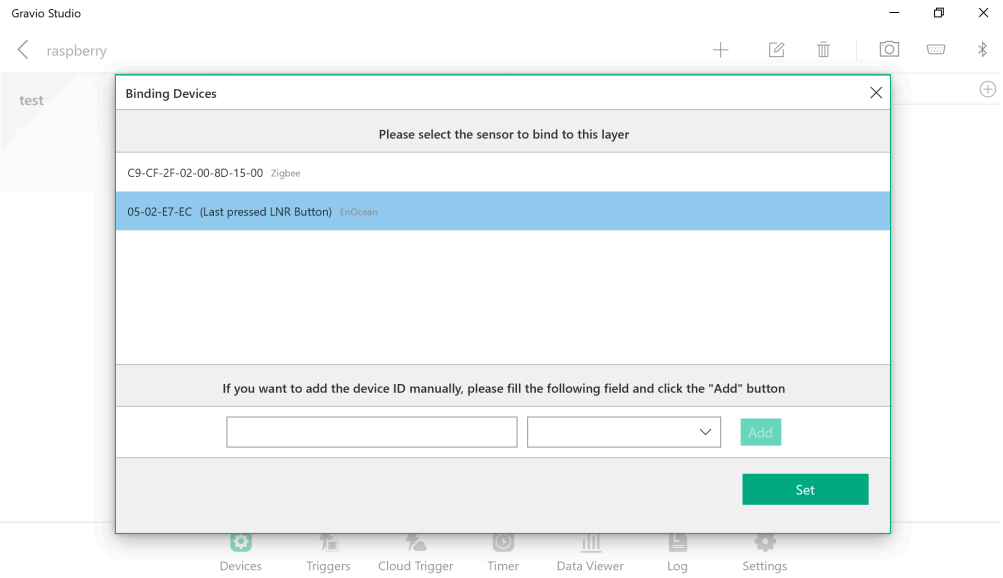


4. At this stage press the **learn** button on the sensor:

5. This will show the sensor ID in the list of sensors:

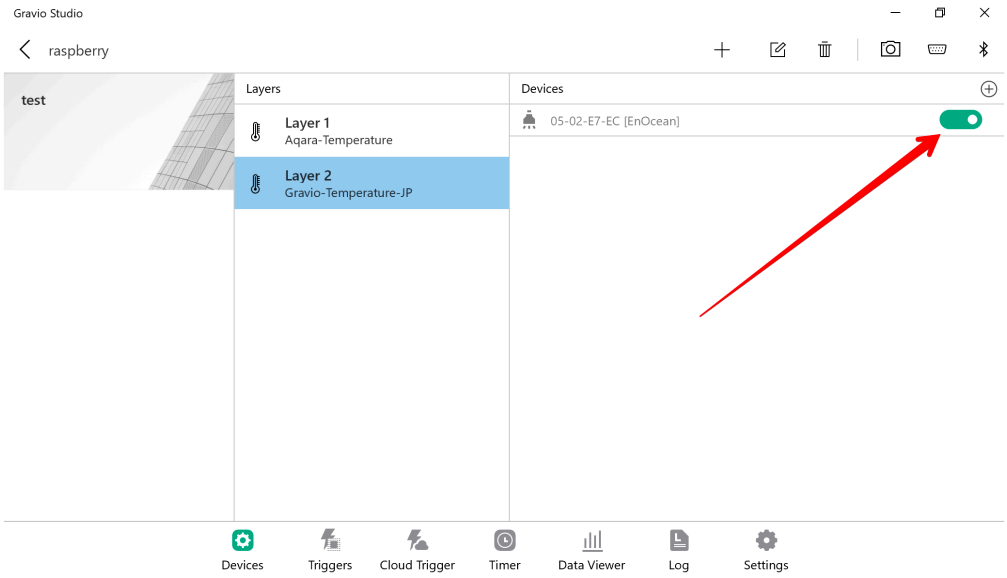


6. Select the sensor you want to connect to this layer and press “Set”:



This concludes the pairing process and you can close the window.

7. Don’t forget to enable the newly connected sensor(s):



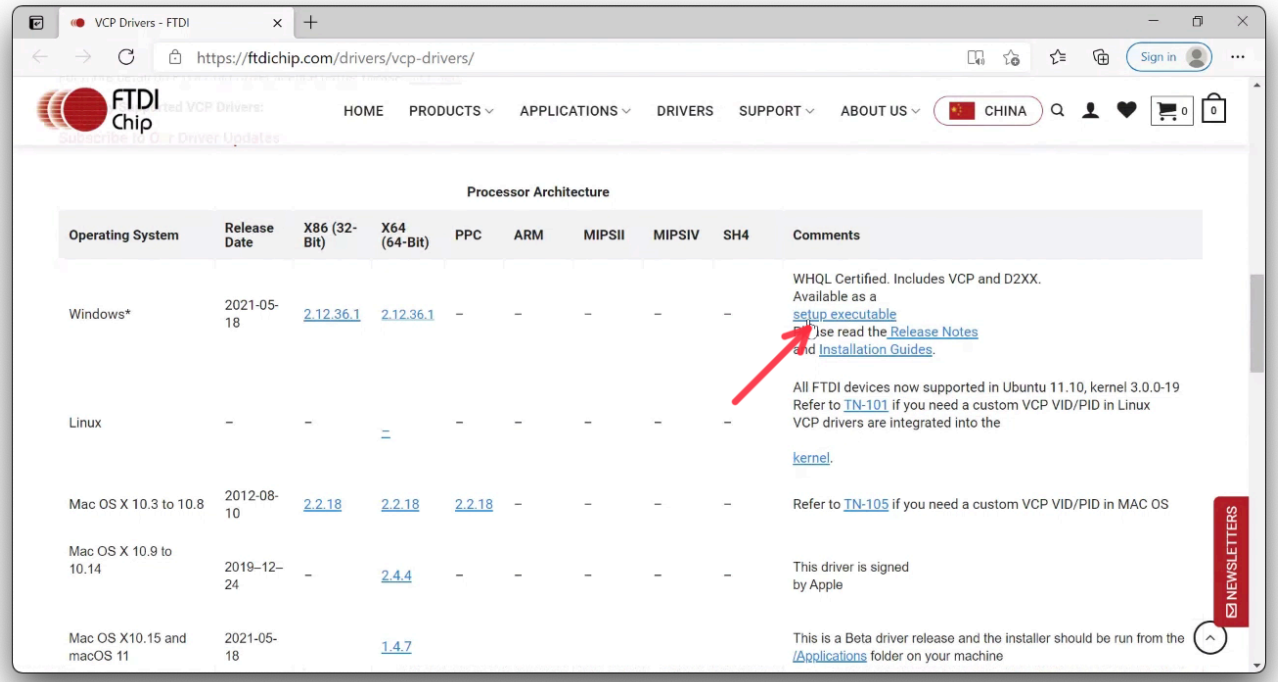
To check if the sensor data is received correctly, you can switch to the “Data Viewer” tab, select “Live View” at the top right and trigger the sensors.

6.1.1.1.2. Zigbee Driver Installation on Windows

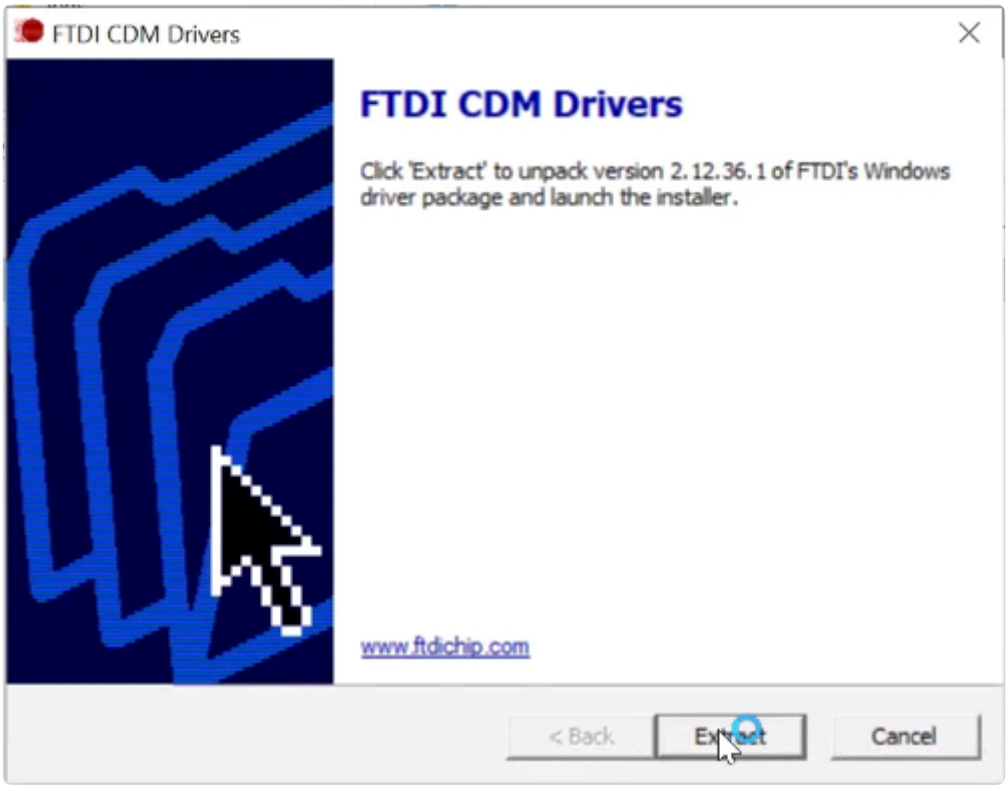
Once you have installed and started the GravioTools that come with your Gravio HubKit installation, the Windows drivers for your dongle are automatically installed in most cases. But if your drivers are not installed and you see an error in your “Device Manager”, please follow the instructions below to install the drivers manually:

Firstly, download the latest driver from FTDI website. The direct link is here: https://ftdichip.com/drivers/vcp-drivers/

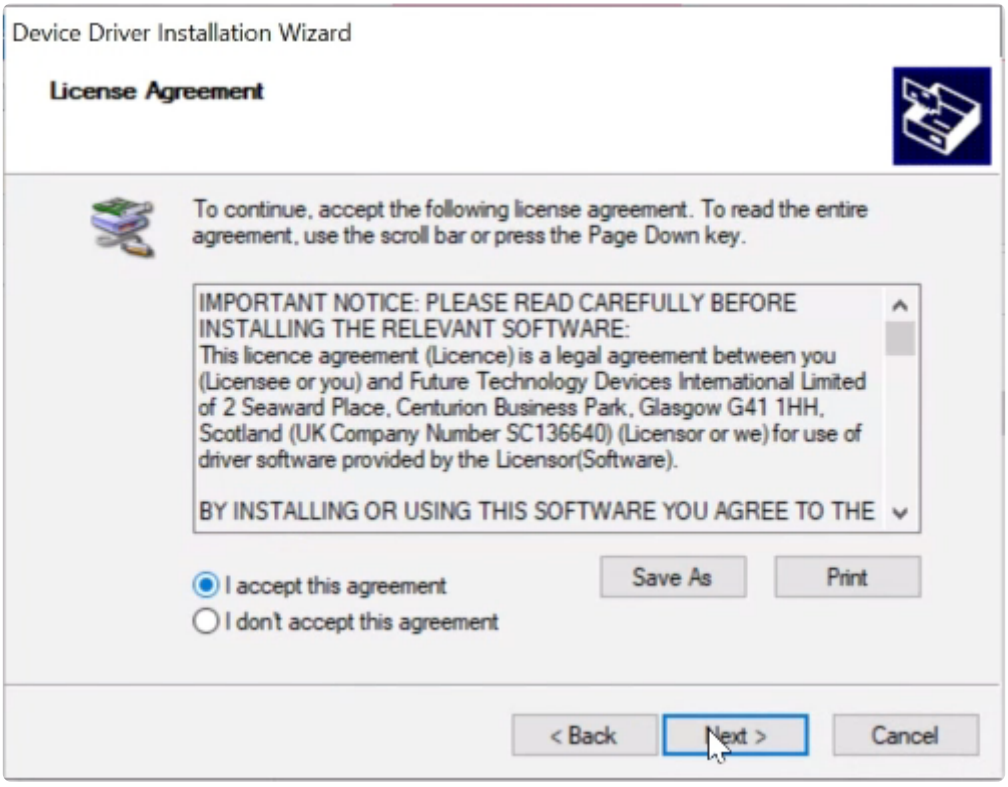
Download this file:



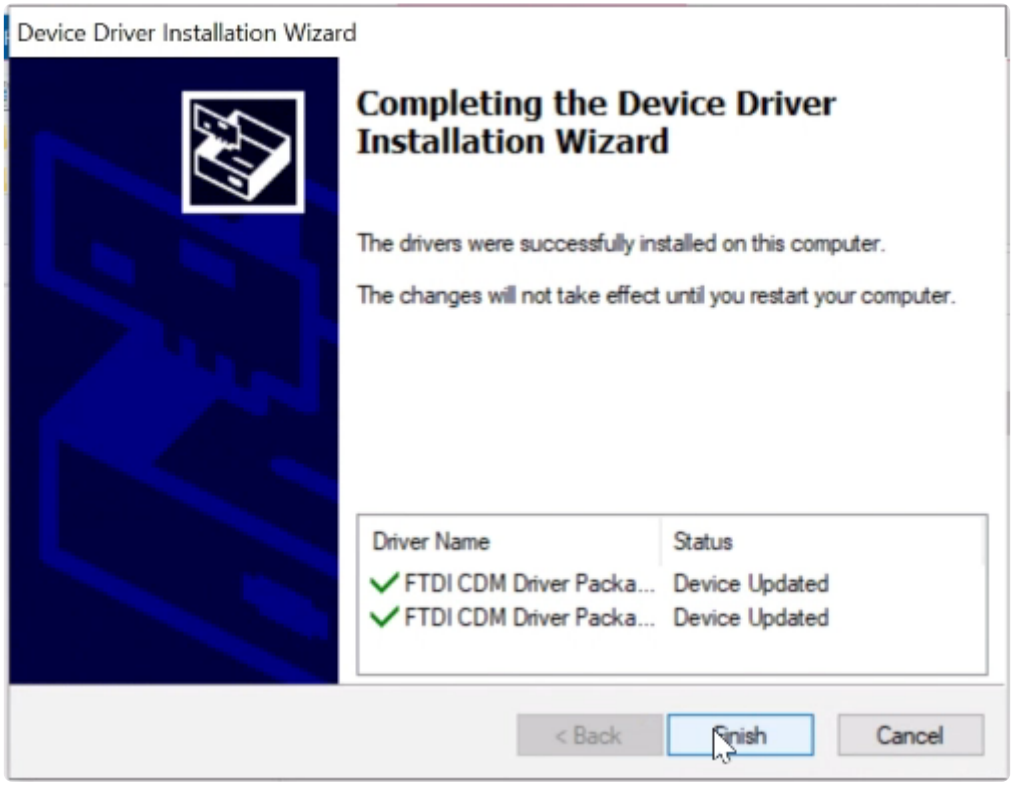
Once downloaded, extract the file and run the setup:



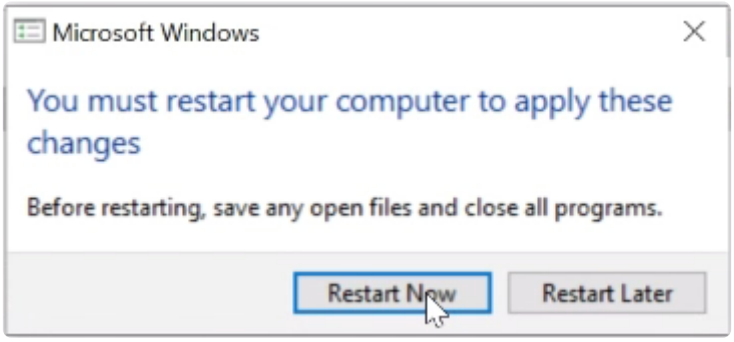
Install the driver.



Read the terms and accept them if you are ok with them.



Confirm that the drivers have been successfully installed.

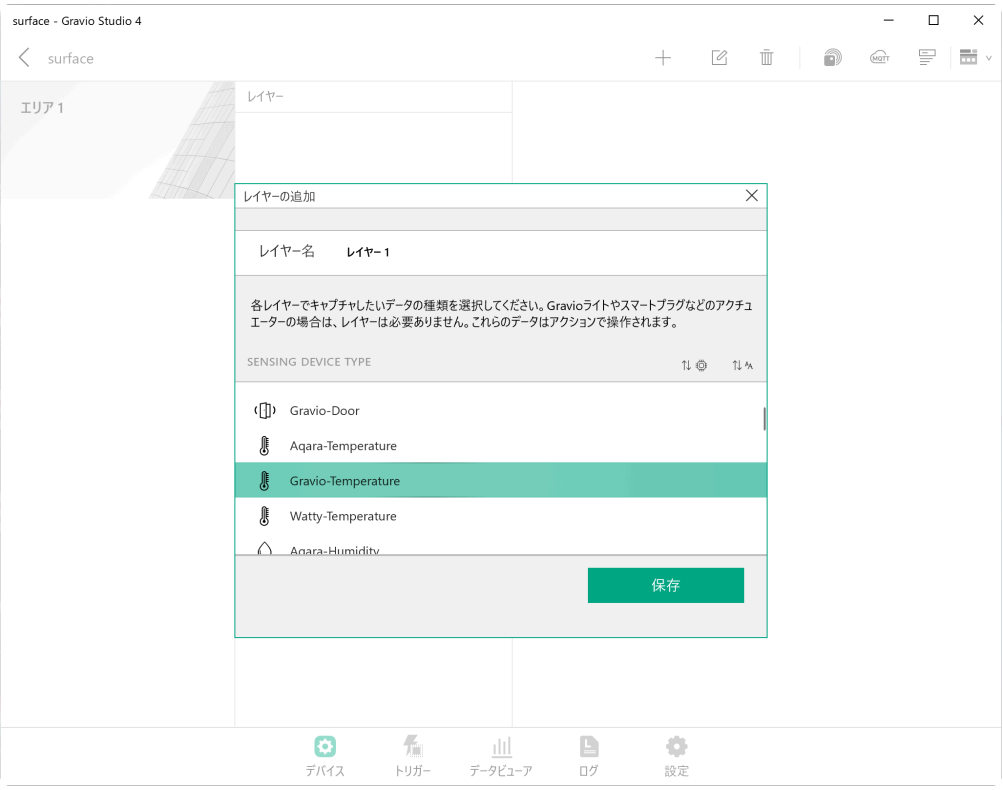


Reboot your machine.

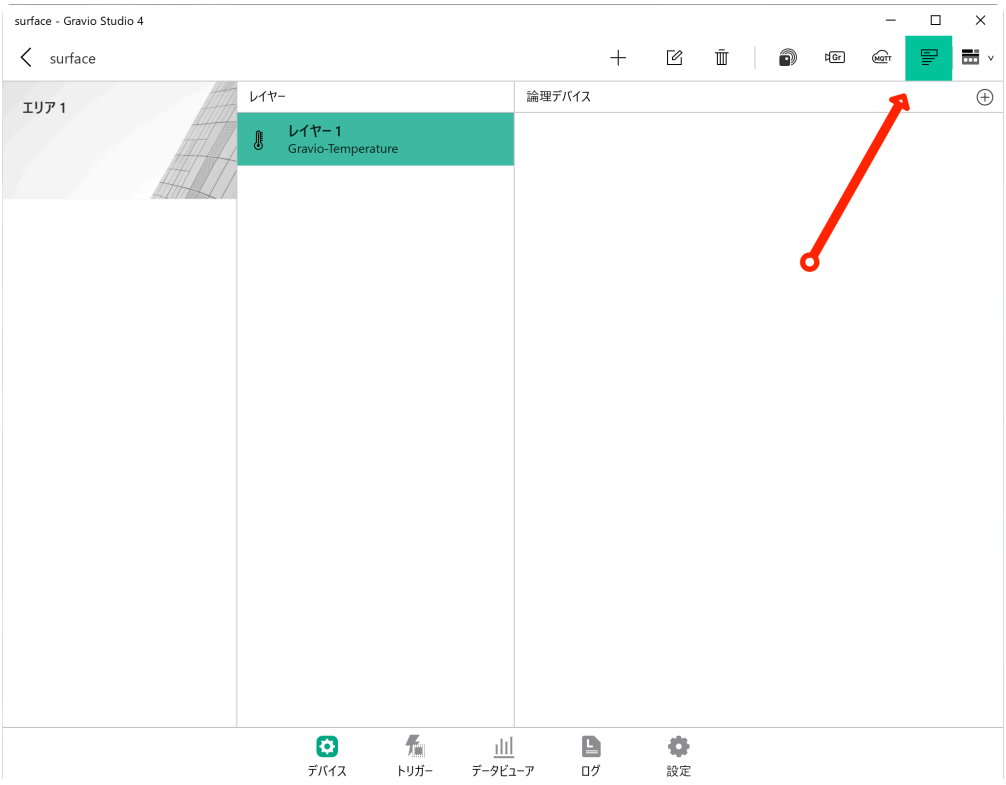
6.1.1.1.3. Adding EnOcean-enabled devices

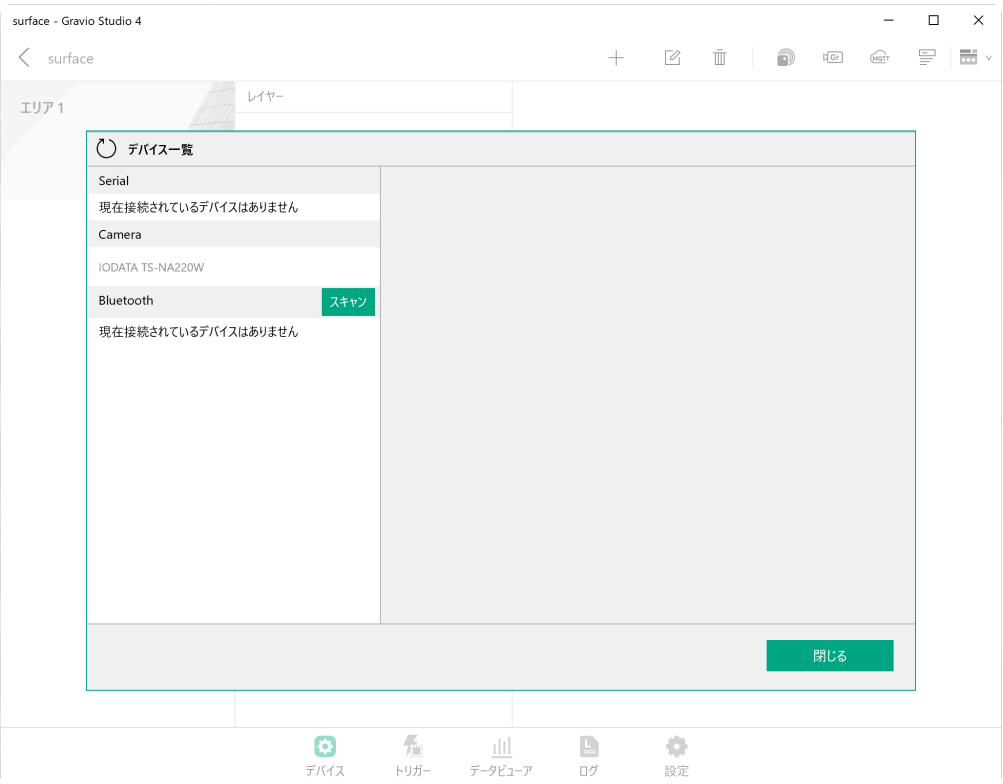
We have a Gravio USB dongle that can work with EnOcean sensor devices. On this page, we learn how to add those devices to Gravio

1. Under “Devices”, select the type of EnOcean sensor you want to use (DataKind) and add a new area and layer.

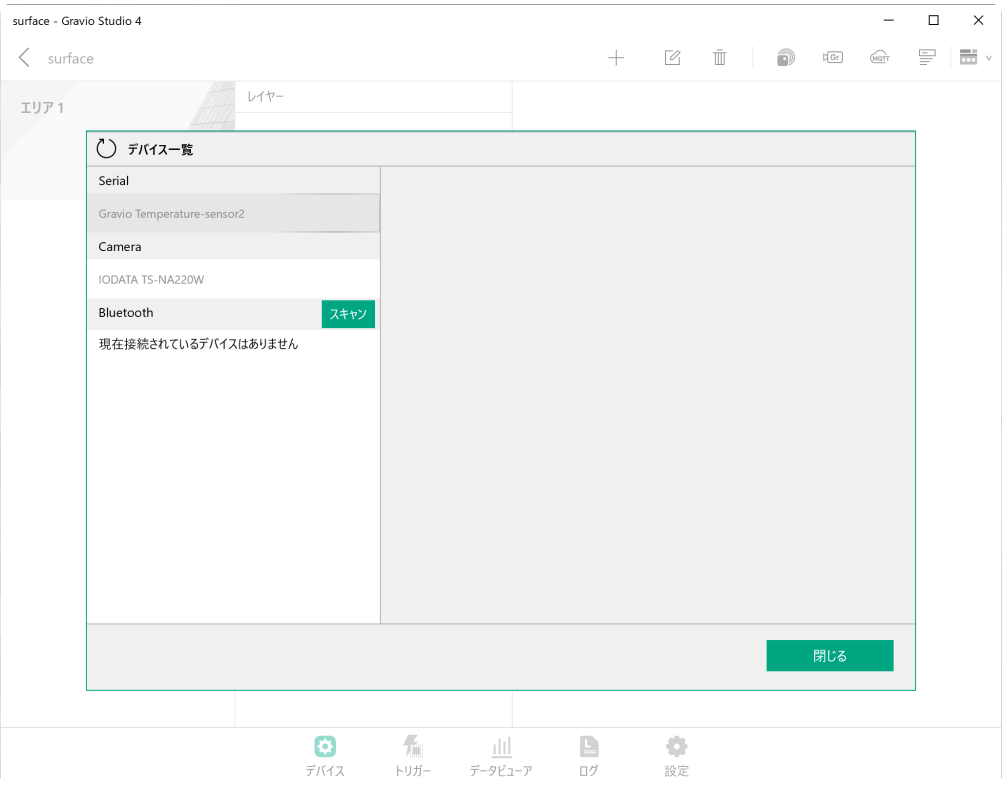


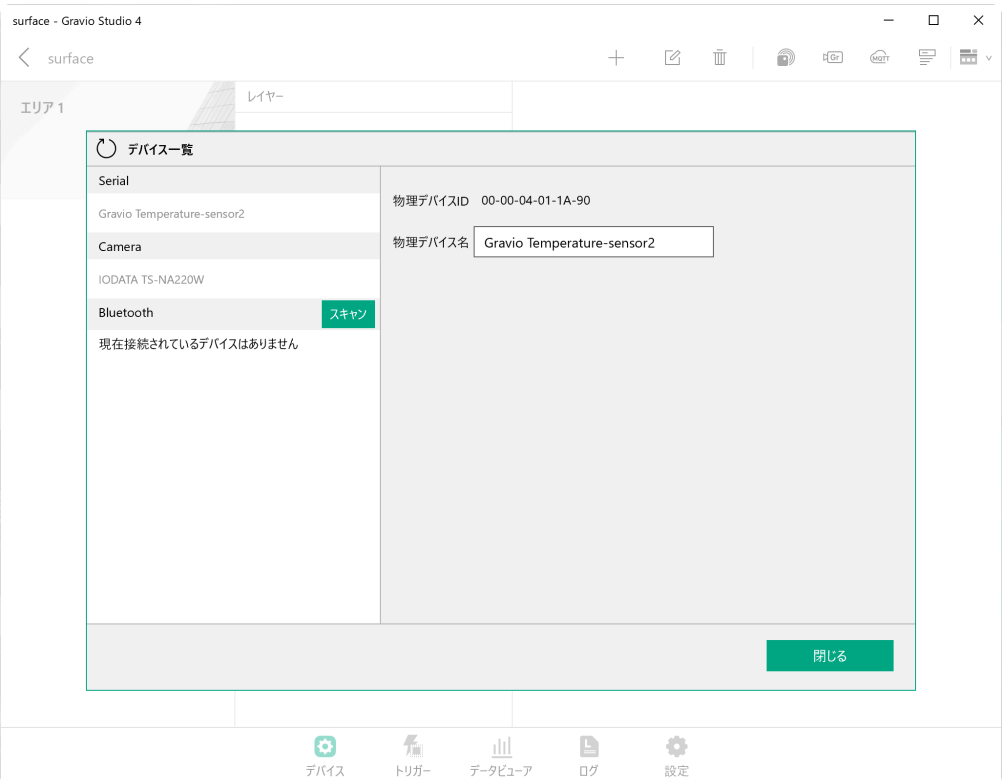
2. In order to receive sensor data from EnOcean-enabled devices, you must first pair the sensor with a sensor receiver. Press the button below to display the configuration screen for the sensor receiver connected to the serial port.



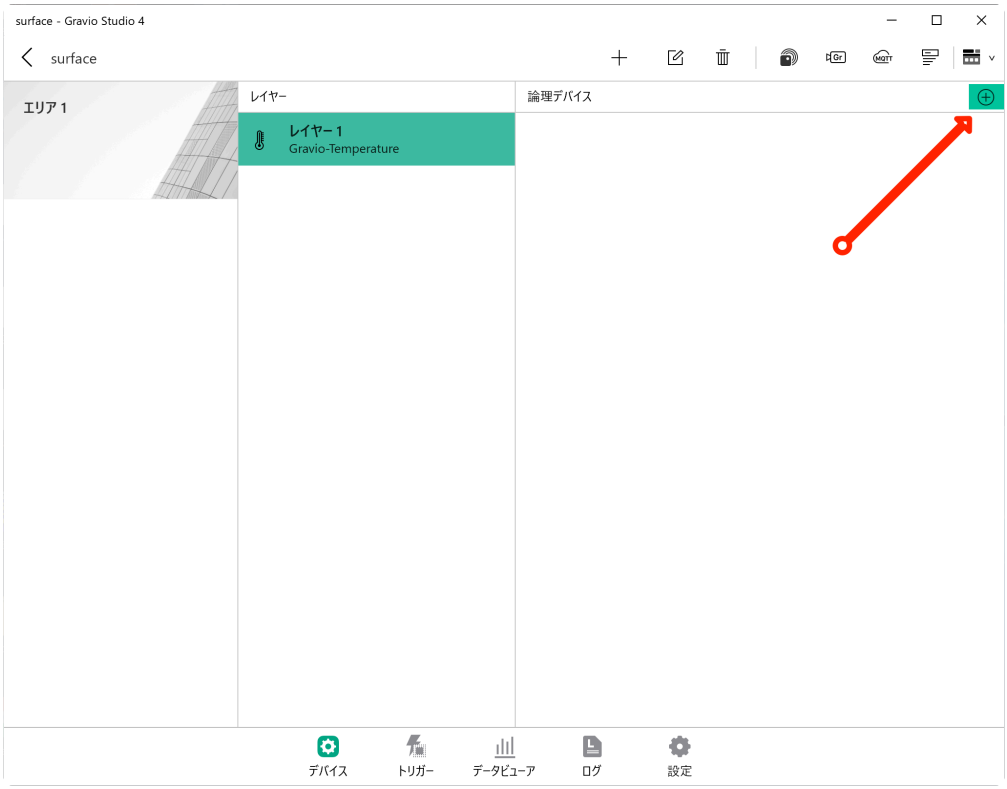


3. Press the button on the EnOcean-enabled device. After a while, the sensor will appear on the serial port.

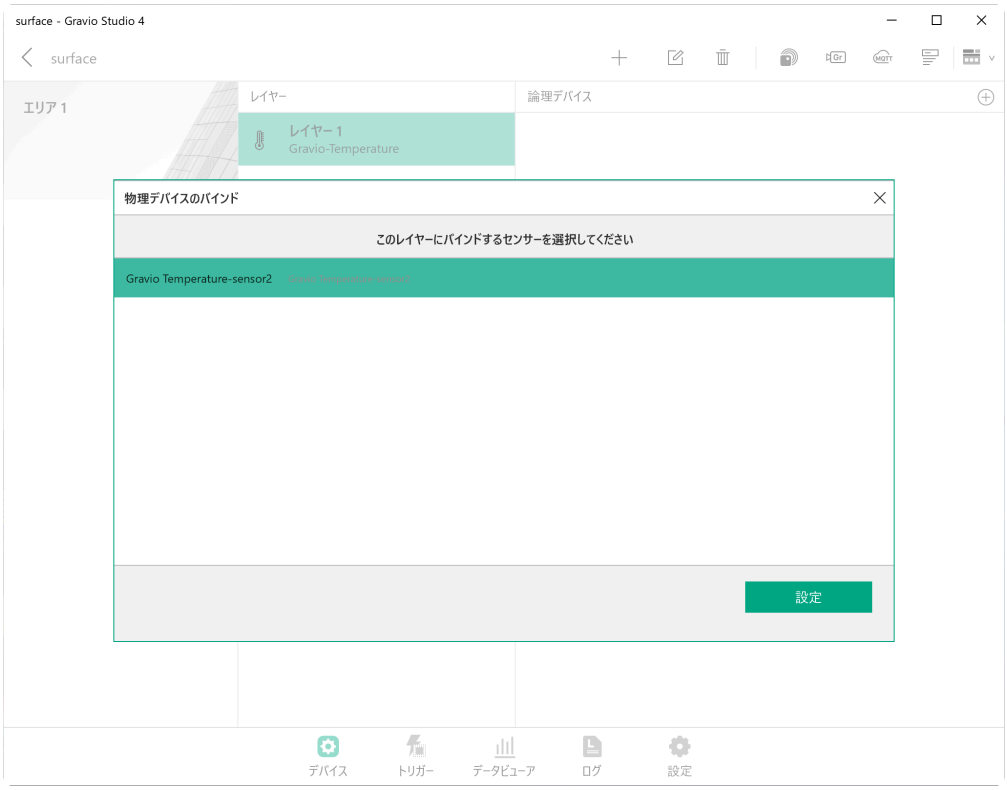




4. Next, click on the circled “+” symbol in the upper right corner of the window.



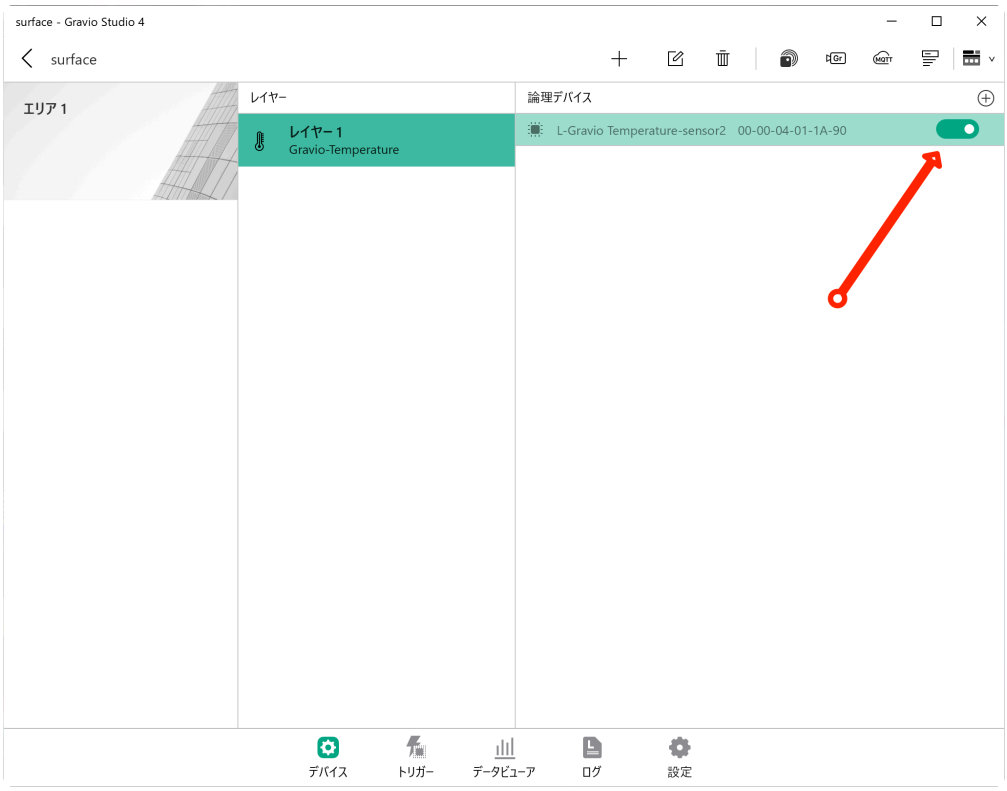
5. Select the sensor you want to connect to the layer and configure it.



When you have finished the settings, close this screen and.

6. You will see a screen similar to the one below. Finally, turn on the switch to receive data from this

sensor receiver.

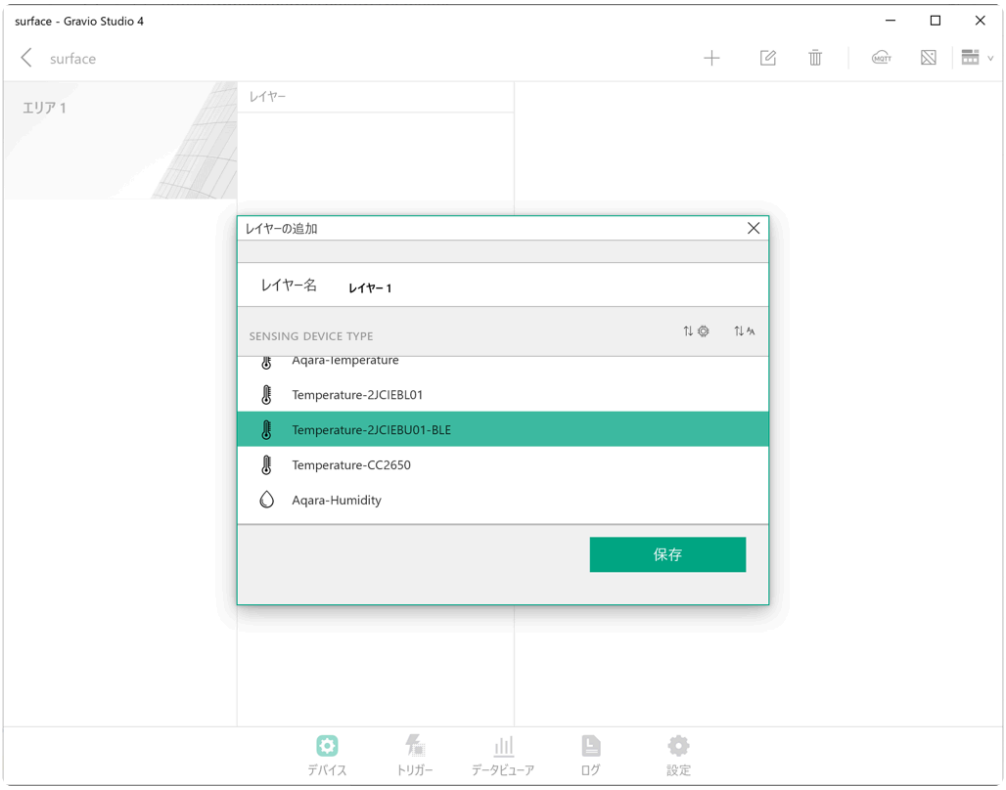


Now you can start receiving data from the sensor.

6.1.1.1.4. Adding a compatible Bluetooth device

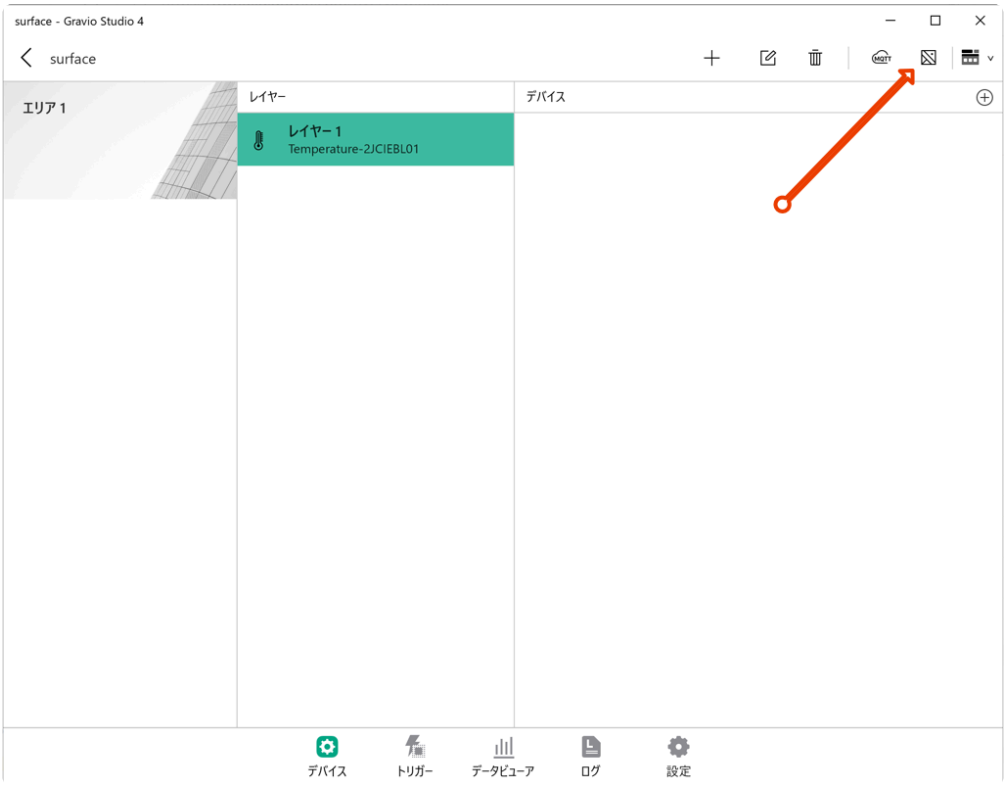
Use Gravio Studio to connect the sensor receiver and layer of a Bluetooth-enabled device.

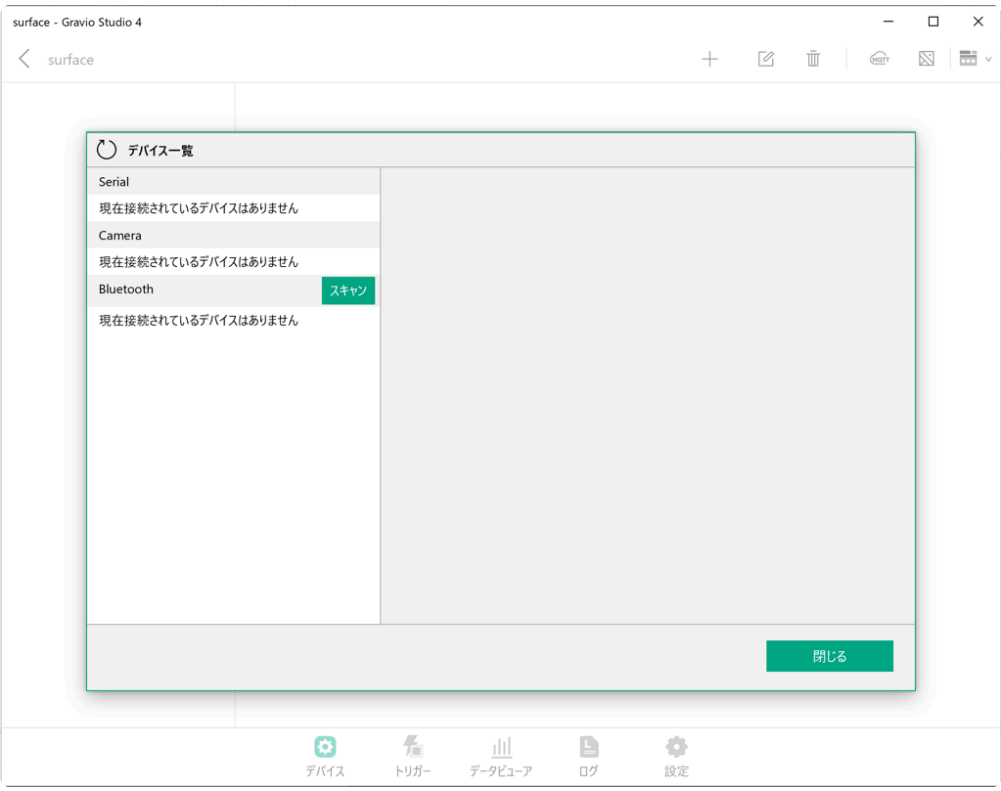
1. Under “Devices”, select the type of Bluetooth sensor you want to use (DataKind) and add a new area and layer.



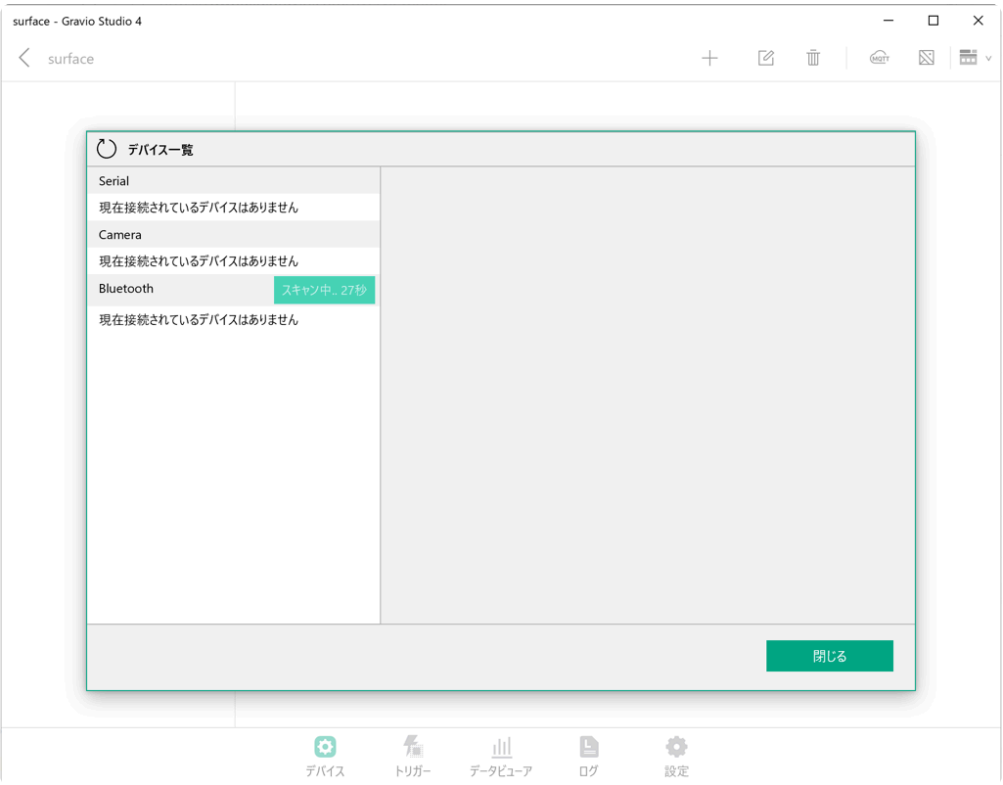
As a sample, use OMRON’s 2JCIE-BU01 to get temperature data.

1. To receive sensor data from a Bluetooth-enabled device, you must first pair the sensor with a sensor receiver. Press the button below to display the sensor receiver configuration screen.

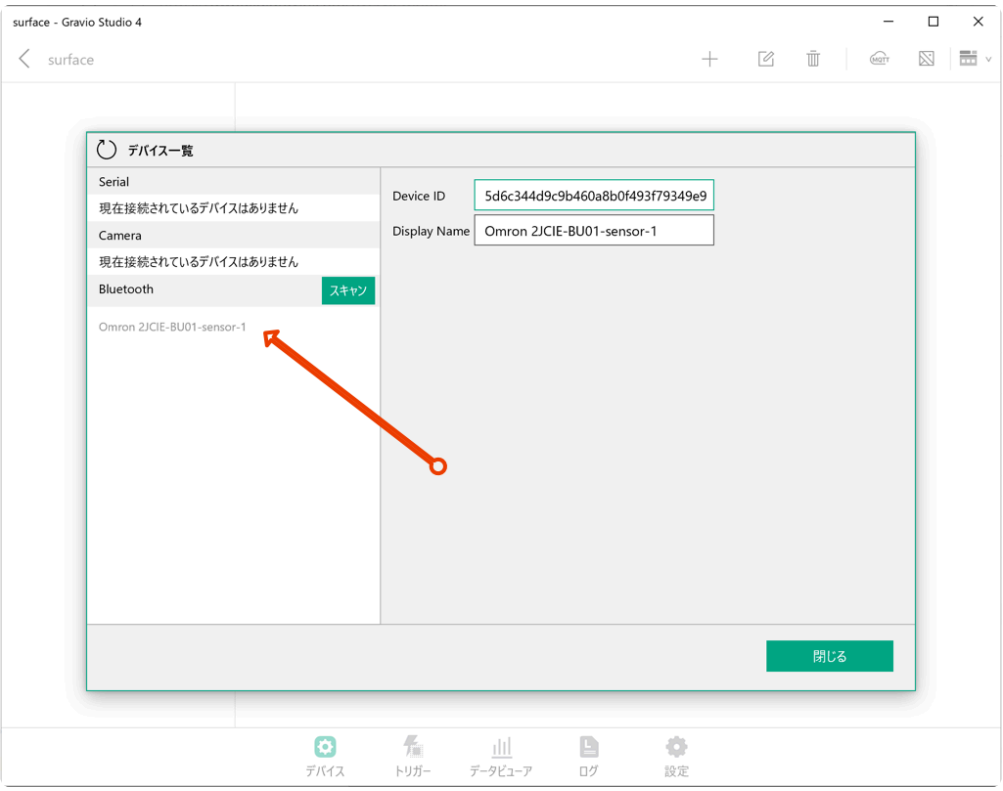




1. Click the right arrow in the Bluetooth row and press the Scan button. The scan mode is on for one minute.

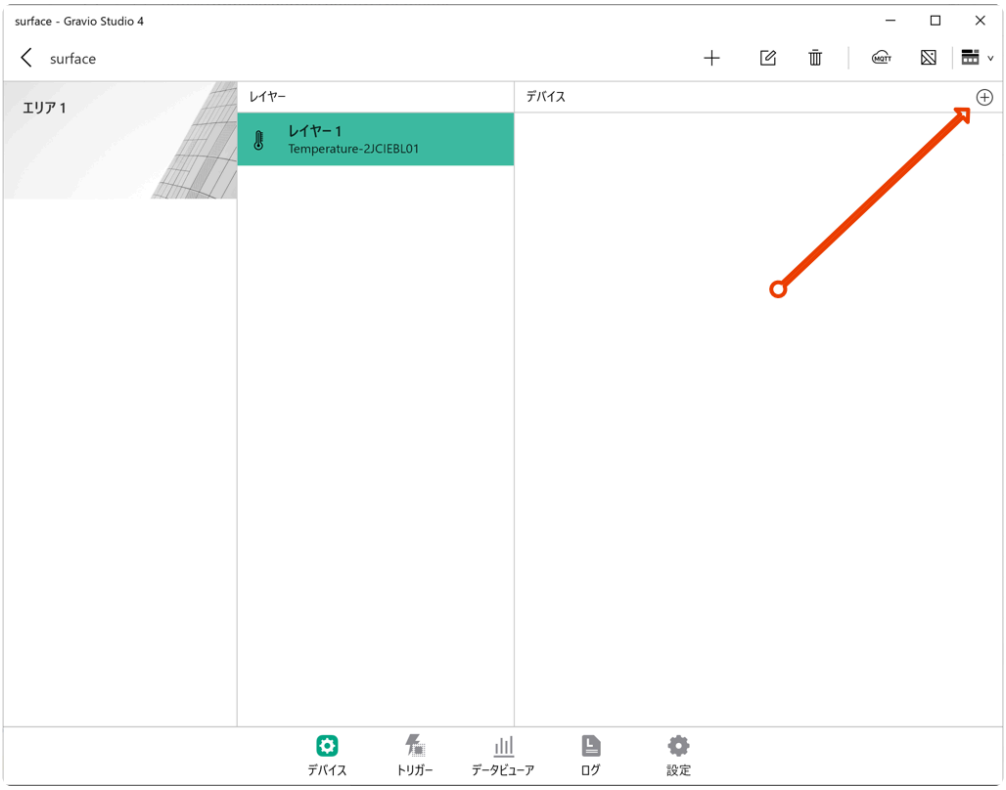


2. If a sensor is found while in scan mode, the following screen will be displayed:

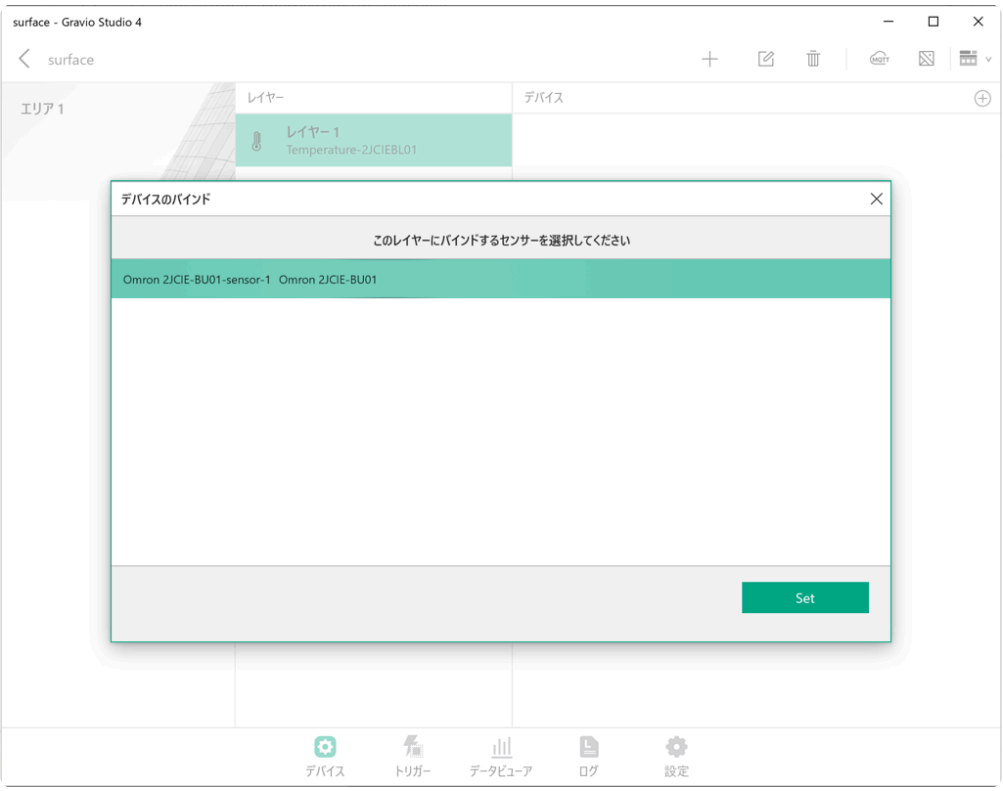


The sensor pairing process is now complete. Close the dialogue.

3. Next, click on the circled “+” mark in the upper right corner of the window.

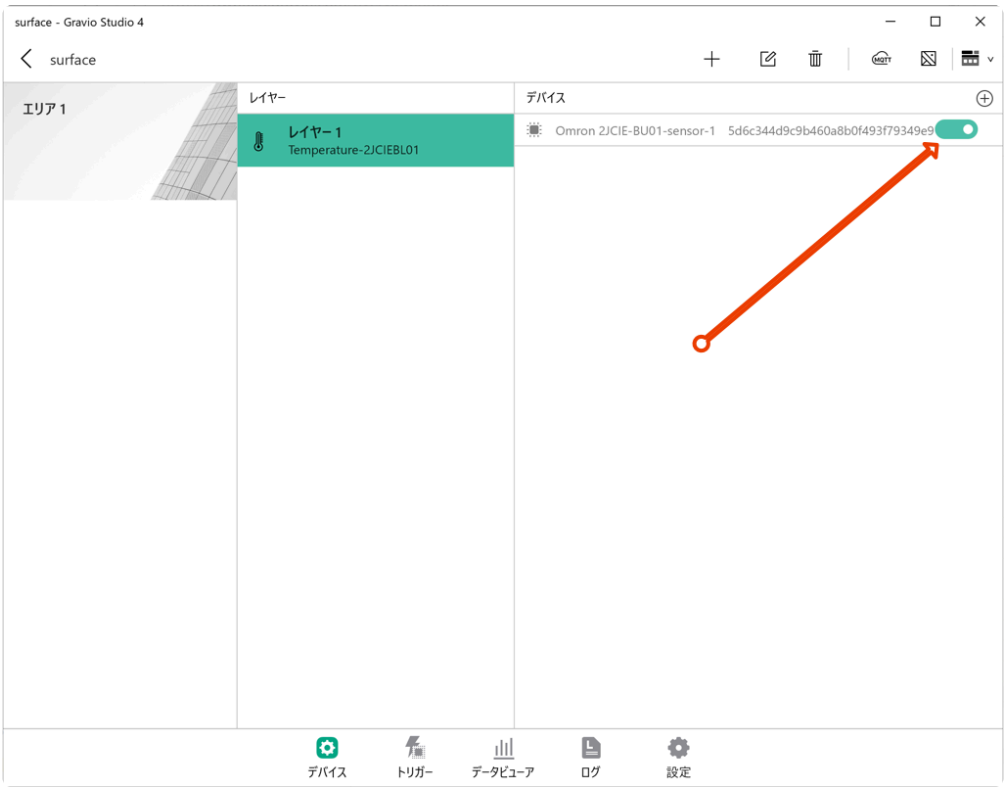


4. Select the sensor you want to connect to the layer and configure it.



When you have finished the settings, close this screen.

5. You will see a screen similar to the one below. Finally, turn on the switch to receive data from this sensor receiver.



This will start receiving data from the sensor.

6.1.1.1.5. Adding a barcode reader

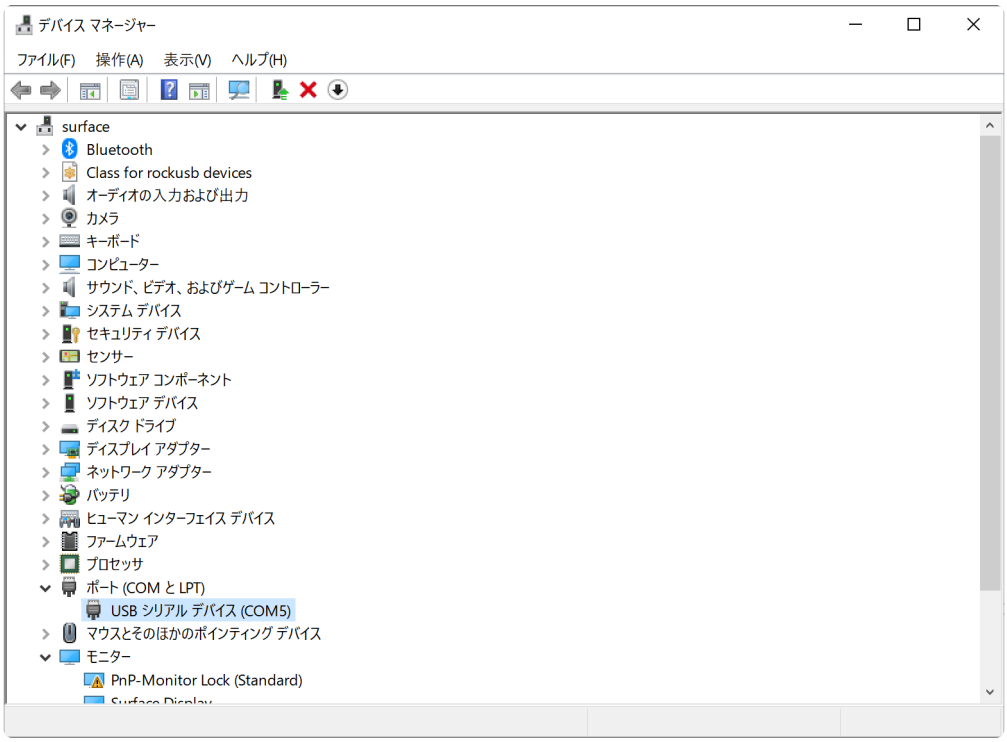
Using Gravio Studio to link a barcode reader to a layer

Gravio allows you to use a barcode reader (barcode reader with USB COM port emulation function) as a device that can input data as one layer.

Currently, there are three types of barcode readers that have been confirmed to work: BC-NL1100U, BC-NL2200U, and BC-NL3000U from BUSICOM, and NLS-FM430 from NewLand.

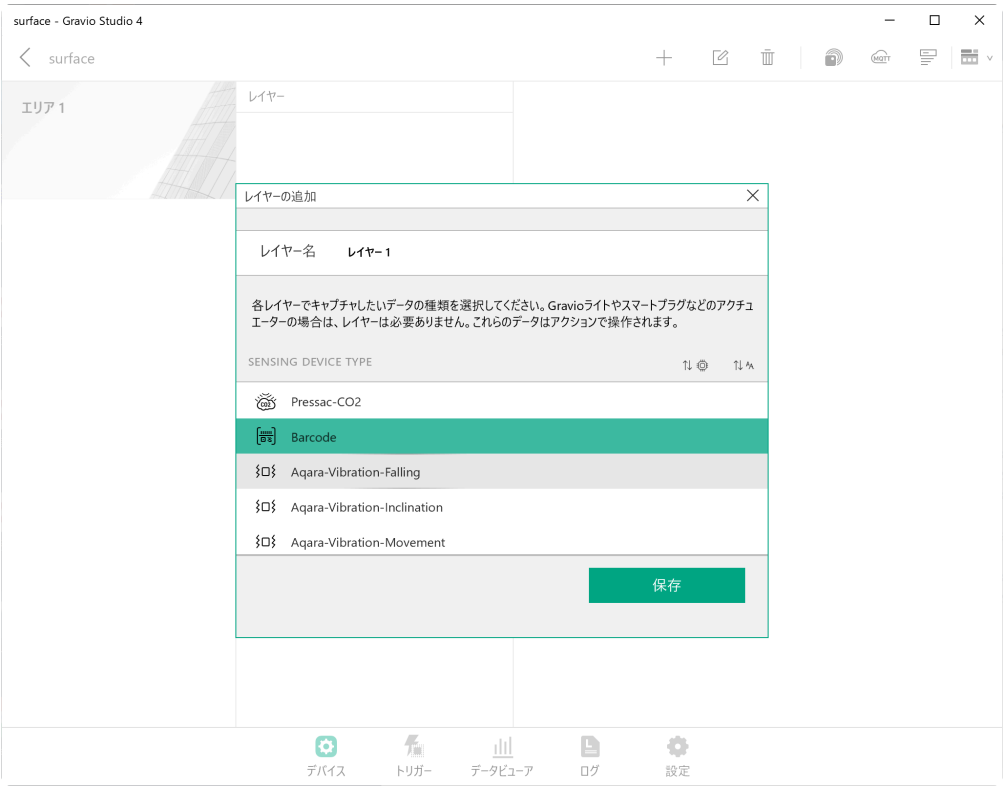
barcode reader connection check

To check the connection of the barcode reader, in Windows 10, check if COMx (COM5 in this case) is displayed in the device manager.



adding a barcode reader

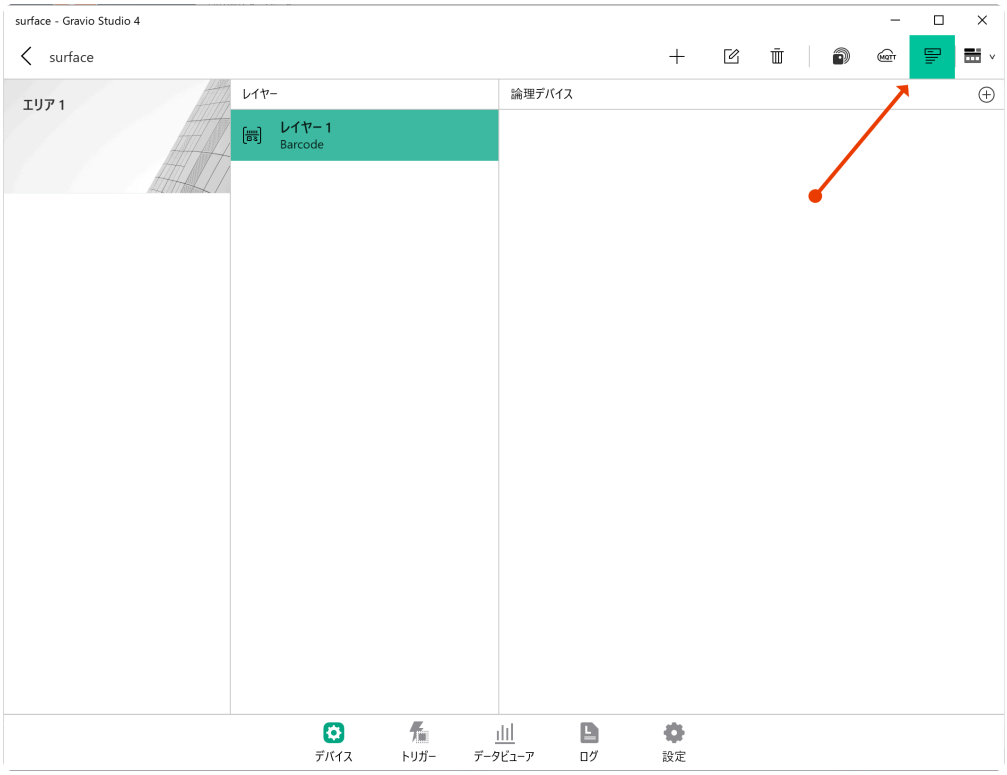
1. in “Device”, select the Barcode (DataKind) you want to use and add a new area and layer.

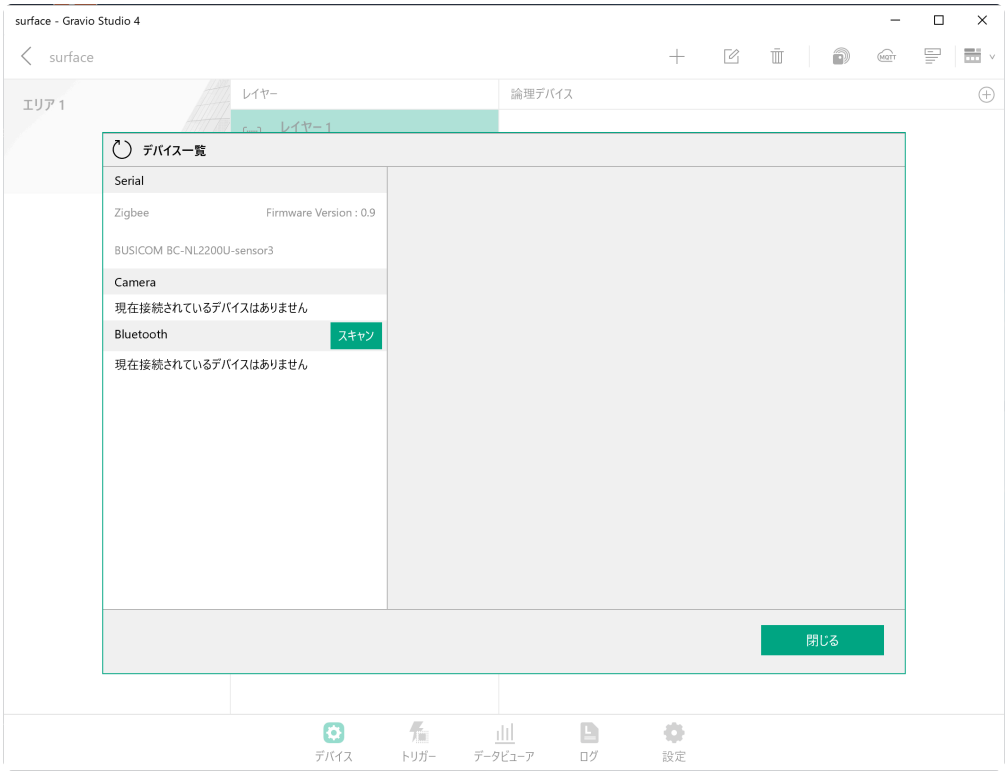


Here is an example where the barcode reader uses BC-NL2200U from BUSICOM to retrieve the barcode data.

Note: The BC-NL2200U can switch between the USB HID-KBW function, which is simulated as a USB keyboard input, and the USB COM port emulation function, which allows the host to receive data as if it were a serial port. For details, refer to the BC-NL2200U. For details, please refer to the BC-NL2200U manual.

To receive barcode data, connect the barcode reader to the USB port, and then press the button below to display the barcode setting screen.

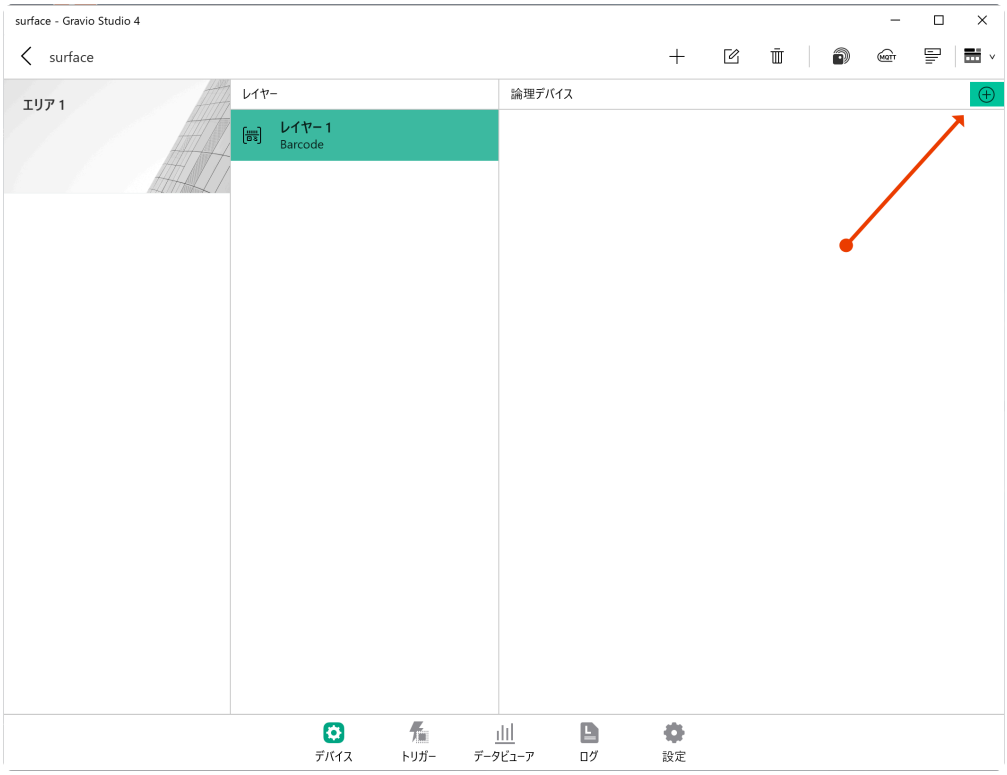




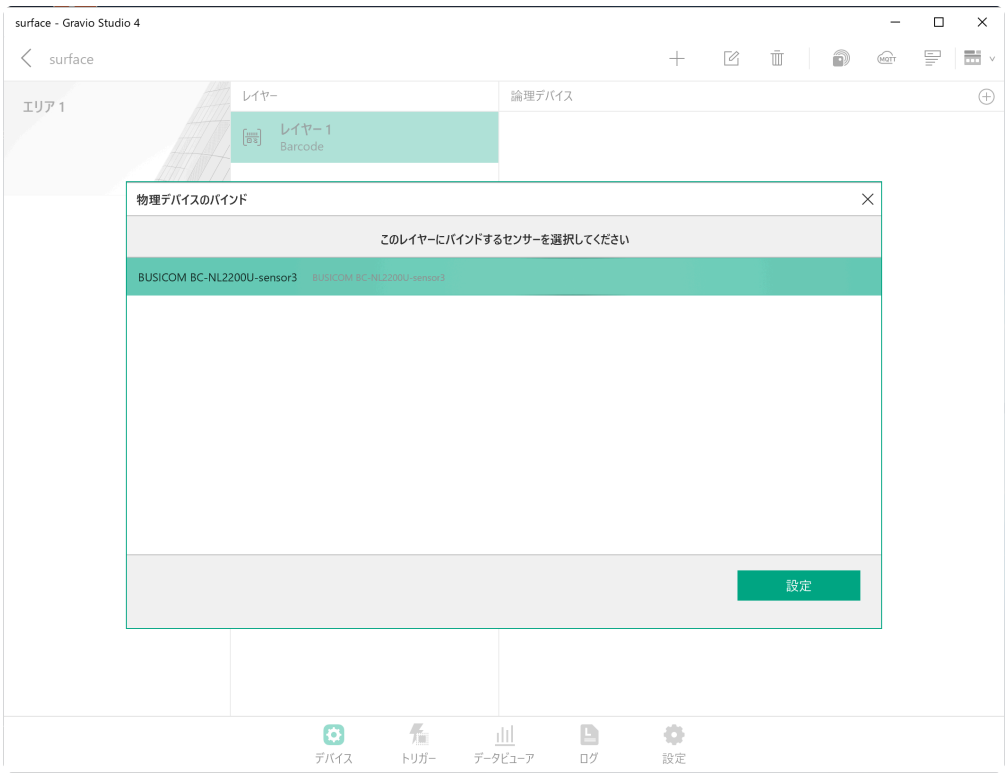
Note: If BUSICOM BC-NL2200U does not appear, please make sure that the barcode reader is properly connected to the USB port and restart HubKit.

If the BUSICOM BC-NL2200U is displayed, you are done. Close the dialog. 3.

Next, click on the circled “+” mark in the upper right corner of the window.

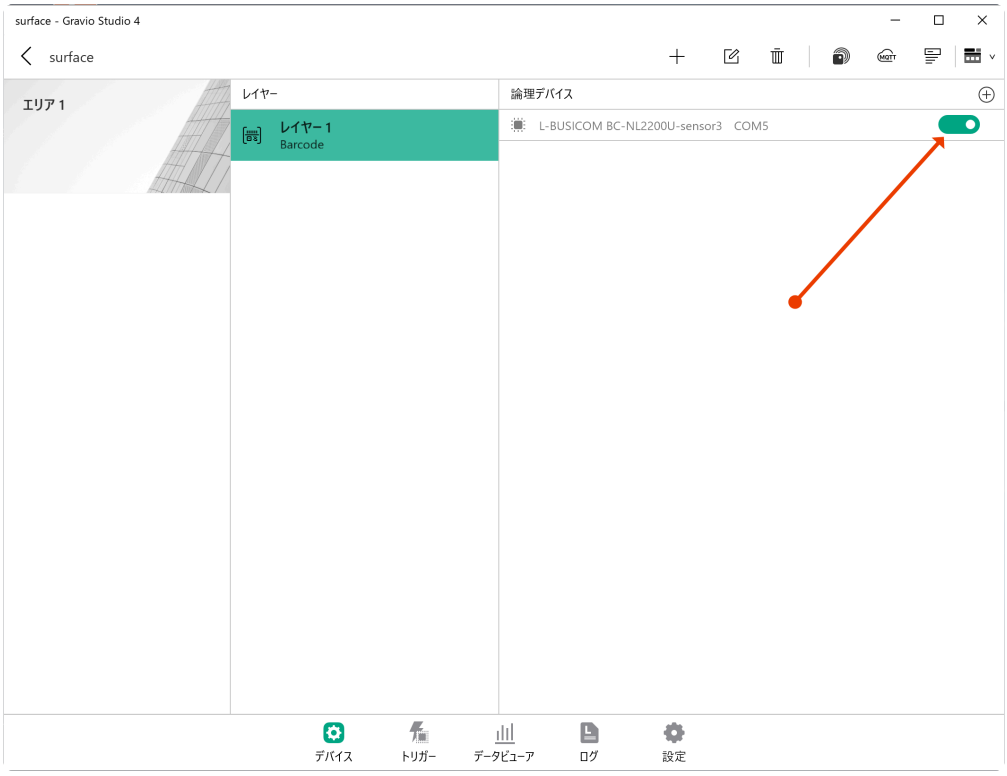


4. select the barcode reader you want to connect to the layer and configure it.



After finishing the settings, close this screen. 5.

You will see a screen like the one below. Finally, turn on the switch to receive data from this barcode reader.



Now you can read the barcode with the barcode reader, and it will start receiving data.

6.1.1.1.6. Adding GPS Devices

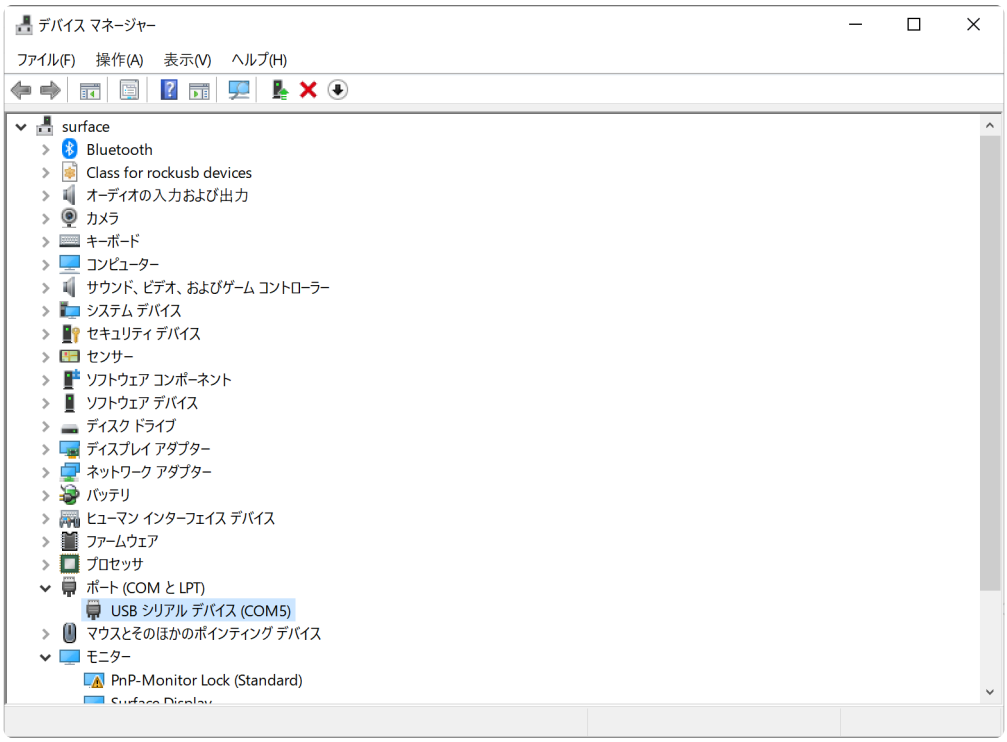
Using Gravio Studio to link GPS devices to layers

Gravio allows GPS devices ((colour-red) USB COM port emulation enabled GPS devices) to be used as a data input layer similar to other data input devices.

Please visit the officially supported sensor list to find out which devices are supported.

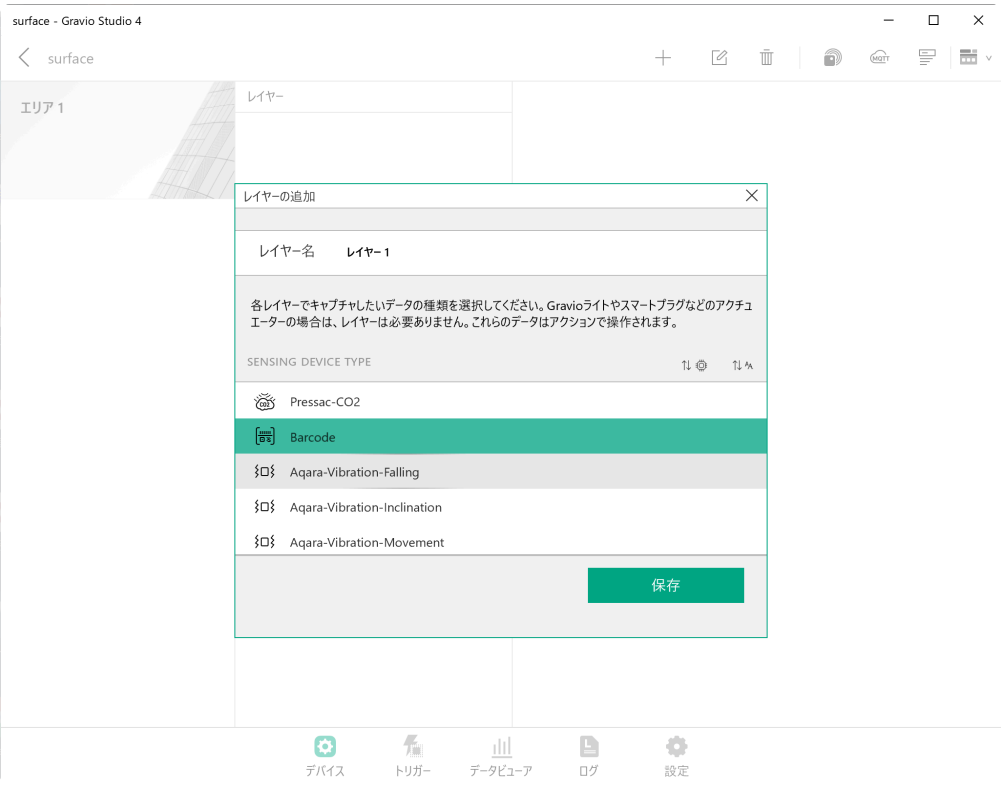
Check the connection of the GPS device

To check the connection of the GPS device, in Windows 10, check whether COMx (in this case COM5) is displayed in the device manager.

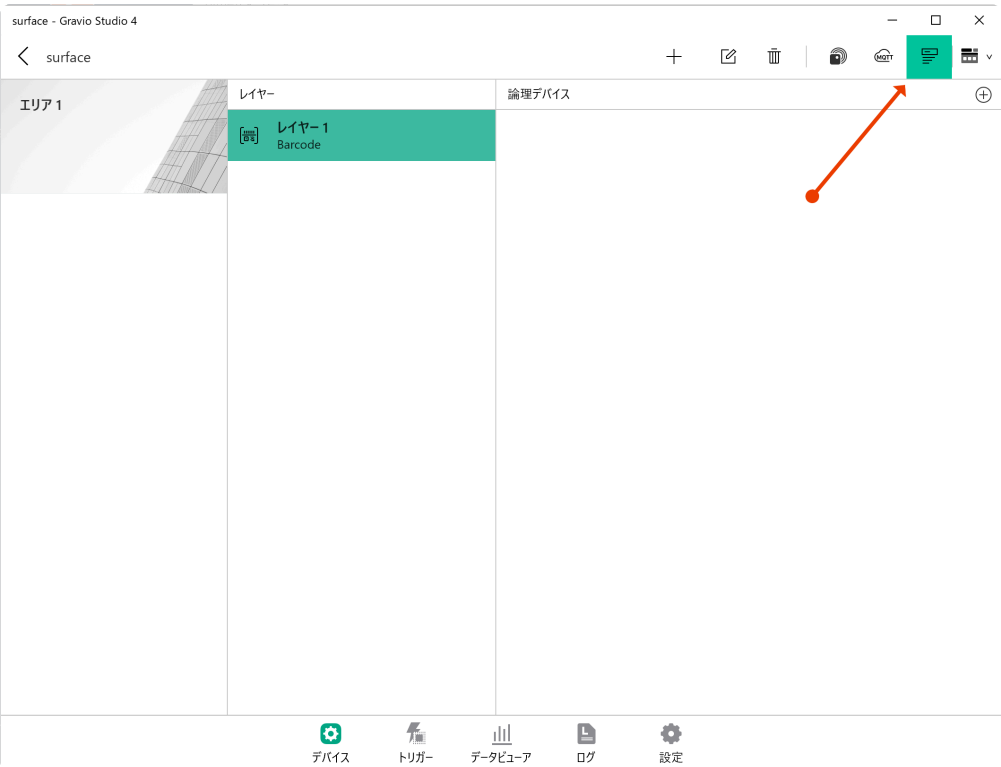


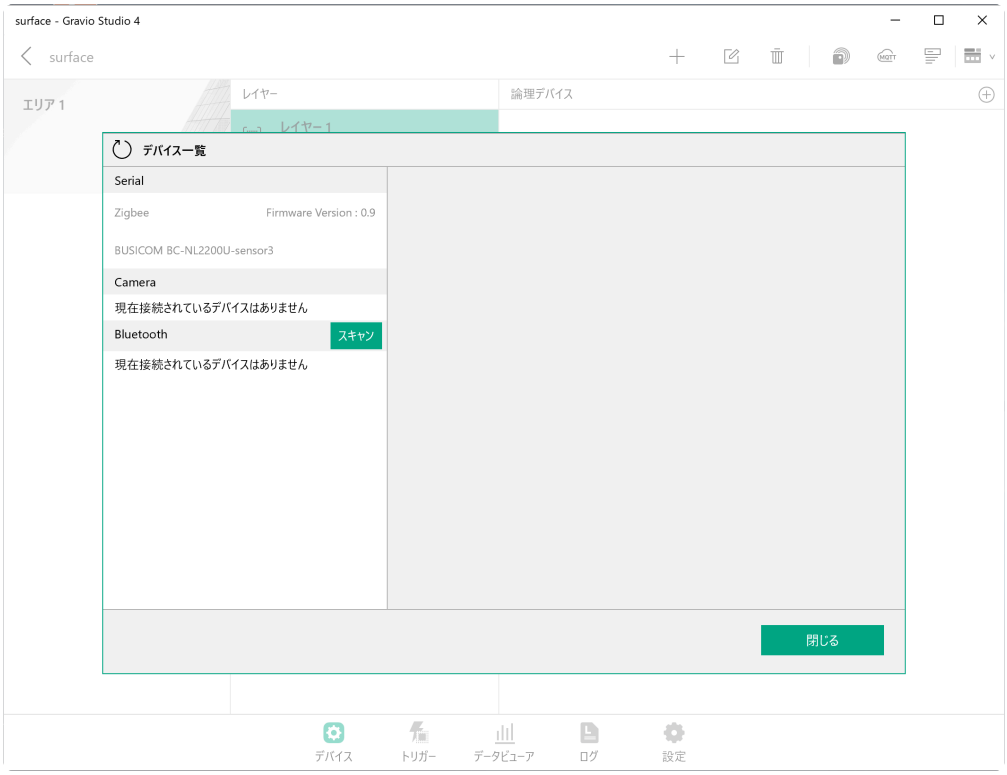
Adding a GPS device

1. Under ‘Devices’, select the GPS (DataKind) you want to use and add a new area and layer.



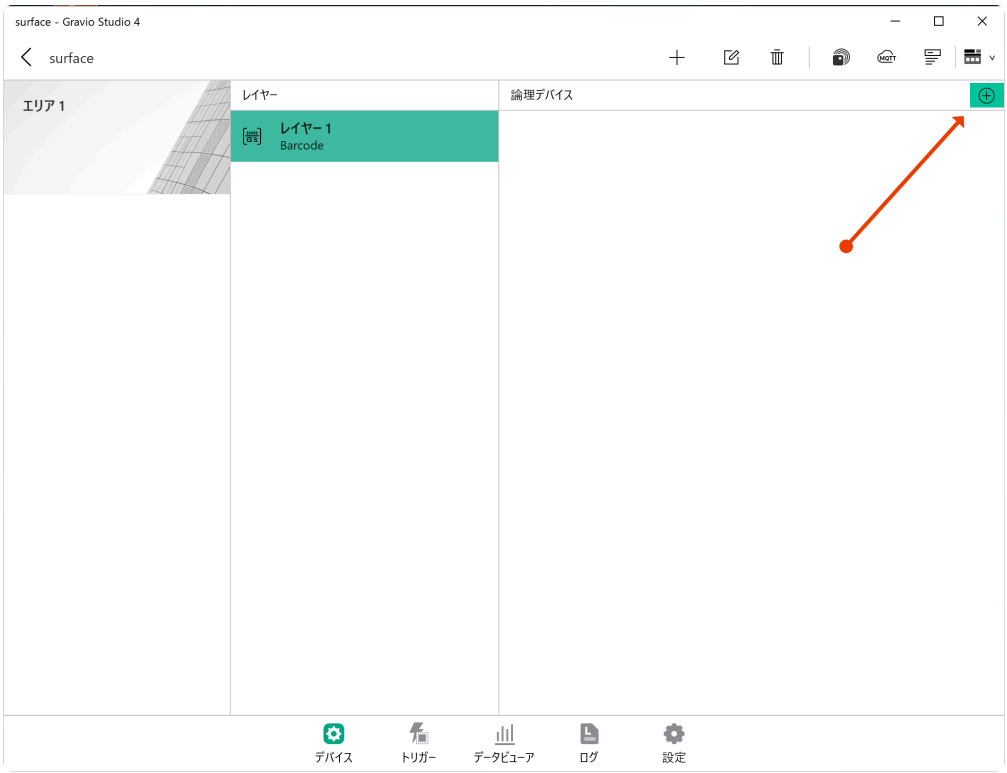
2. to receive GPS data, connect the GPS device to the USB port and then press the following button to display the GPS device settings screen.



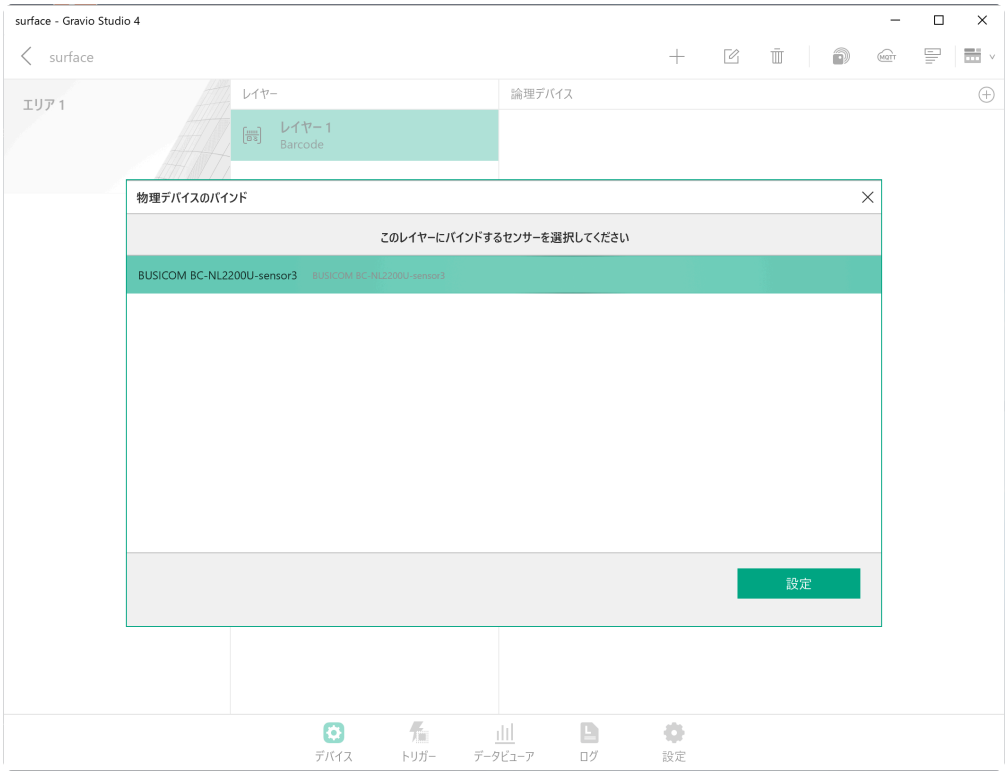


If the BUSICOM BC-NL2200U is displayed, you are done. Close the dialogue.

3. next, click on the circled ‘+’ symbol in the top right-hand corner of the window.

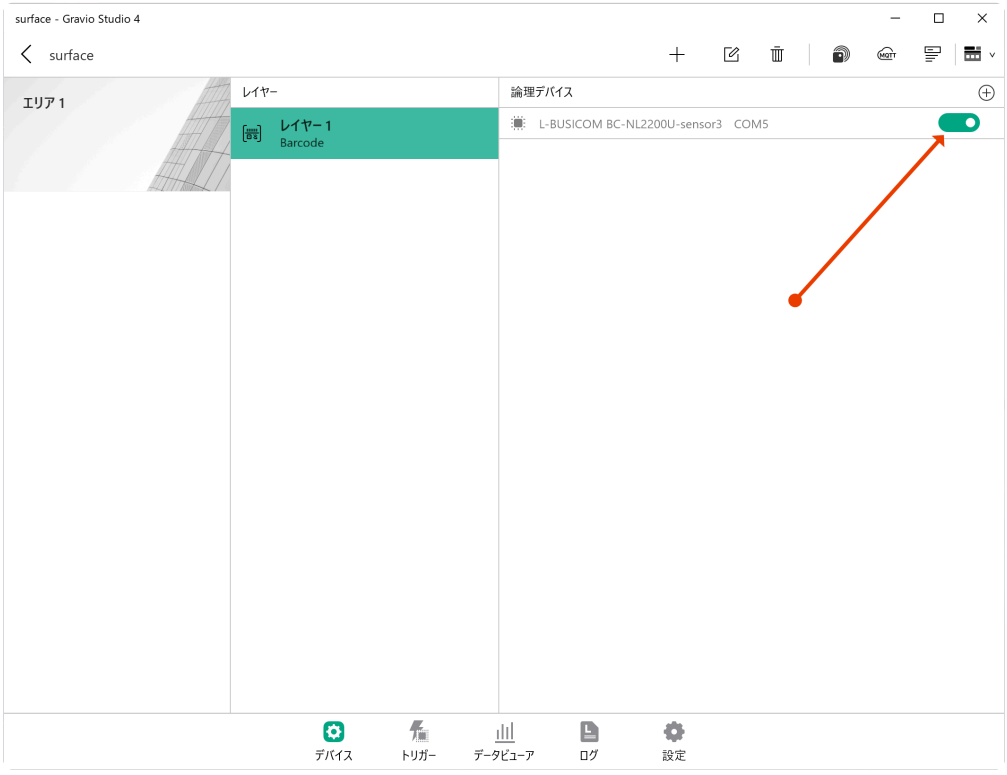


4. select the GPS device you wish to connect to the layer.



When you have finished the settings, close this screen and

5. switch on the device to receive data from this GPS device:



This will automatically start receiving GPS data from the GPS device.

6. The format of the GPS data is as follows.

Filtering the incoming data

This is used to extract only results that match the conditions from the inference results.

Data	Description
Course	The device’s direction of movement (number from 0° north to 359.9° clockwise) if the value is -1, the direction is invalid
SpeedKm	Speed at which the device is moving (km/h)
SpeedKn	Device movement speed in knots
LatitudeDeg	Latitude (DEG format)
LongitudeDeg	Longitude (DEG format)
LatitudeDeg	Latitude (DEG format)
LongitudeDir	Longitude (DEGM format)
LatitudeDir	Latitude direction (N or S)
LongitudeDir	Azimuth in longitude (E or W)

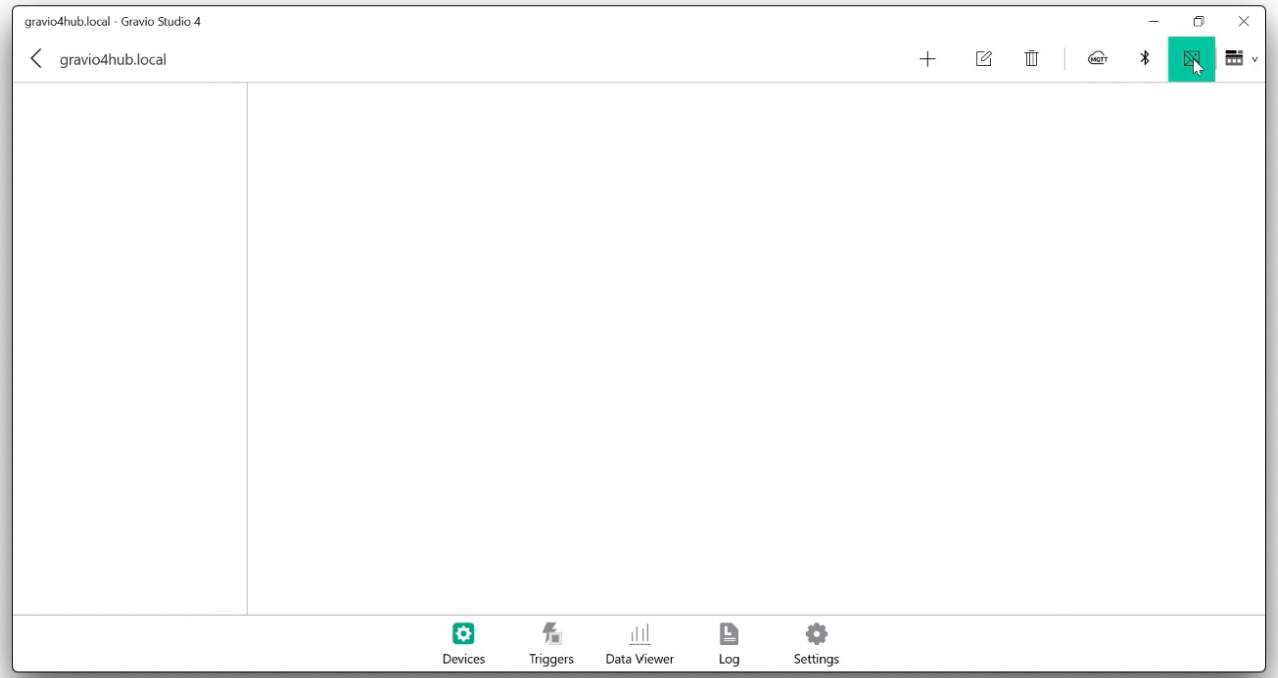
6.1.1.2. Software Sensors

Gravio also supports software sensors. An example for a software sensor could be a camera that detects how many people are in a room and triggers certain Actions if certain thresholds are reached. These sensors are created using software and artificial intelligence (with inference files) using TensorFlow.

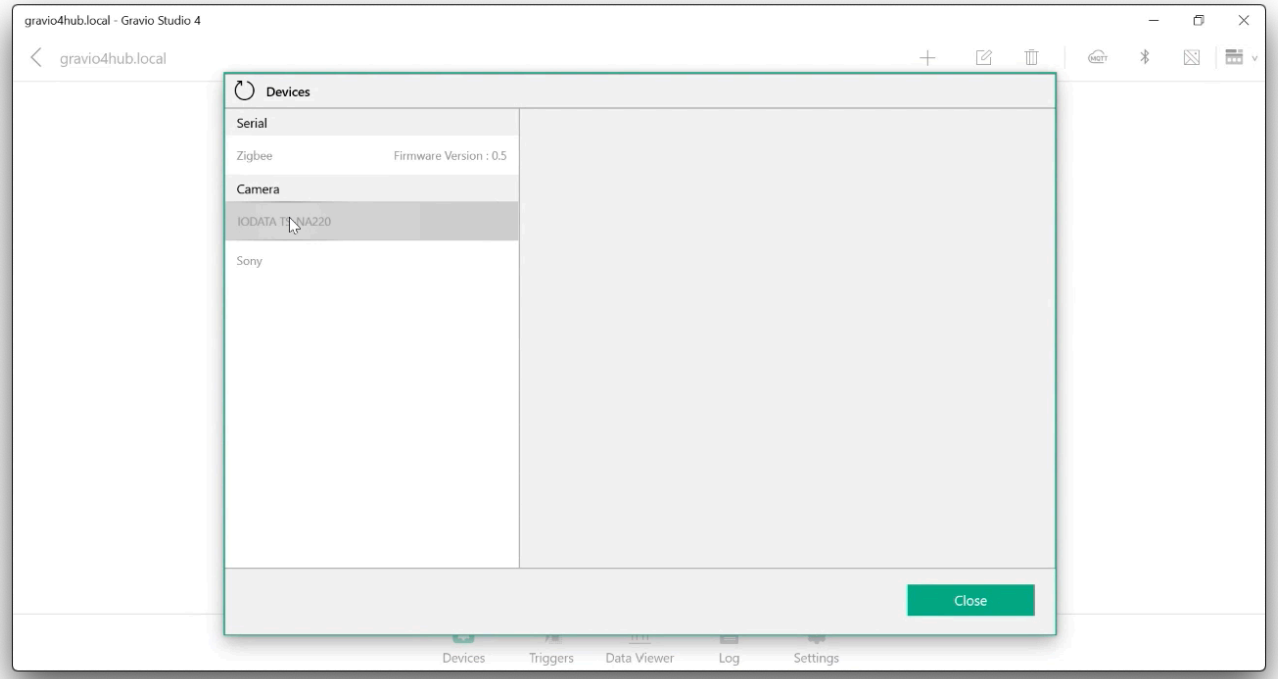
Creating a Software Sensor

Adding a Camera

First you need to add a ONVIF compatible camera to the network. Gravio supports the ONVIF standard (need to provide Profile T), which is very common in security camera settings. For this, please find the camera in your devices window:



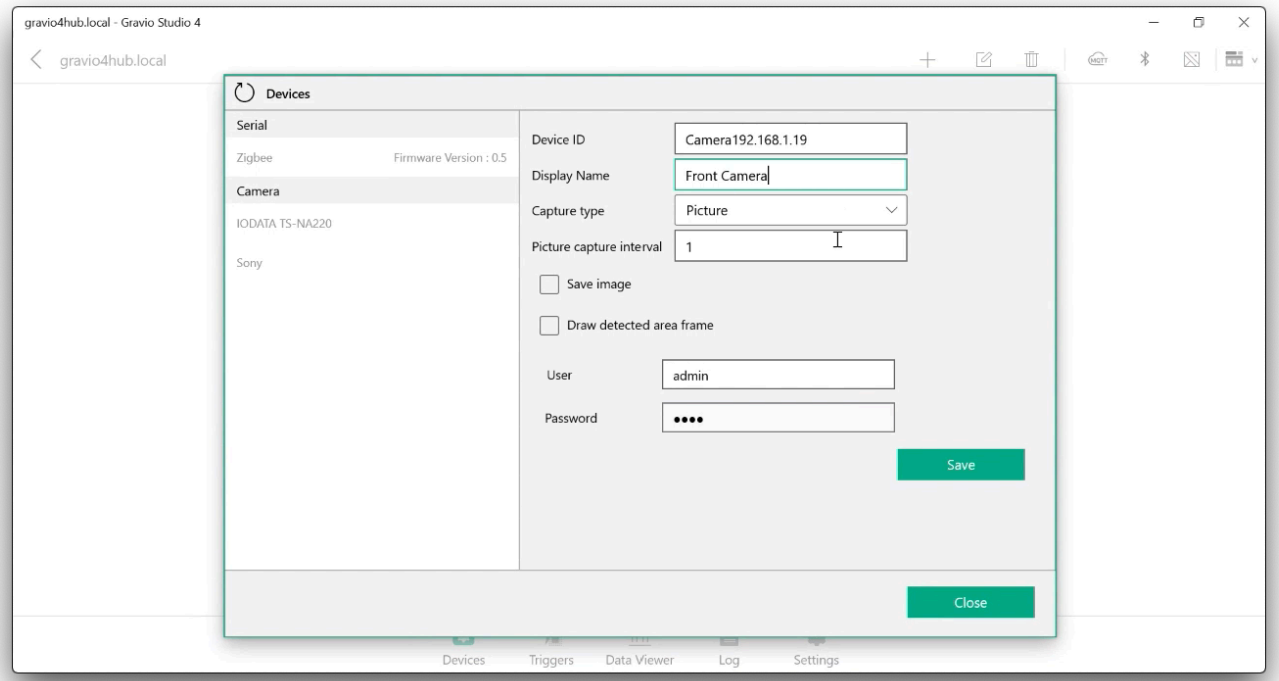
You will see ONVIF compatible camera(s) in your network. Please note that Gravio can detect ONVIF cameras that adhere to the ONVIF discovery standard. If you don't see your camera, please check your network or camera setting:



In this case we connect to a IODATA TS-NA220 and to a Sony IPELA camera.

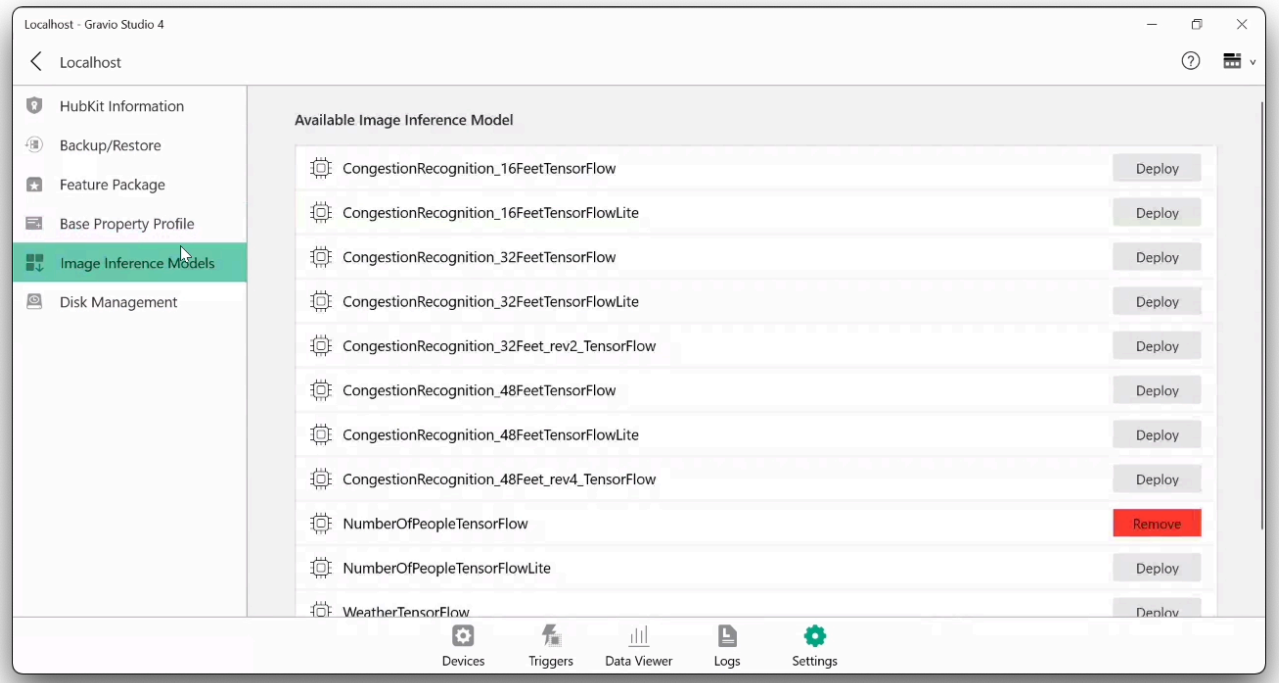
In the following screen you can set the parameters of the camera:

- The Device ID is fixed and needed to identify all information associated with this camera. It typically consists of the word Camera followed by the IP address
- You can choose the display name to make sense for your setting
- Pick either Picture or Video depending on what you like to capture. We recommend picture as it uses less space.
- The interval sets how often a picture should be taken. Note again, the less often, the less space you need on your hard drive.
- Both the above items are relevant if you decide to save the picture. The Save image tickbox determines if the picture should be saved after it has been processed. Leave it blank to save space, leave it ticked, to verify if the computer is doing the right thing
- If you save the picture, you can also draw a box around the detected area. For example, if you are detecting people in a picture, it will draw a line around the detected people. This helps to verify if the visual recognition model works correctly. Here you see where the images get stored. Note, if you store the images, your hard disk might fill up very quickly. You may therefore go to “Settings” and under “Disk Management” set an auto-delete of the media files after a suitable amount of days.
- The username and password are the authentication details to access your camera.



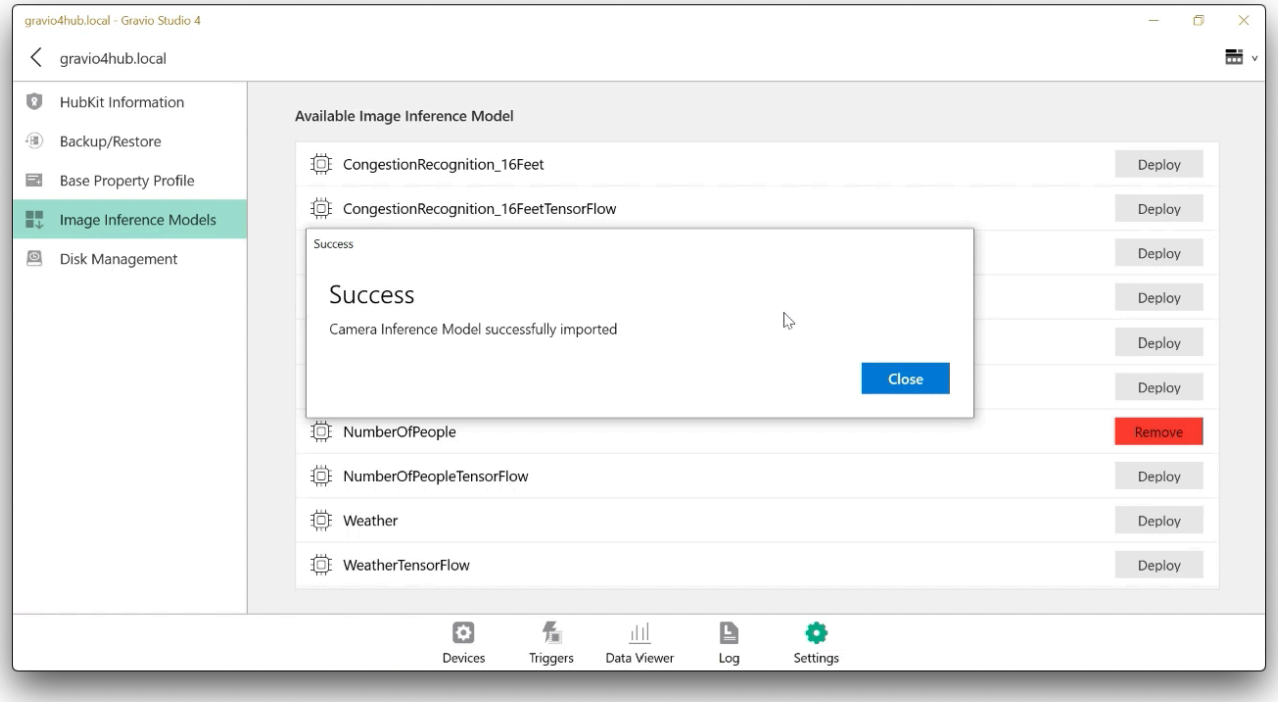
Adding the AI/Image Recognition Model

Now that you have added the camera, you can add the Image Recognition models to Gravio that the camera should process. You can find the available models in the Settings tab of Gravio Studio, then in the “Image Inference Models” section:



On the right-hand side, you will see all the available models. Make sure your Gravio has access to the internet, then just press the “Deploy” button to the right of the model you require. It will download the

necessary model files and make them available for the final setup. Please note, that each model file can easily reach 200-300 megabytes in data, so be aware of this when downloading and storing the models.

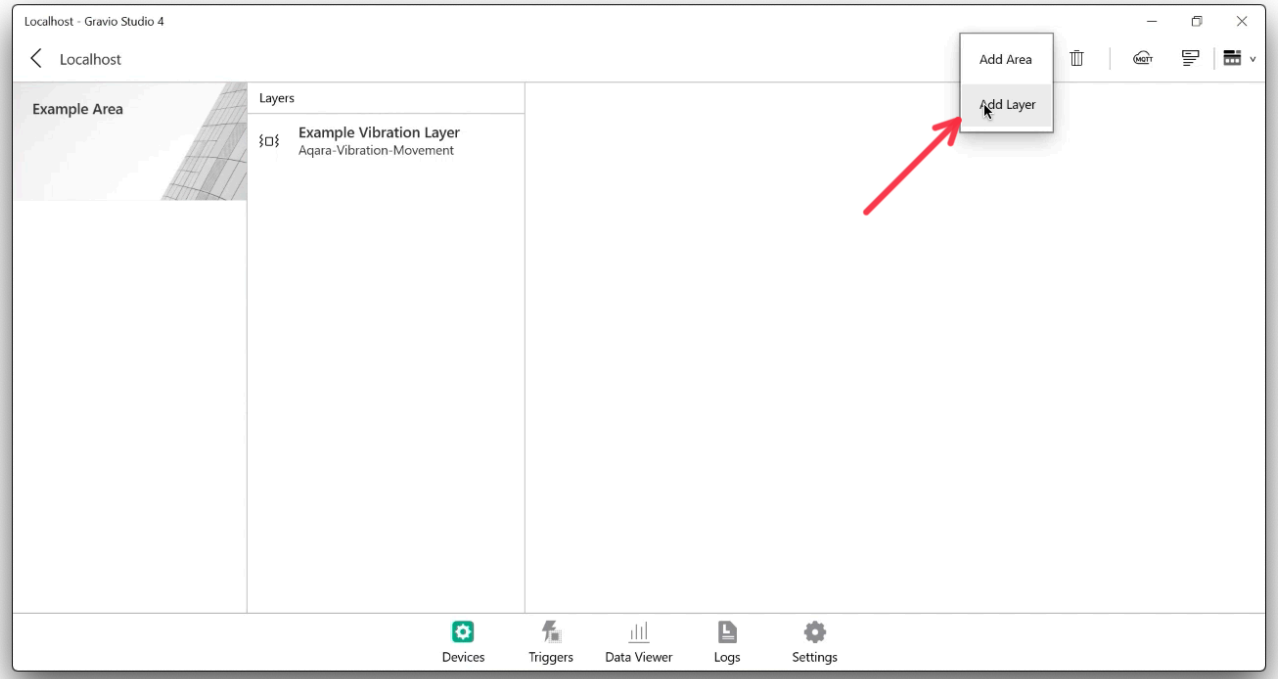


You can also create your own models, however, this is more advanced, and needs a bit more bespoke work. Please don't hesitate to get in touch with us in our slack channel if you are interested in doing this: [Join Gravio Slack]

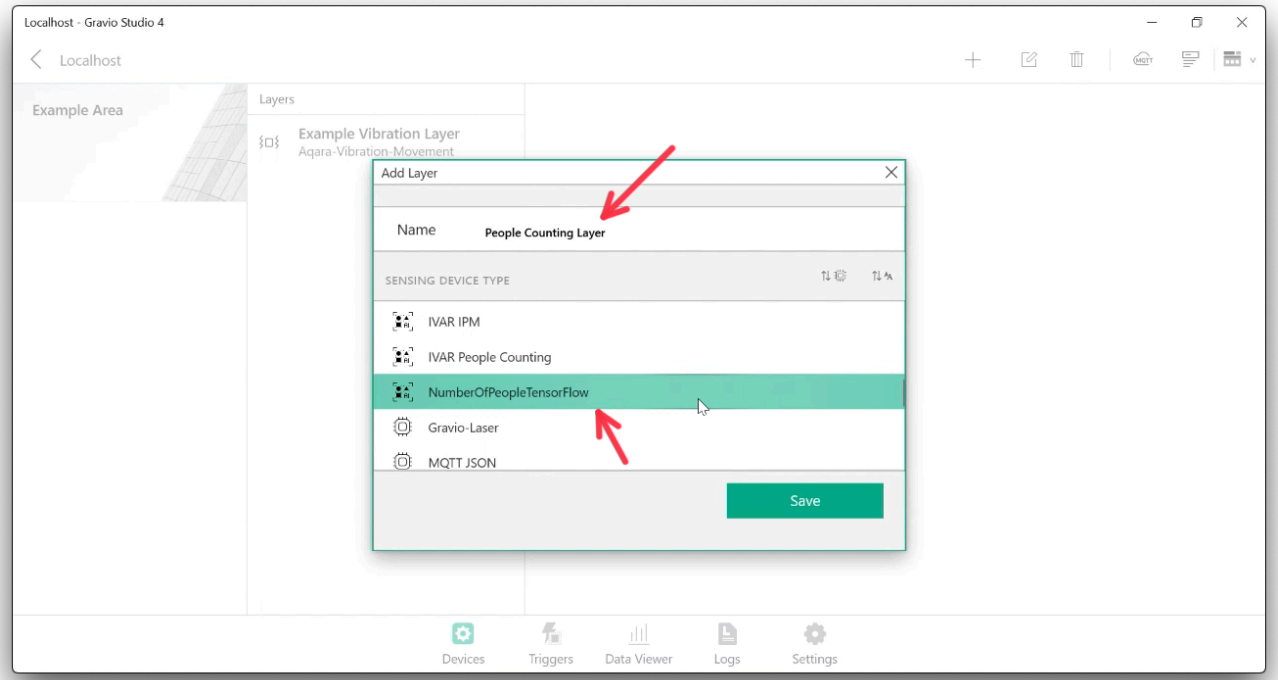
Adding the Layer and connecting the device and trigger

Now that you have installed and connected the camera and deployed the AI model, you have a new layer available that you can add in the Area section of the Gravio Devices tab. Just add the layer as you would any layer and pick the layer named after the AI model. This will then behave like any other hardware sensor.

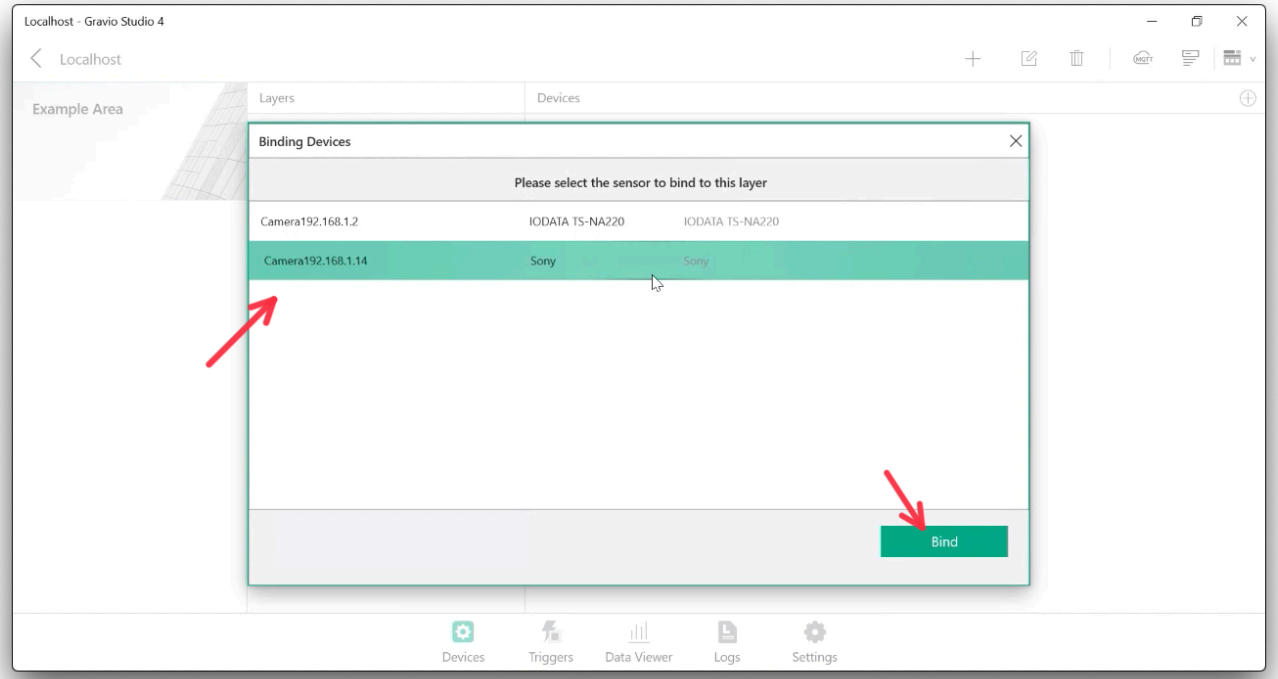
Add a layer to the area:



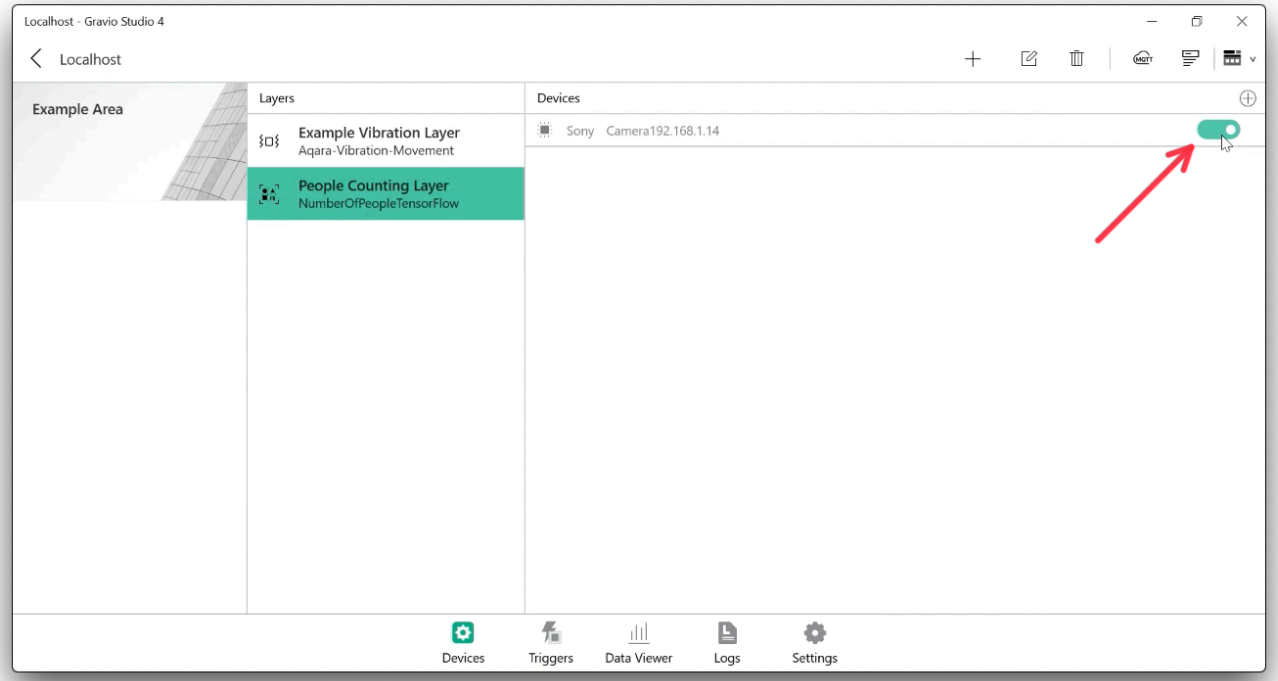
Give it a meaningful name. You now have a new Sensing Device Type from the AI model, choose it and save it:



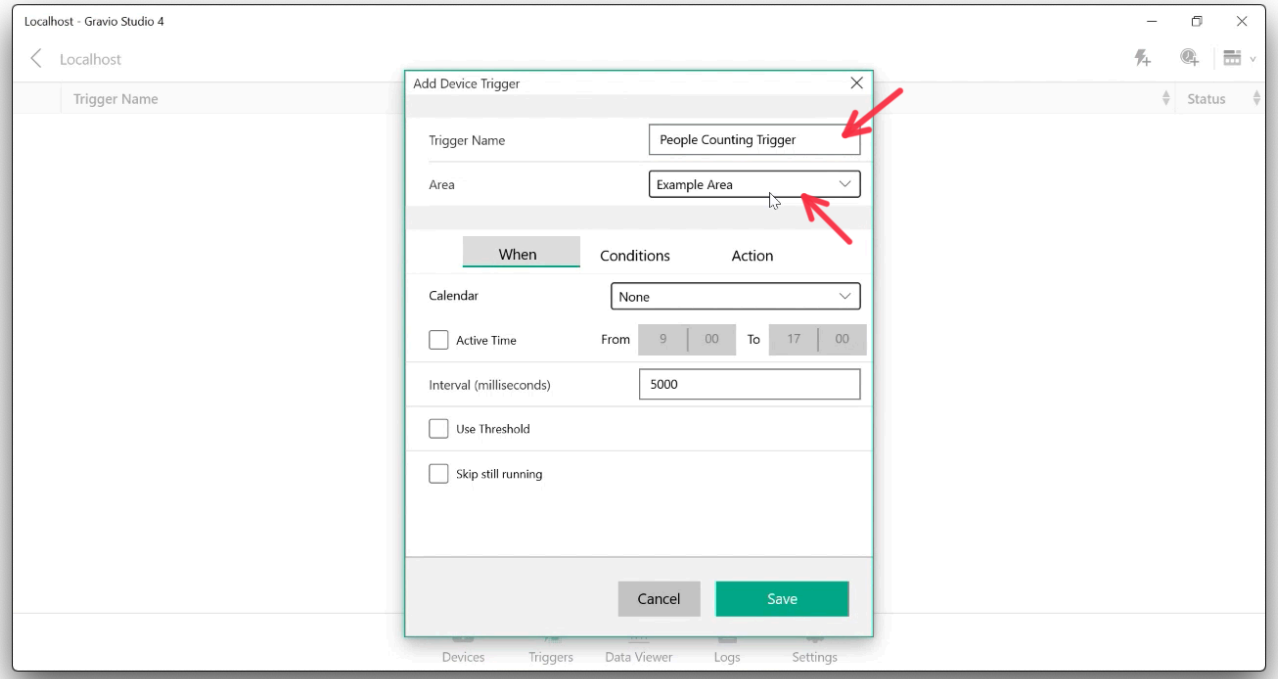
Pick the camera feed you want to use with this model. These camera devices will automatically appear here if they are using the standard ONVIF and its discovery protocol (see above):



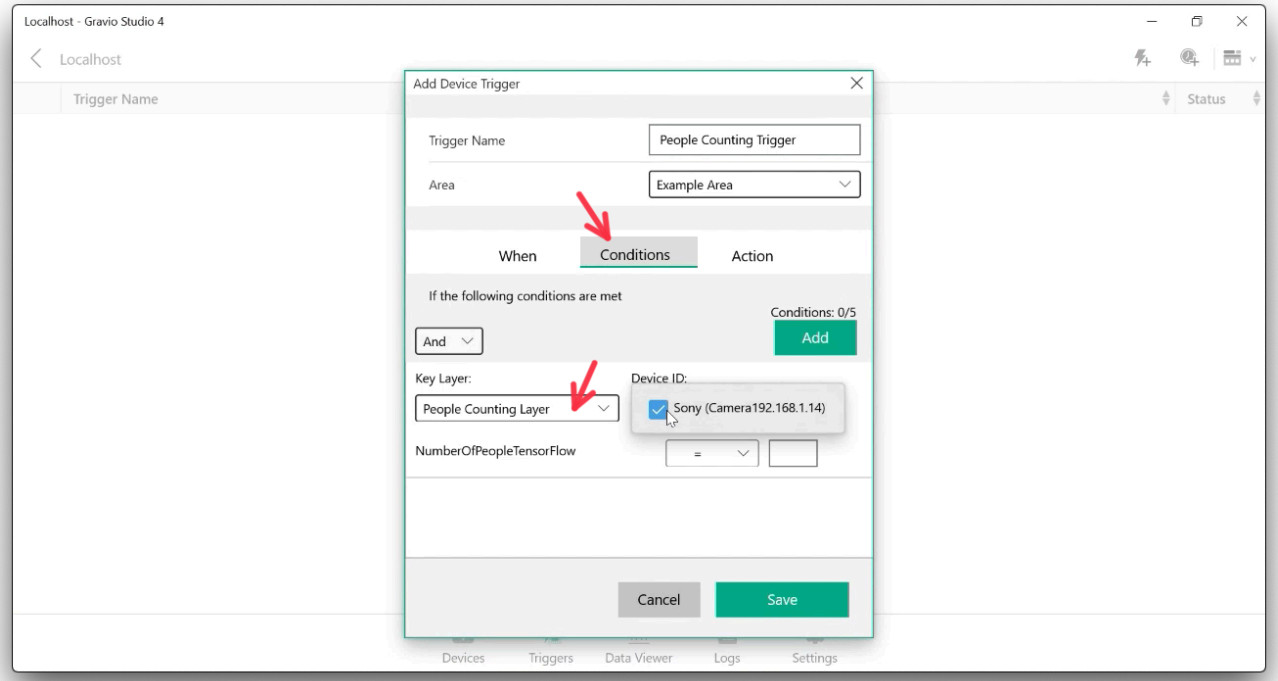
Enable the layer:



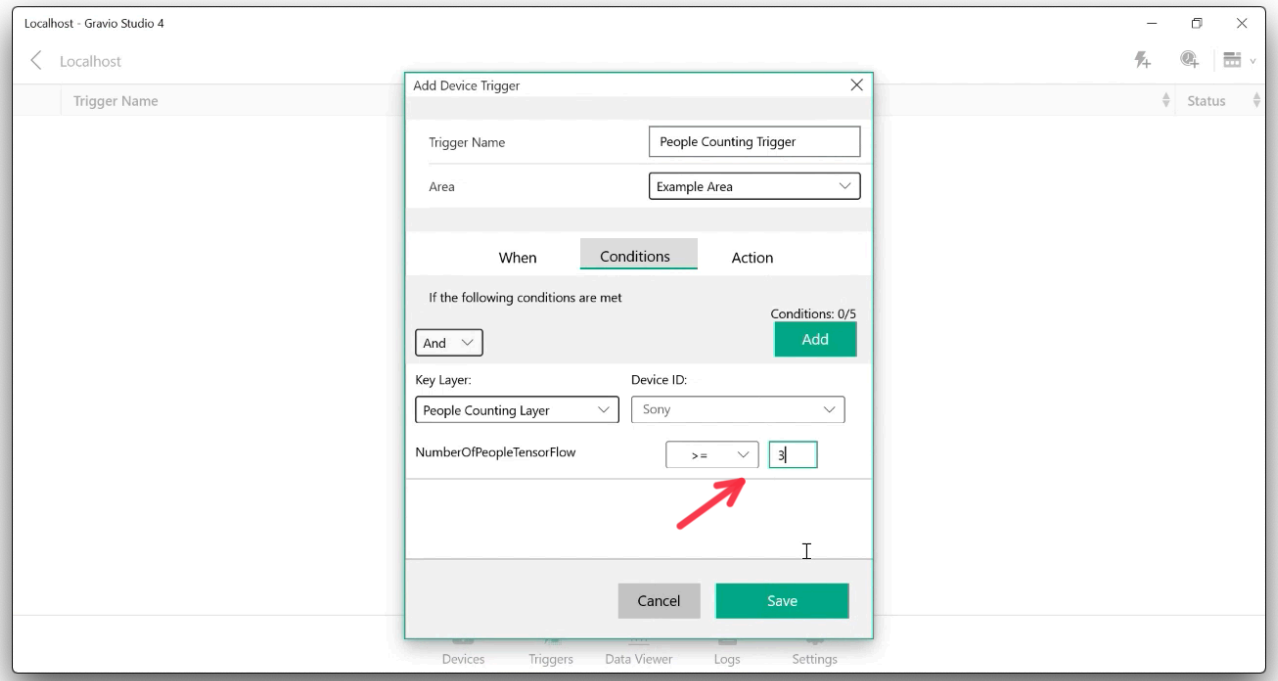
Now let's add a trigger to the software sensor. Pick the trigger tab and create a new trigger. Give it a sensible name and pick the right Area that contains the software sensor layer:



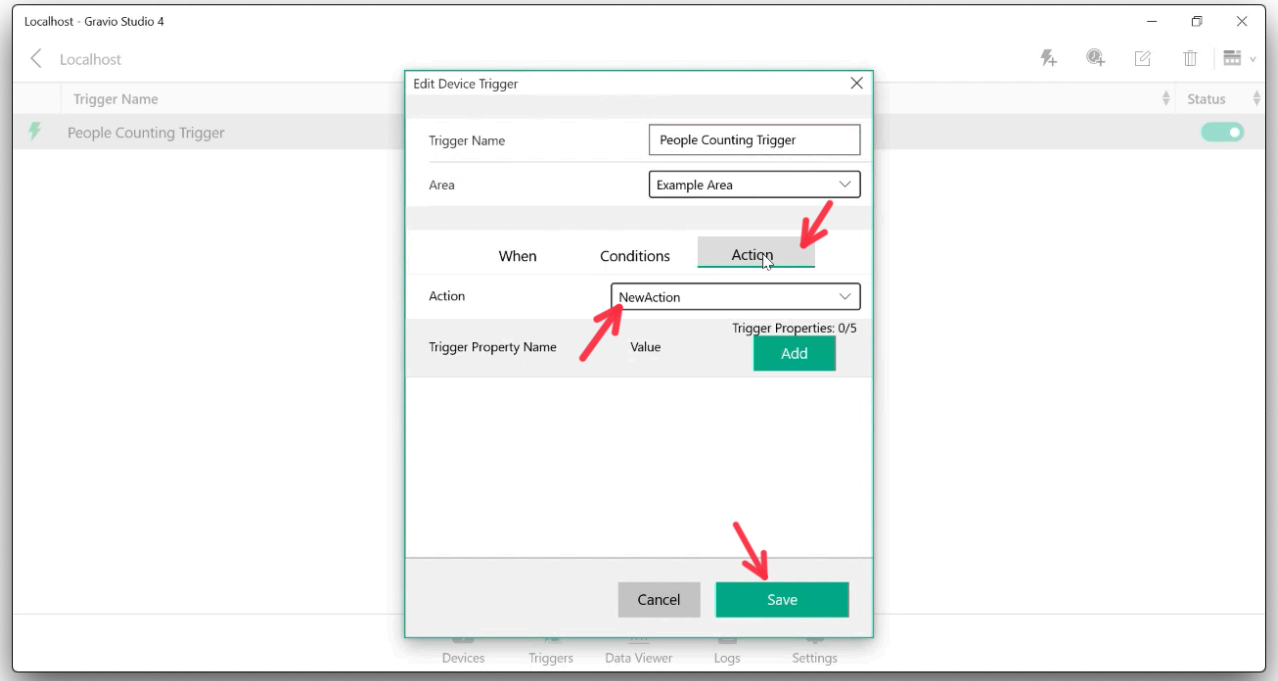
Under “Conditions” select the layer and the camera:



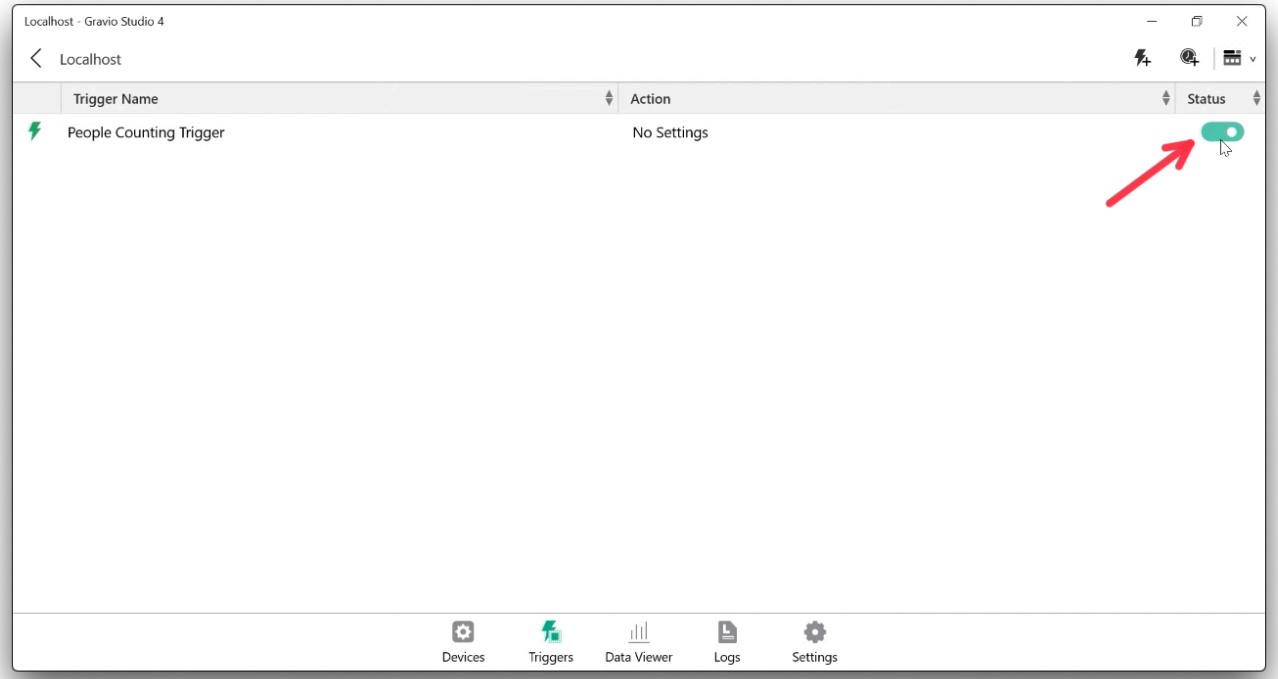
Then set the condition under which the trigger should fire:



And an action it should execute:



Don’t forget to enable the trigger



You are now ready to go and have created a camera vision layer and trigger. You can find more info about creating triggers under the section Trigger Tab
You can find more about how to setup a ONVIF Camera by watching this video:
https://www.youtube.com/watch?v=8dkSIz_5810
You can find out where the images are stored here

6.1.1.2.1. Inference Models

Gravio can infer camera images using TensorFlow , a Google system used for Artificial Intelligence and Machine Learning. Two types of TensorFlow files are available:

- TensorFlow and
- TensorFlowLite.

There is a difference in the number of deep learning layers in the inference process:

- TensorFlow has 32 layers and
- TensorFlowLite has 16 layers

While TensorFlow is more accurate in inference, it requires more processing time and memory. We suggest that you verify with your environmental conditions such as lighting, contrast, camera resolution etc. if the TensorFlowLite model is sufficient.

Available Inference Files

The following inference files can be used in Gravio per default:

Recognition of congestion

Approximate distance	Inference file name
16Feet	CongestionRecognition_16FeetTensorFlow
32Feet	CongestionRecognition_32FeetTensorFlow
48Feet	CongestionRecognition_48FeetTensorFlow

Recognizes the degree of congestion in the area (range) captured by the camera.
16Feet, 32Feet, and 48Feet are approximate distances to the area recognized by the camera.
48Feet is a wider range than 16Feet, so please use the larger number to identify people in the distance.

Outputs: Integer of number of people detected (counting heads, optimised for an indoor setting.)

Number of people counting

NumberOfPeopleTensorFlow

This inference counts the number of people recognized by the camera.

Outputs: Integer of number of people detected (counting full bodies, optimised for an outdoor setting.)

Weather recognition

WeatherTensorFlow

Recognizes the weather in the sky recognized by the camera.

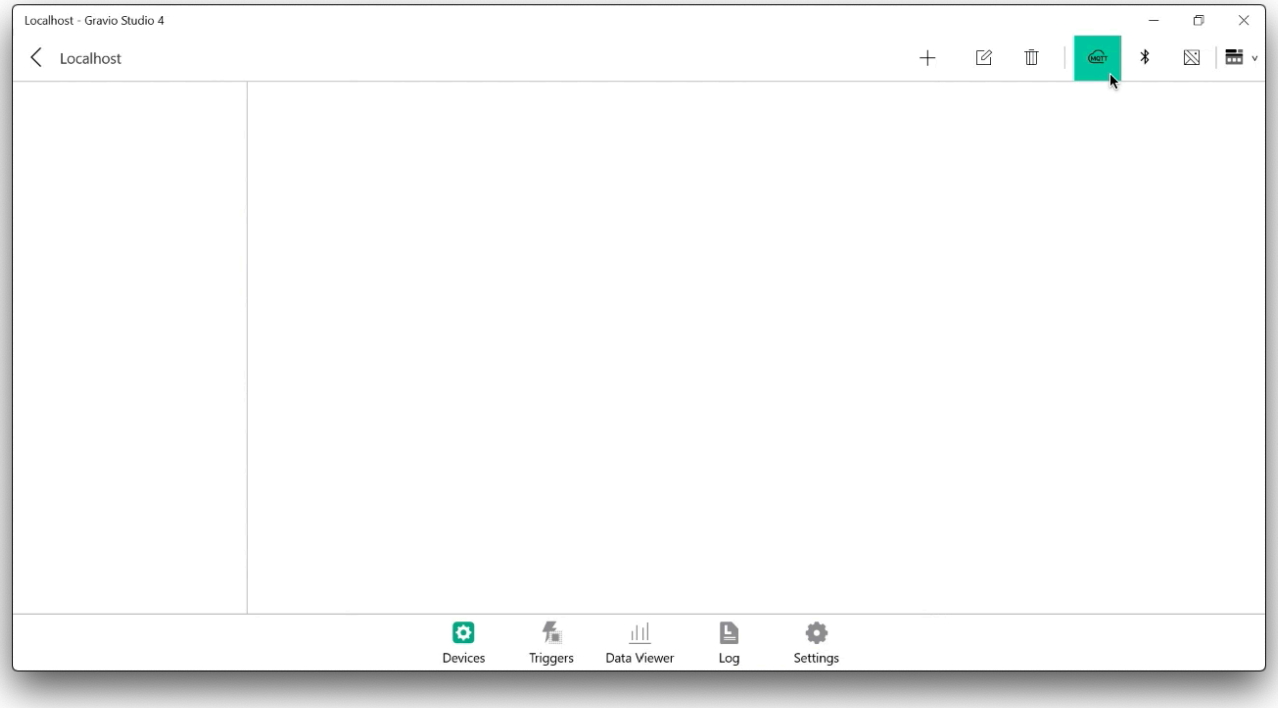
Outputs: windy, thunder, cloudy, typhoon, snowy, sunny, rainy, unknown

If you have a free or basic subscription, you get the people counter and weather models in the bundle. If you have the Enterprise subscription, you can upload your own AI/ML models to your Coordinator back end and push them to the Gravio HubKits. Note, the models are general purpose models and they may not work specifically for your circumstance. If you need more accurate models, we recommend training them specifically for your setting.

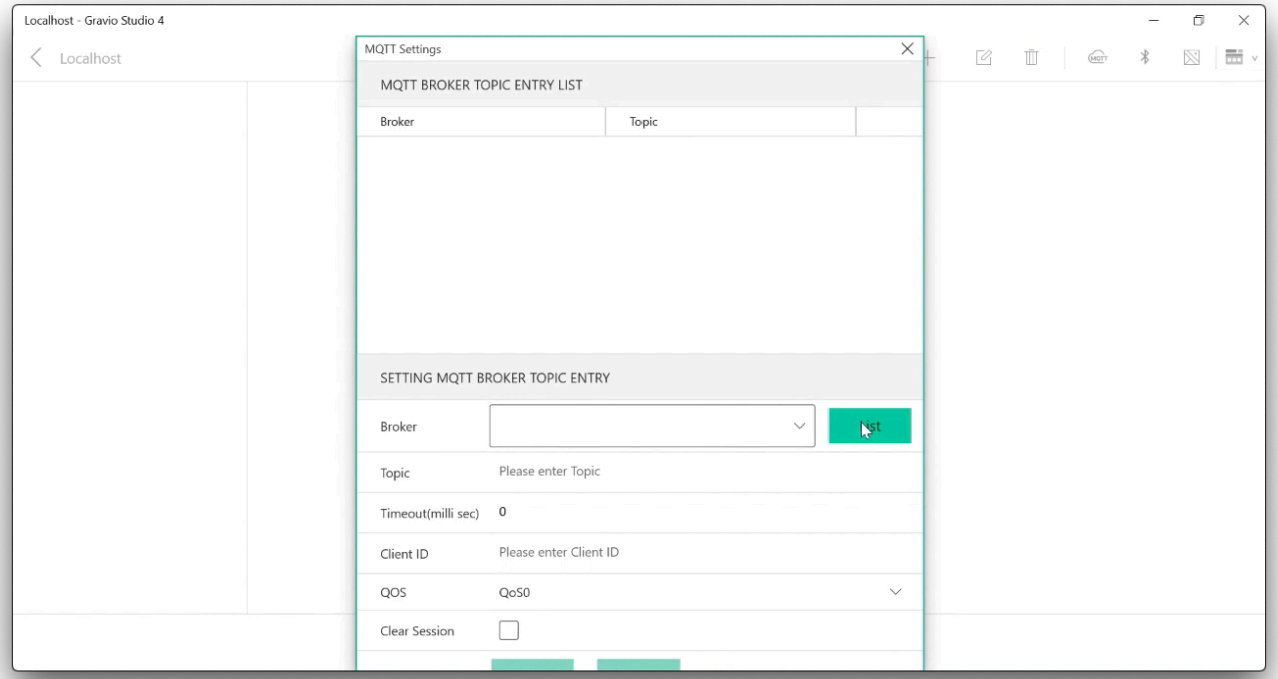
6.1.1.3. MQTT Subscriptions

You can subscribe to topics from an MQTT broker. These incoming messages would then be treated like sensor input data.

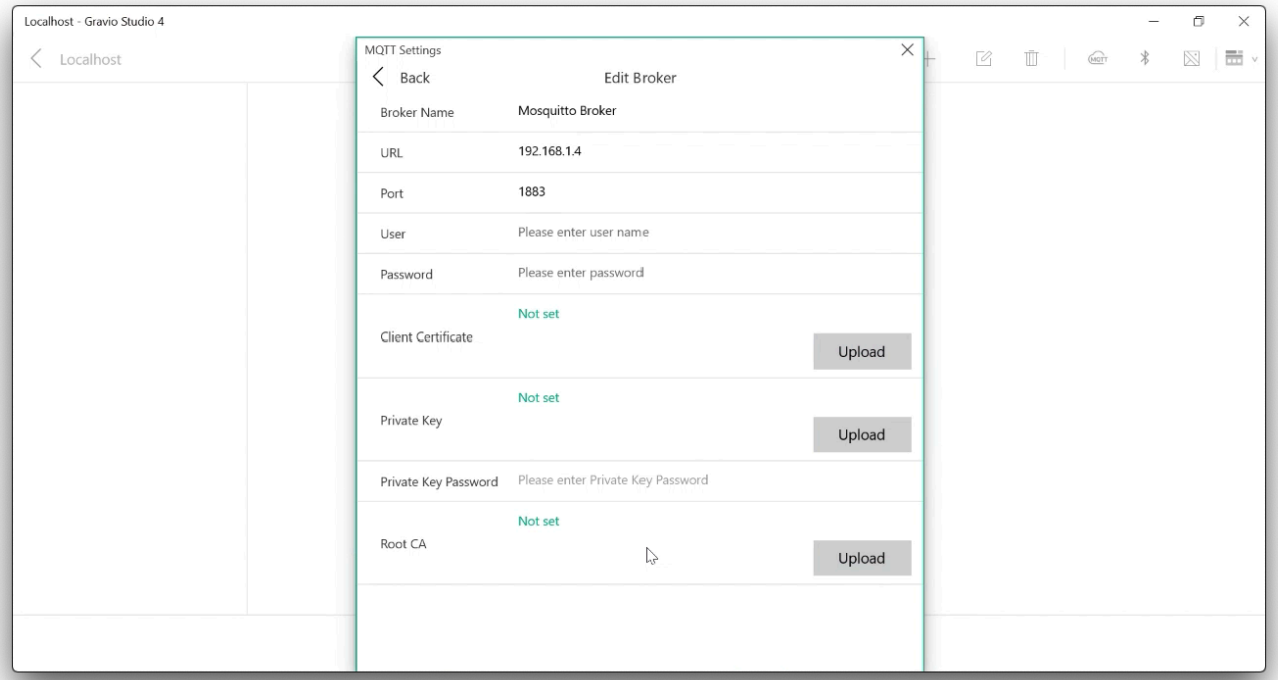
To create an MQTT layer, click on the MQTT button on the top right of the screen:



This will open the overview of your available MQTT Brokers. You can add your brokers by clicking on the “List” button in the Broker row:

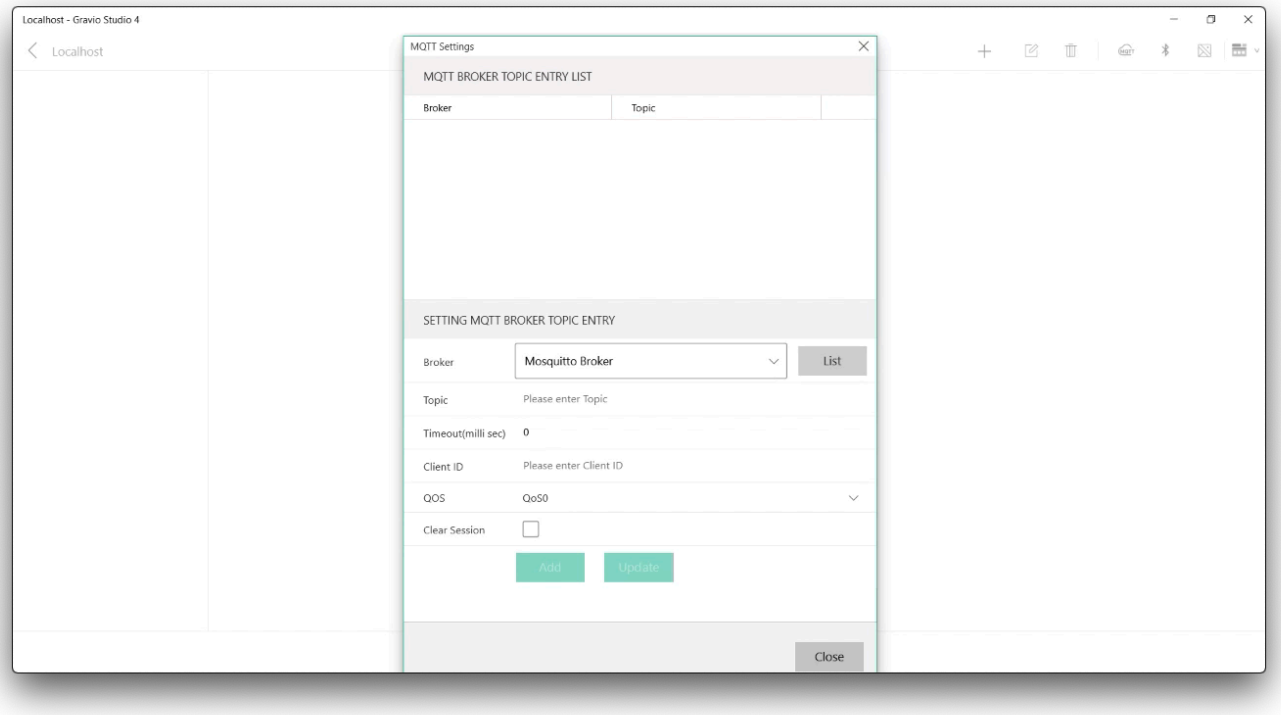


Clicking on that button will open the dialogue to add new brokers. Note, for encrypted connections, prefix the URL with `ssl://`:

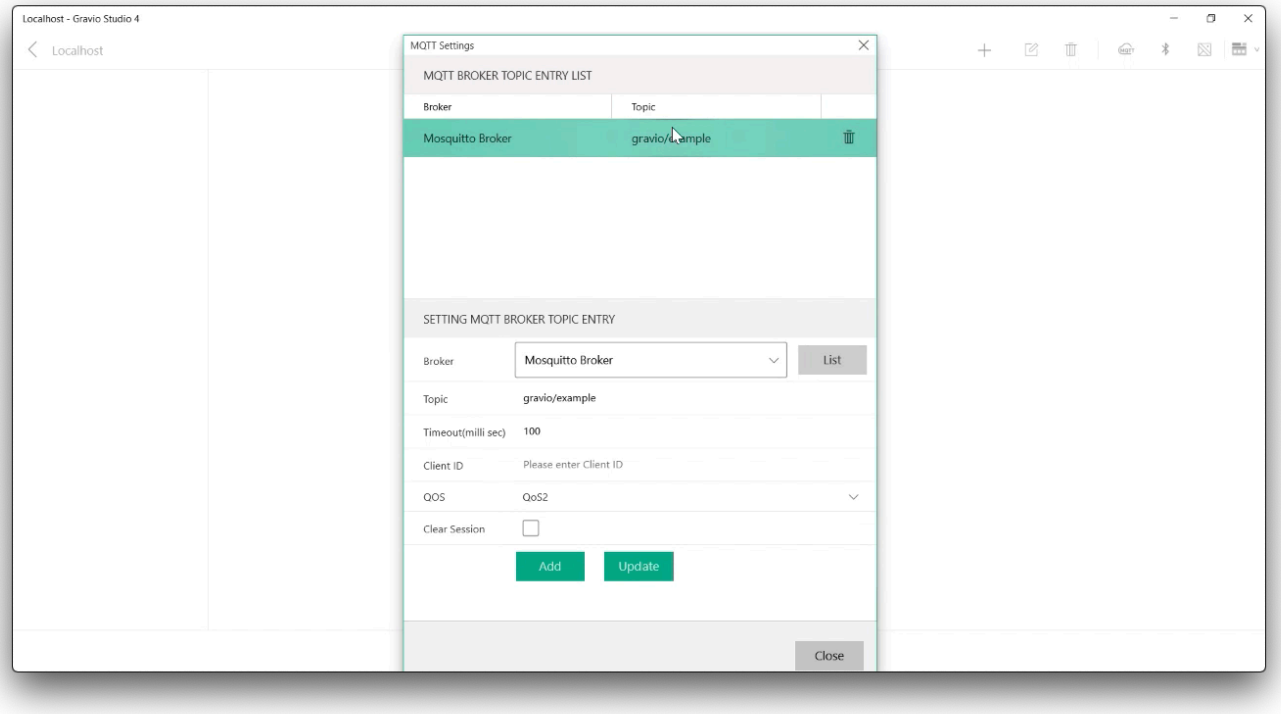


Fill in all the details of your broker and upload the necessary certificates, keys and key passwords.

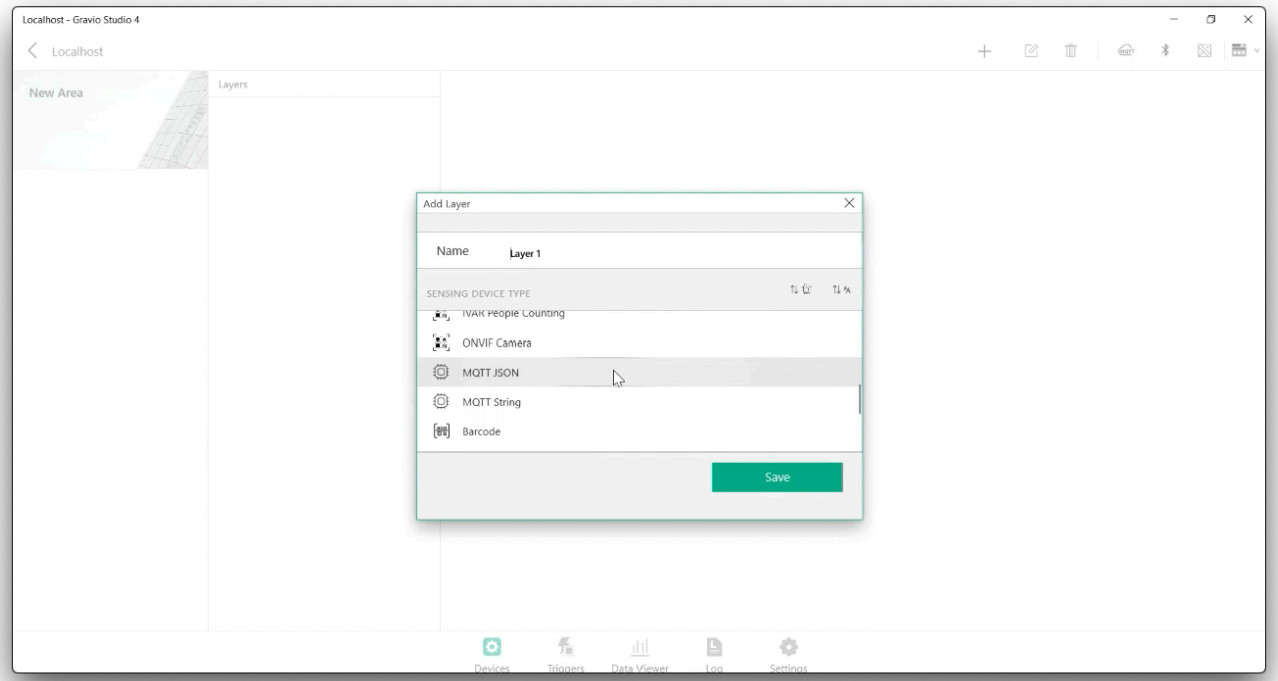
After you have added this, your Broker will appear in the drop-down menu:



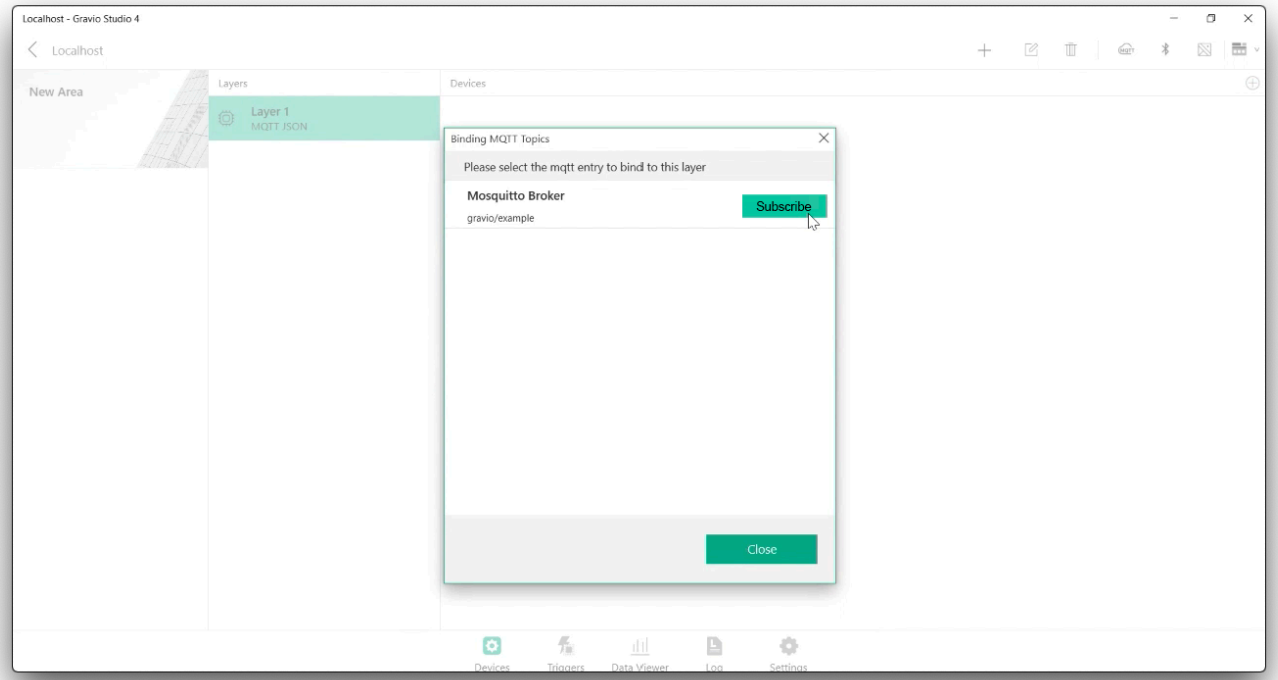
Choose your broker, and enter the topic, QOS and timeout you'd like to use to subscribe to a topic. Once you've added these details, your entry will appear in the list above:



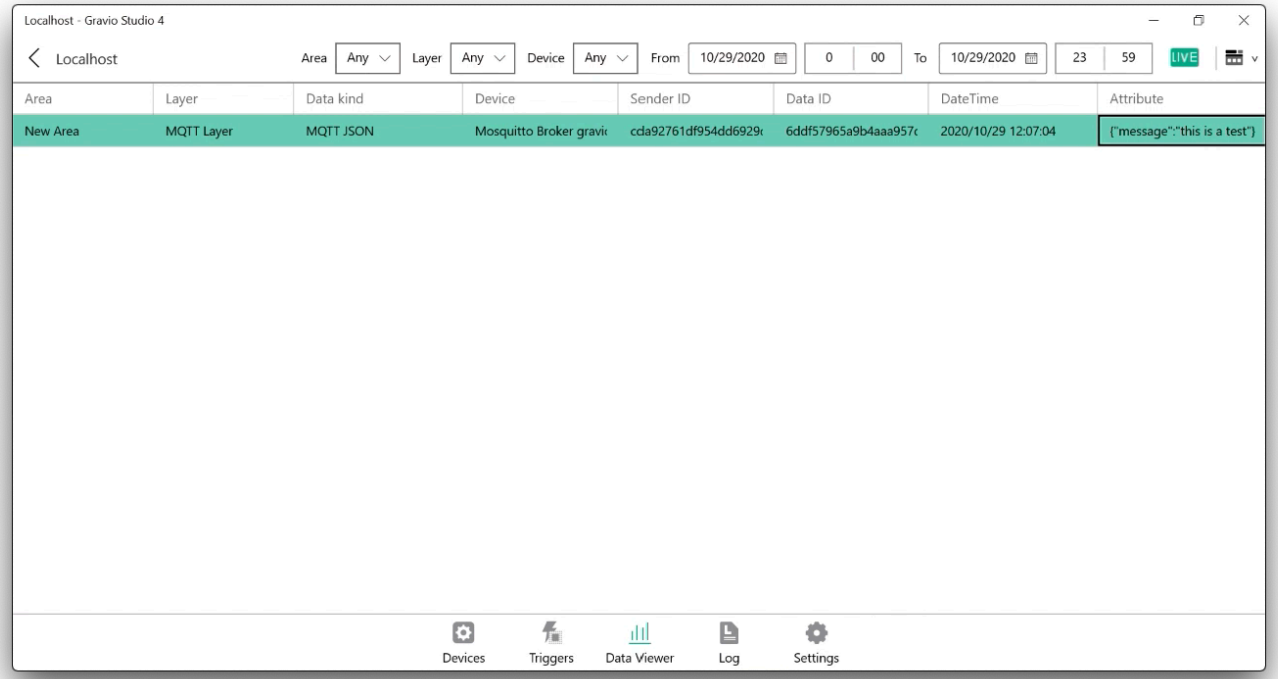
Once it's added there, you will be able to add it as a layer, similar to adding a sensor layer:



And subsequently, after clicking on the plus sign on the top right, you can subscribe the layer to a topic that you have created in the previous steps:



If you have a MQTT broker installed such as for example Mosquitto you can also use a program such as `mosquitto_pub` from the command line or MQTT Explorer to send a command and see if it appears in the Gravio Data log:


Localhost - Gravio Studio 4
Localhost
Area
Any
Layer
Any
Device
Any
From
10/29/2020
0
00
To
10/29/2020
23
59
LIVE
Area
Layer
Data kind
Device
Sender ID
Data ID
DateTime
Attribute
New Area
MQTT Layer
MQTT JSON
Mosquitto Broker gravi
cda92761df954dd6929
6ddf57965a9b4aaa957
2020/10/29 12:07:04
{"message":"this is a test"}
Devices
Triggers
Data Viewer
Log
Settings

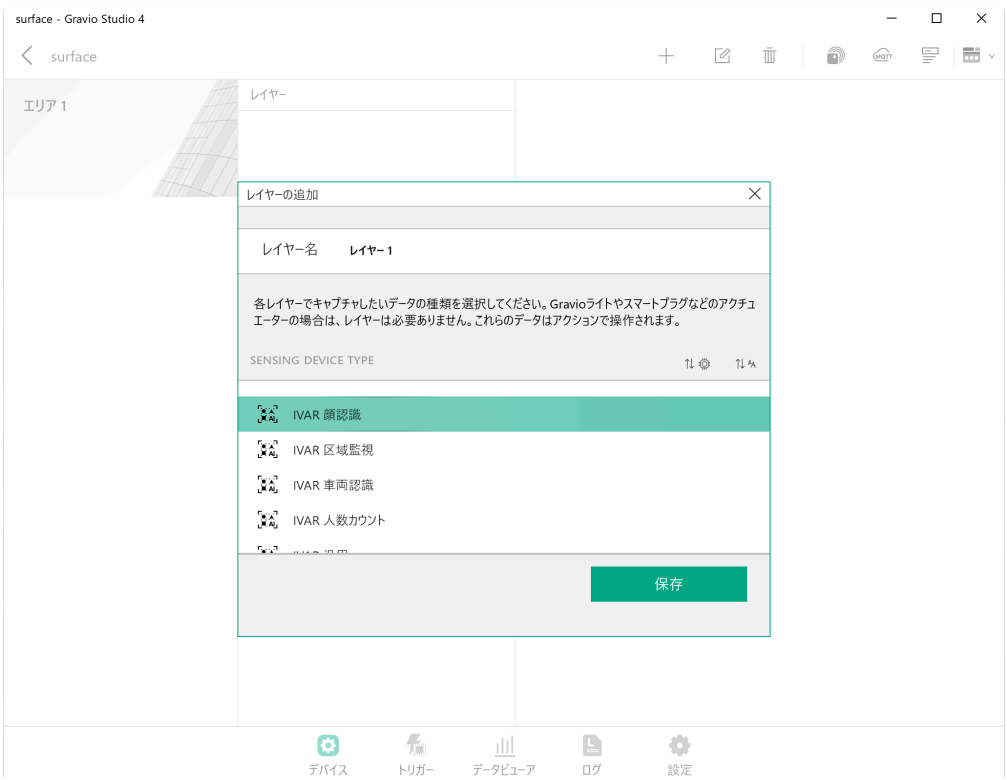
6.1.1.4. Gorilla IVAR

In Japan, Gravio works with the Gorilla IVAR image recognition system and receives the data recognized by IVAR.

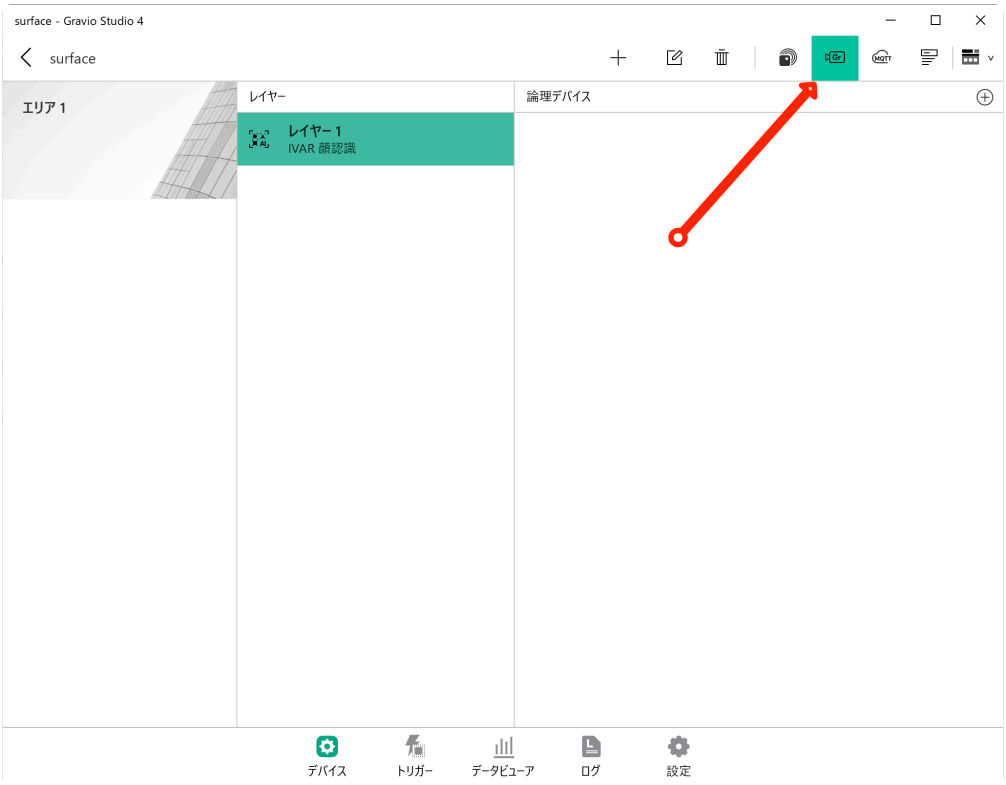
Creating an IVAR

In Device, select the IVAR type (DataKind) and add a new area and layer.

IVAR will receive data in the following formats: face recognition, area monitoring, vehicle recognition, headcount, and general-purpose.

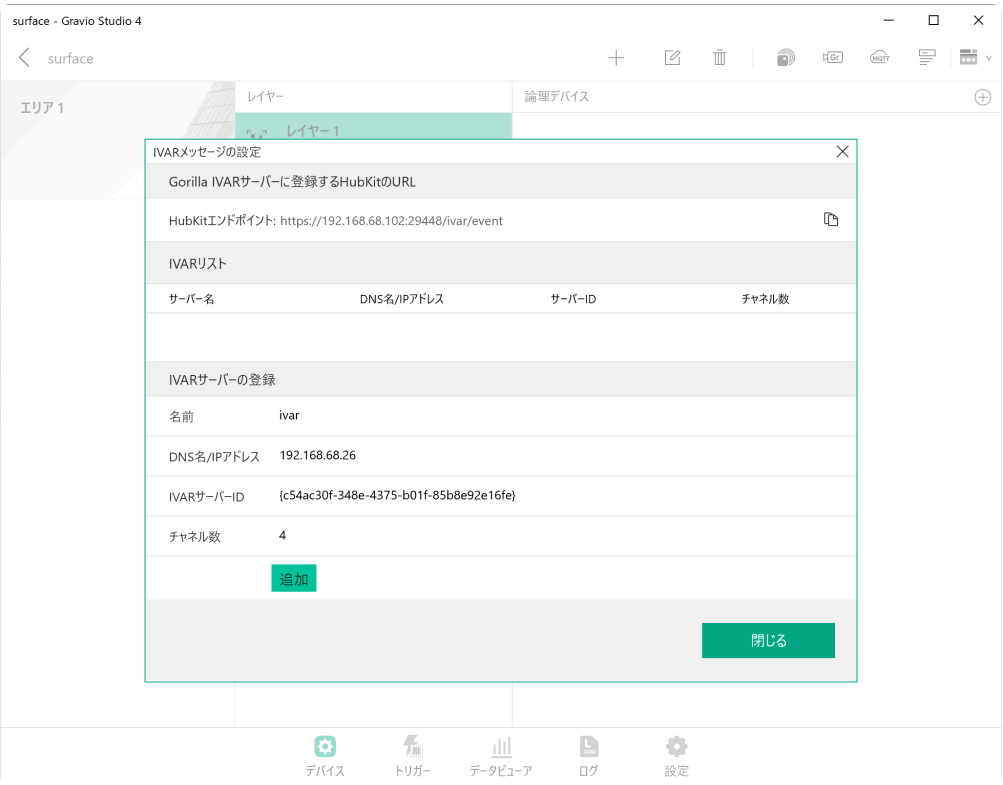


Press the following button to display the IVAR setting screen.

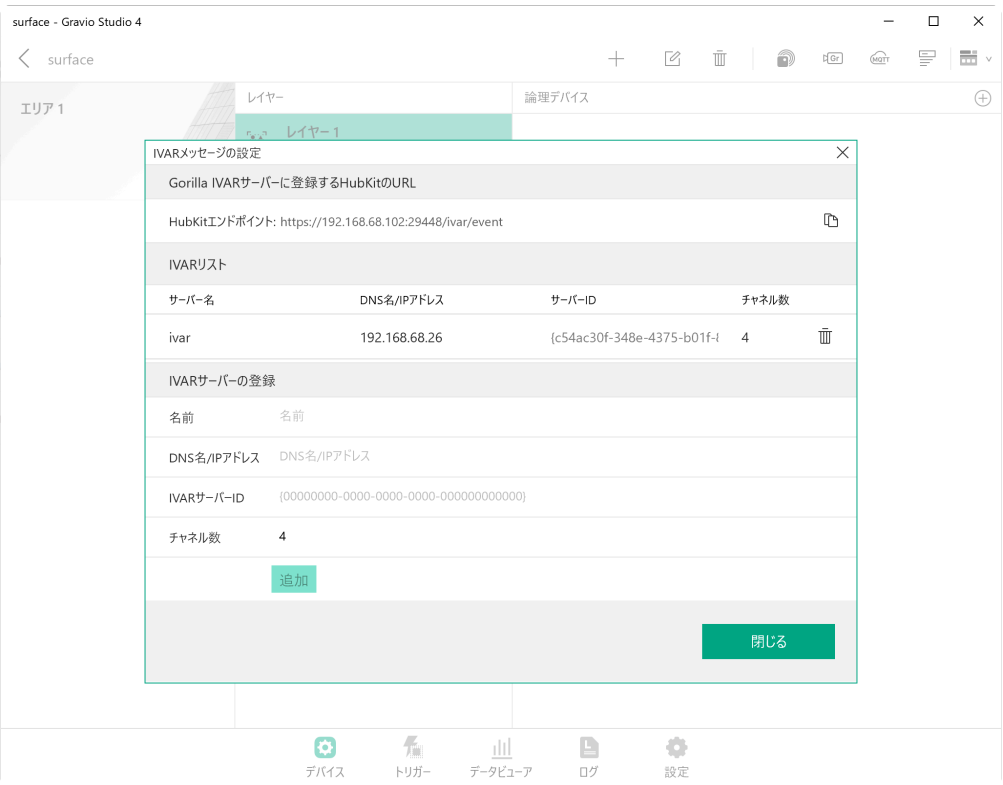


Enter the IVAR and click the Add button. copy the HubKit URL to the clipboard to register it to the Gorilla IVAR server and register it to IVAR.

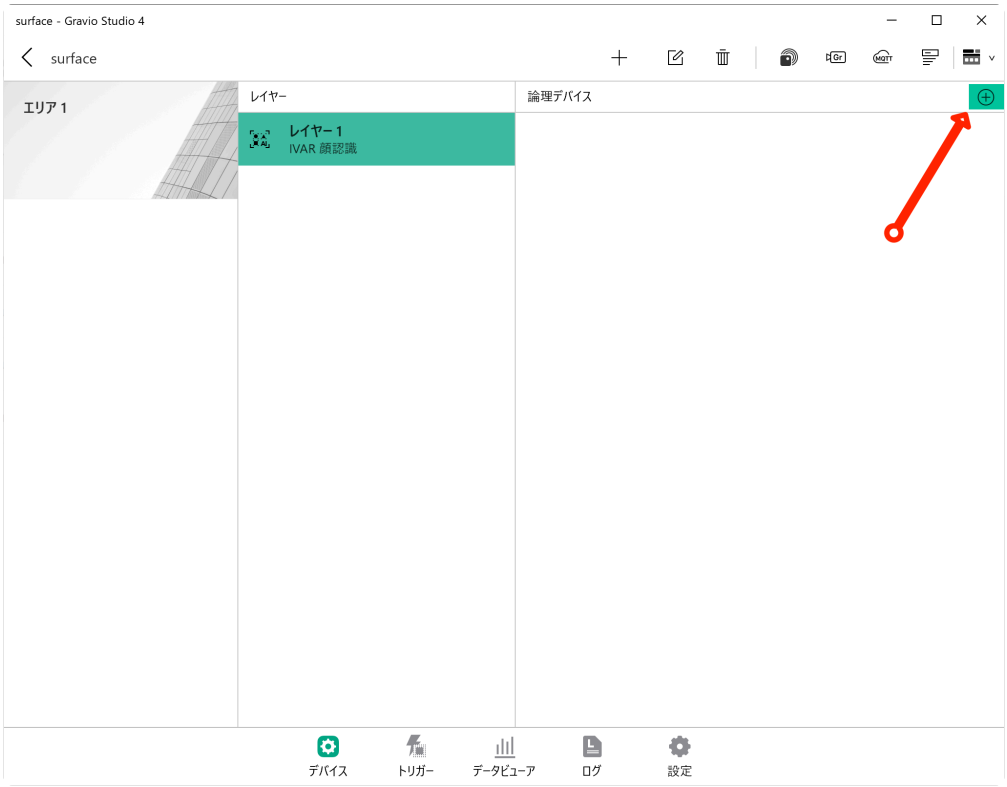
The URL of the HubKit that you want to register to the Gorilla IVAR server should be copied to the clipboard and registered to IVAR. For more information on how to register, please refer to the Coordinator's Setting up Gravio and IVAR integration.



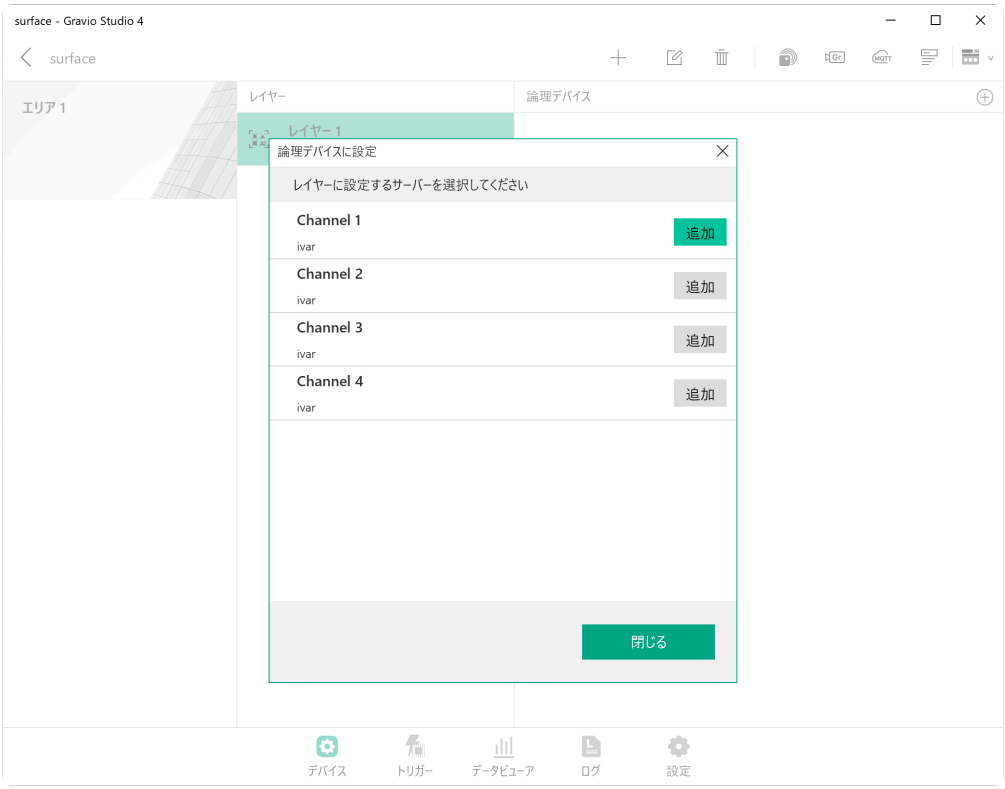
The saved IVAR information will be displayed in the list.



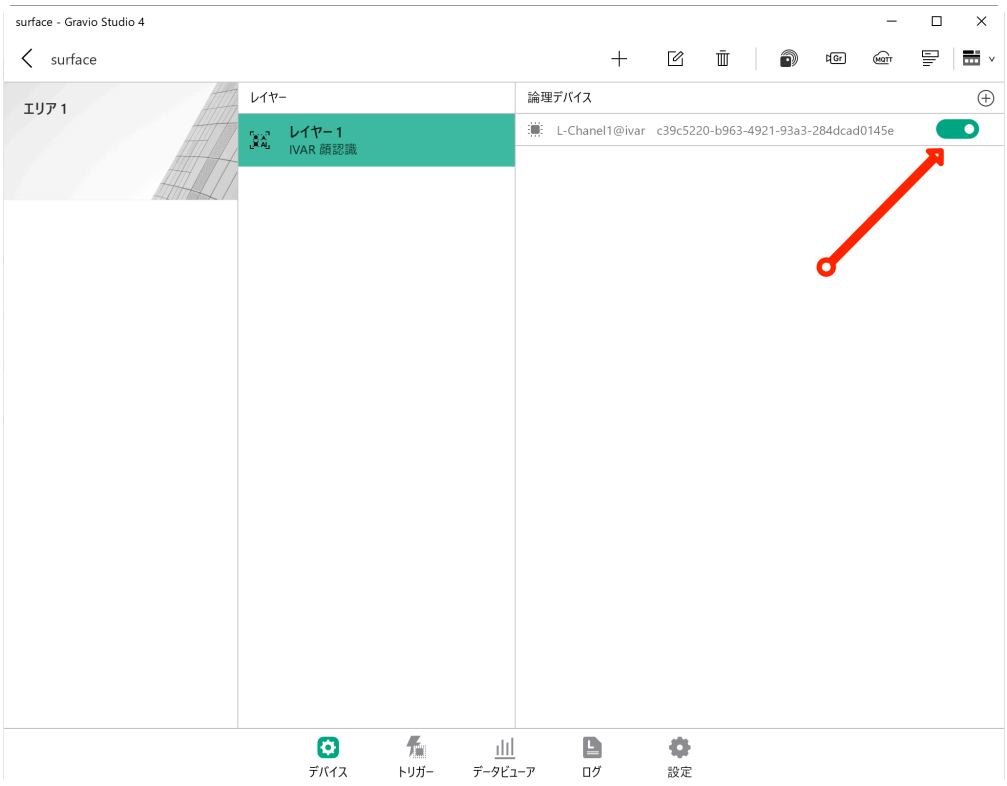
Click the + button to display the Bind Settings screen.



1. Bind the configured IVAR channel to the layer.



1. Finally, turn on the sensor acquisition.

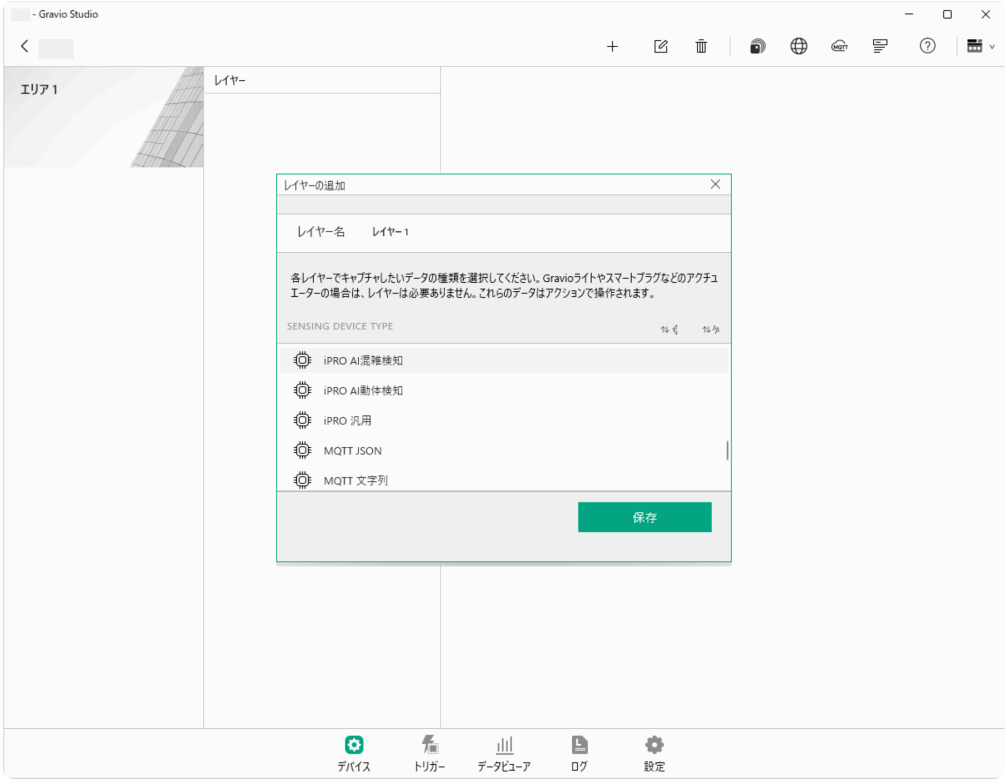


6.1.1.5. i-PRO Computer Vision and Gravio Integration

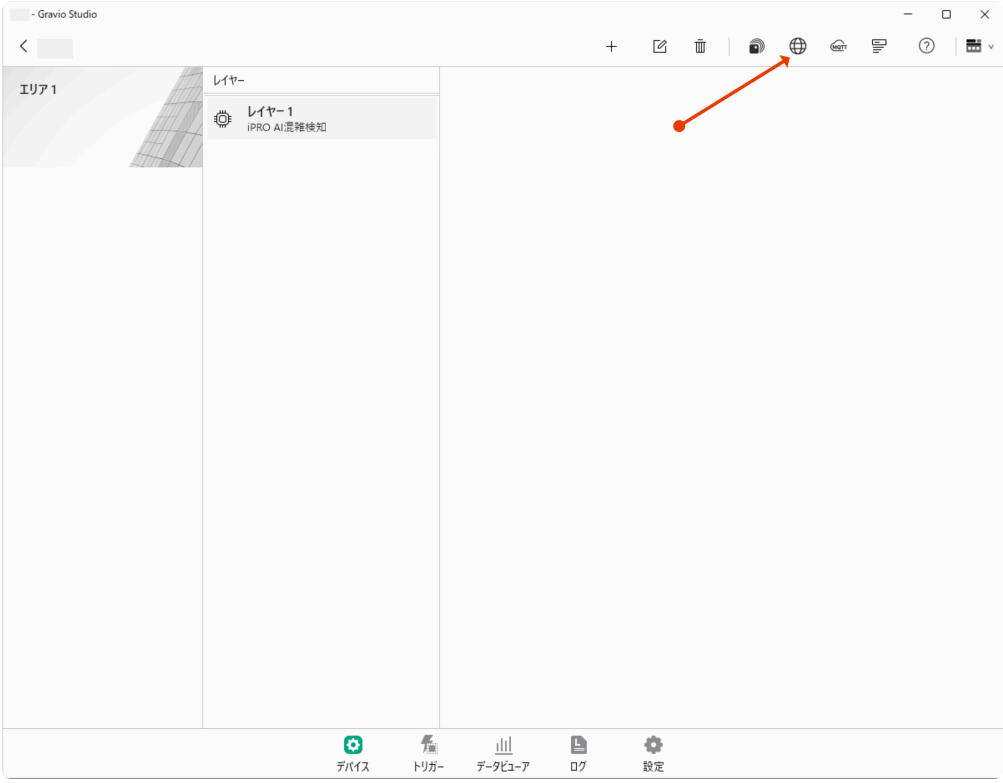
i-PRO is a surveillance camera sold by i-PRO Corporation , which is part of Panasonic. Gravio coordinates with i-PRO and receives data recognized by AI.

Setting up i-PRO

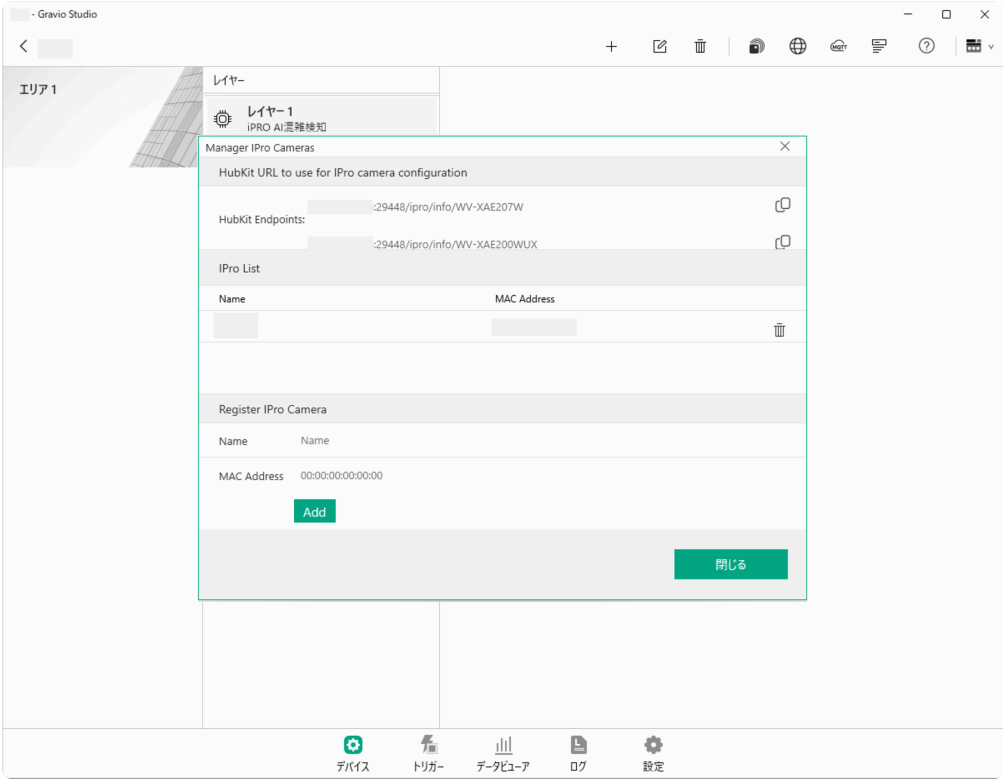
1. In ‘Devices’, select the type of i-PRO (DataKind) and add a new area and layer.
i-PRO chooses from AI crowd detection facial recognition, AI motion detection, and general purpose for the format of the data it receives.



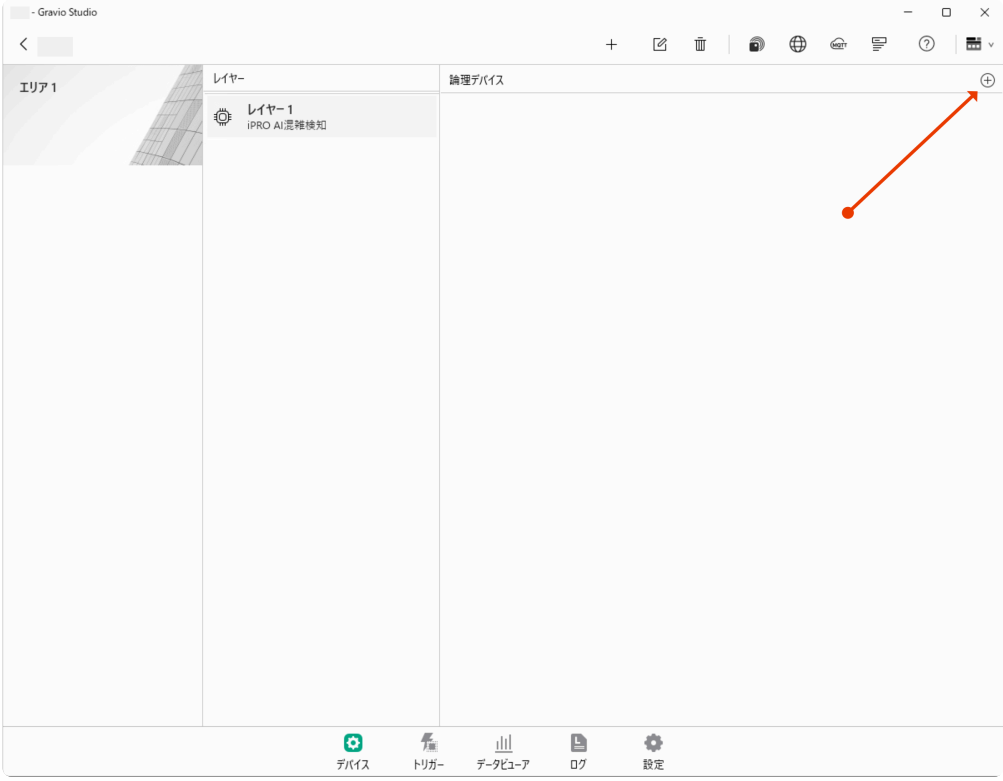
2. Press the following button to display the i-PRO settings screen.



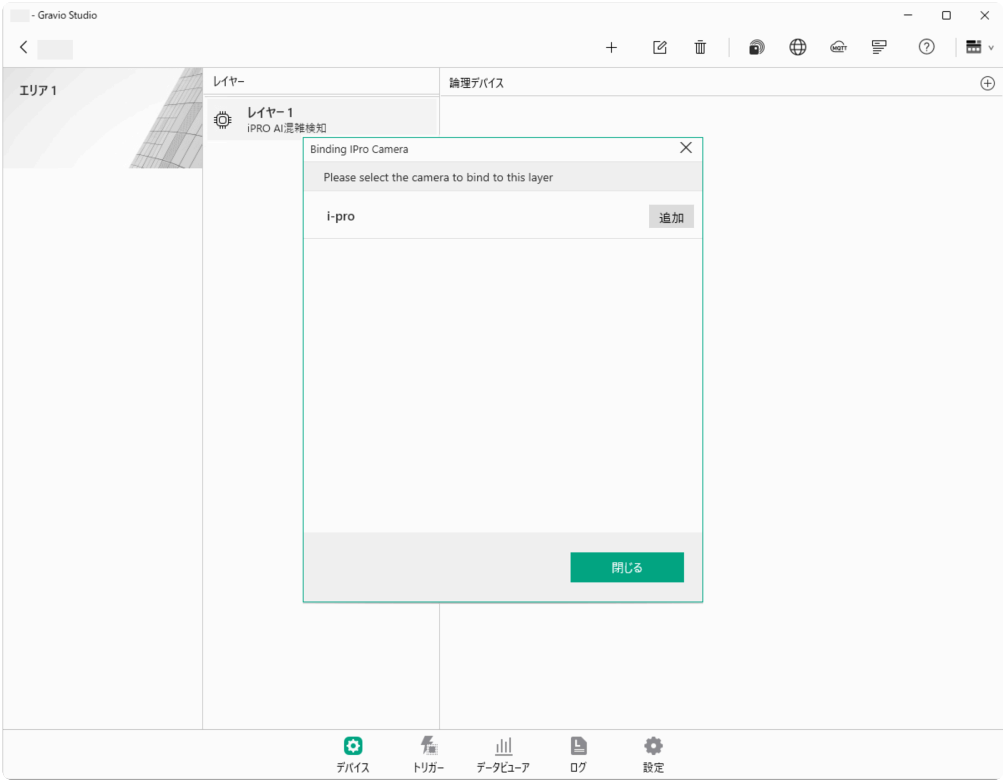
3. Enter the name of the i-PRO camera and the Mac address, and press the add button. The URL of the HubKit to register with i-PRO is copied to the clipboard and registered with i-PRO.
For the registration method, please refer to [How to set up i-PRO camera](#).



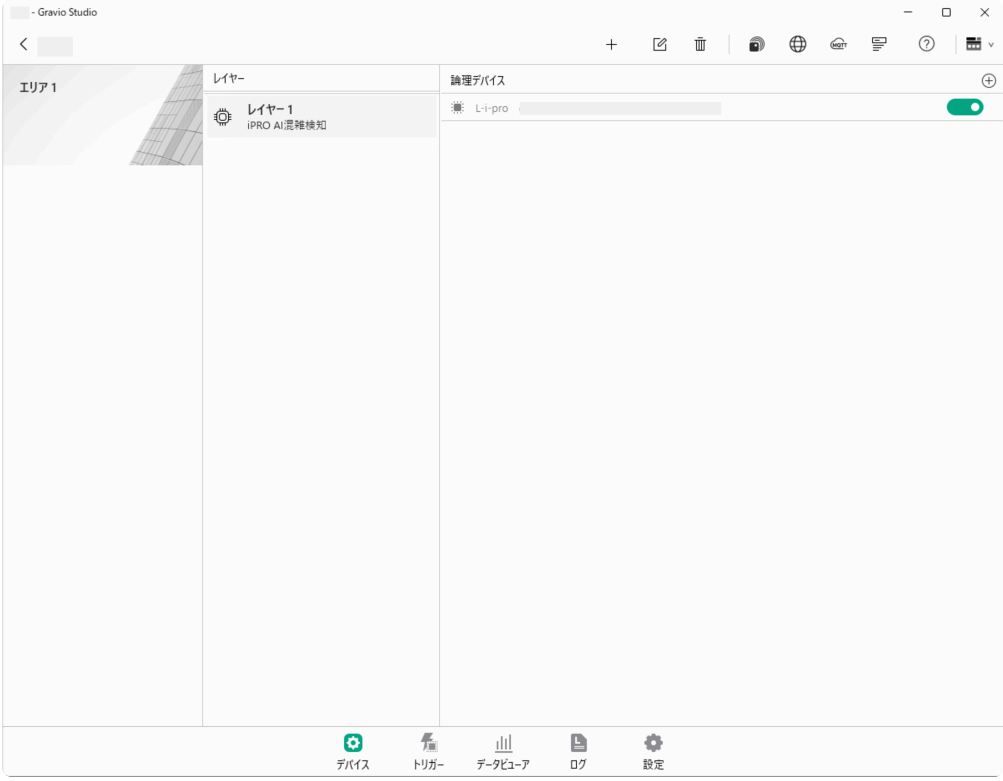
4. Click the ＋ button to display the binding settings screen.



5. Bind the set i-PRO camera to the layer.



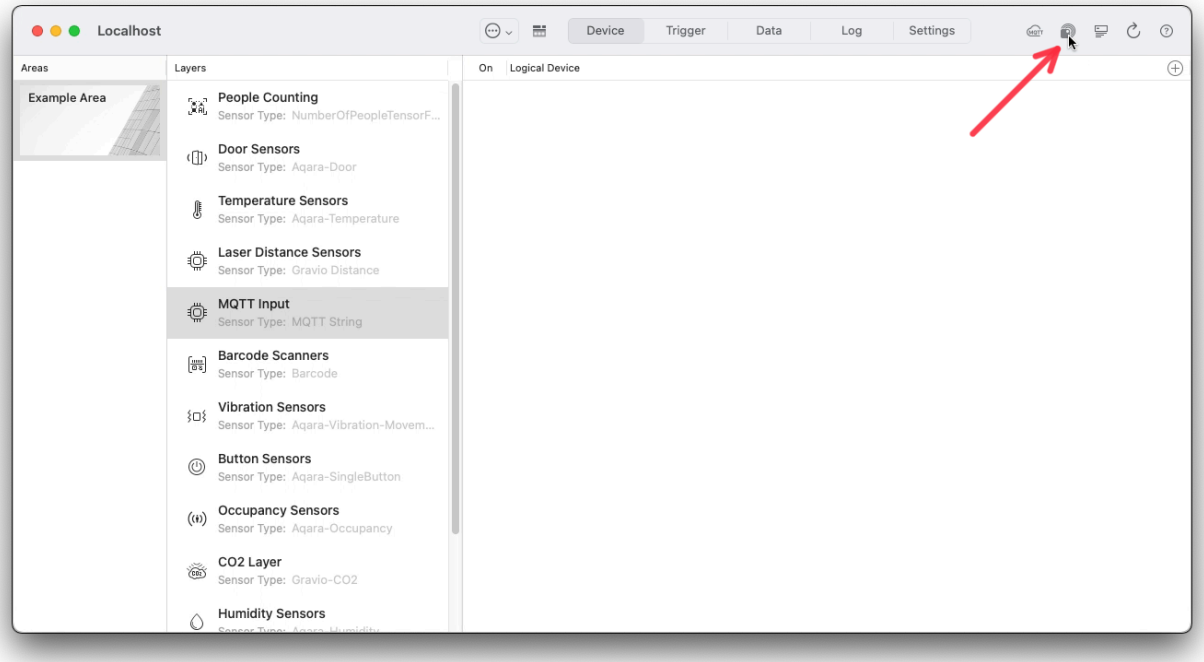
6. Finally, turn ON sensor acquisition.


- Gravio Studio
エリア 1
レイヤー
レイヤー 1
iPRO AI混雑検知
論理デバイス
L-i-pro
デバイス
トリガー
データビューア
ログ
設定

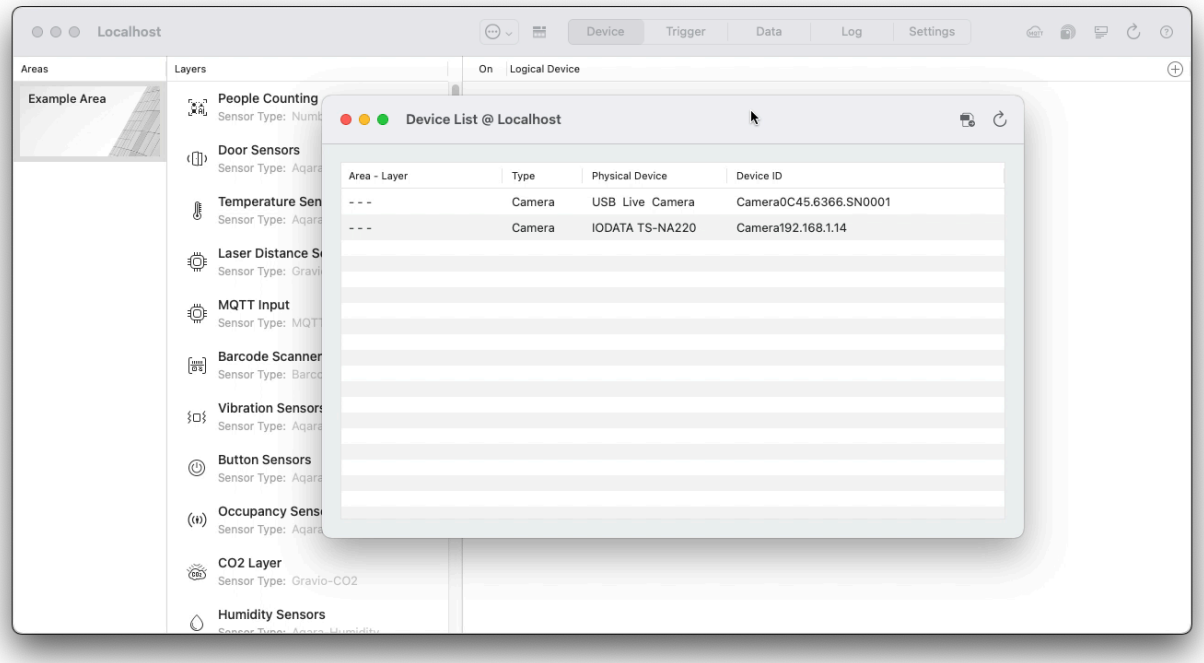
6.1.1.6. Physical Device List

Displays the list of physical devices connected to Gravio, and the list can be downloaded in a CSV format.

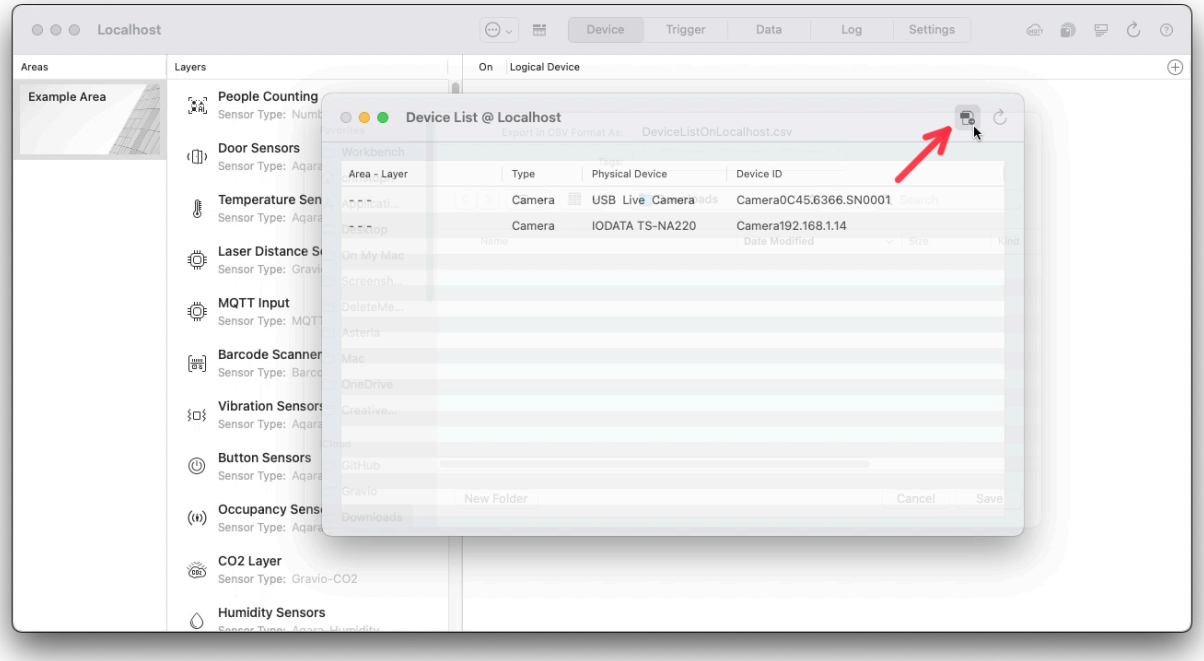
1. Press the following button to display the Physical Device List screen.



2. The area/layer to which the physical device is bound (empty if not bound), the physical device type, the physical device name, and the physical device ID will be displayed in the list.



3. Click the following button to output the physical device list to CSV.



6.1.2. Trigger Tab

In order to have a trigger, it makes sense to have an action first, that you want to trigger with your trigger. You can see how to make actions here.

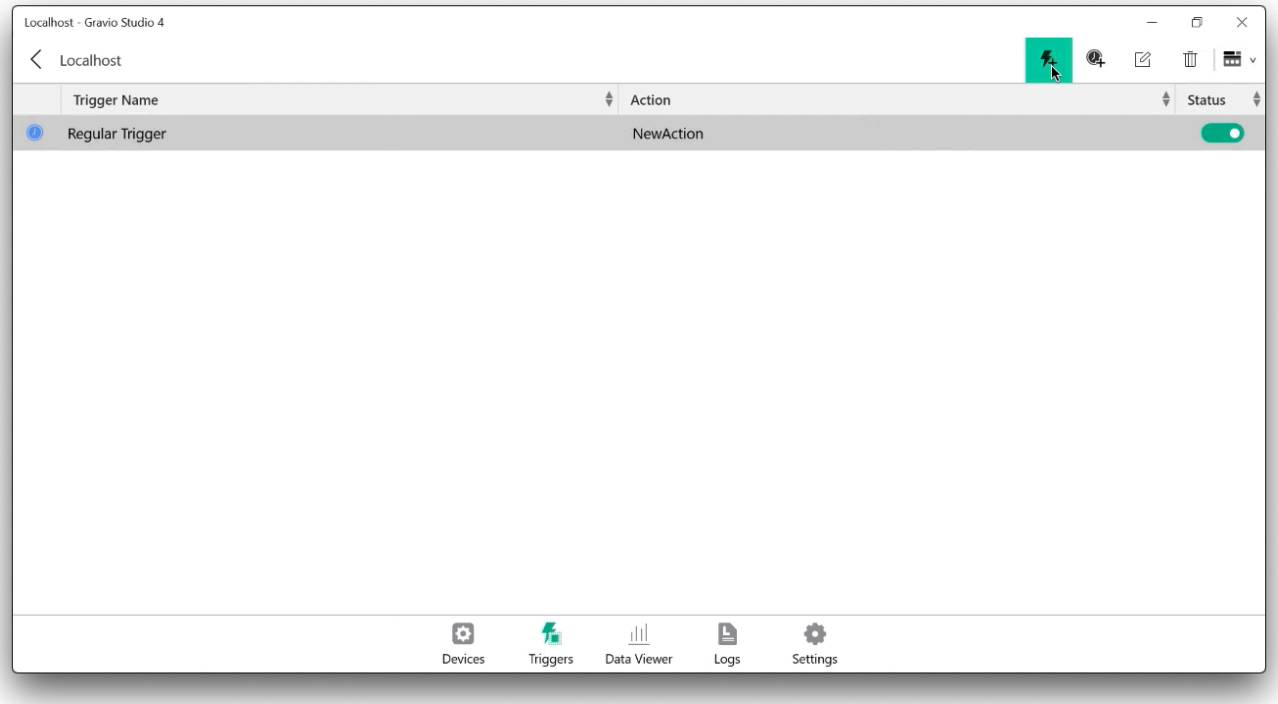
Once you have an action, you can start triggering it using sensor data. There are three types of triggers:

1. Triggered by events identified by sensor data
2. Triggered by time
3. Triggered by variable and property conditions

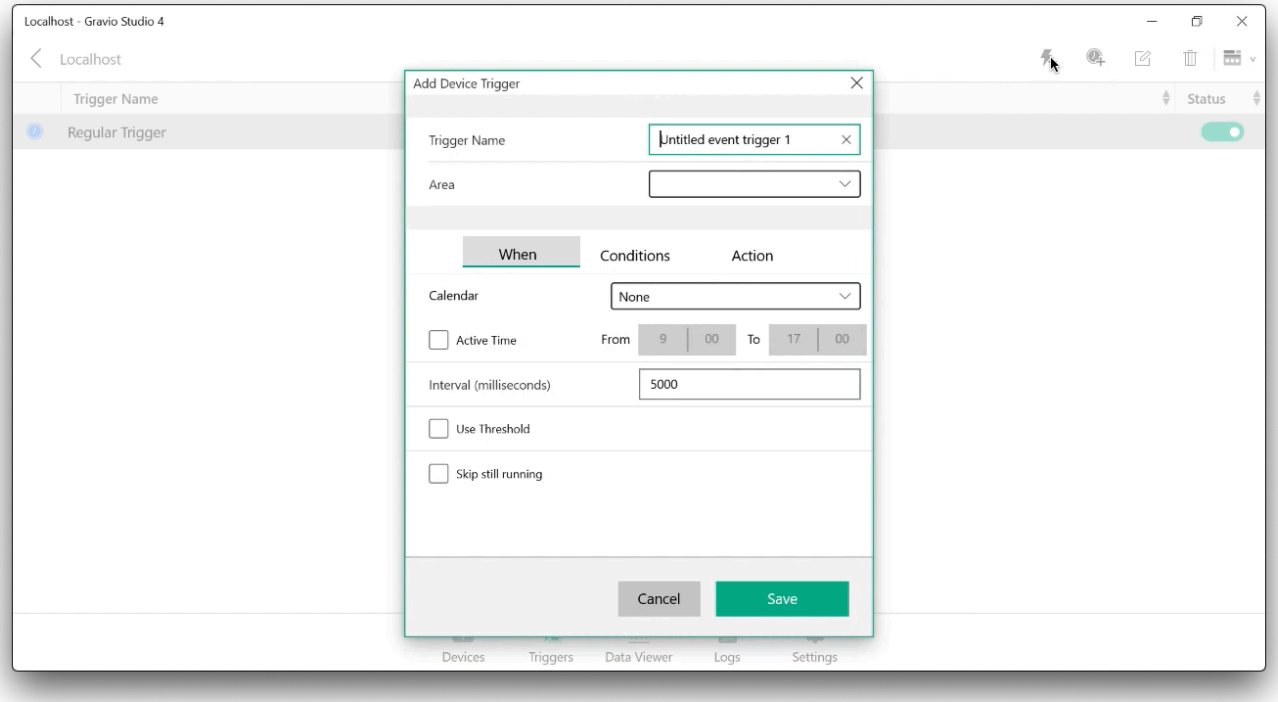
6.1.2.1. Sensor Event Triggers

Sensor Triggers

Click on the icon on the top right with the plus in the square to create a sensor-based trigger:



This will open the trigger popup:



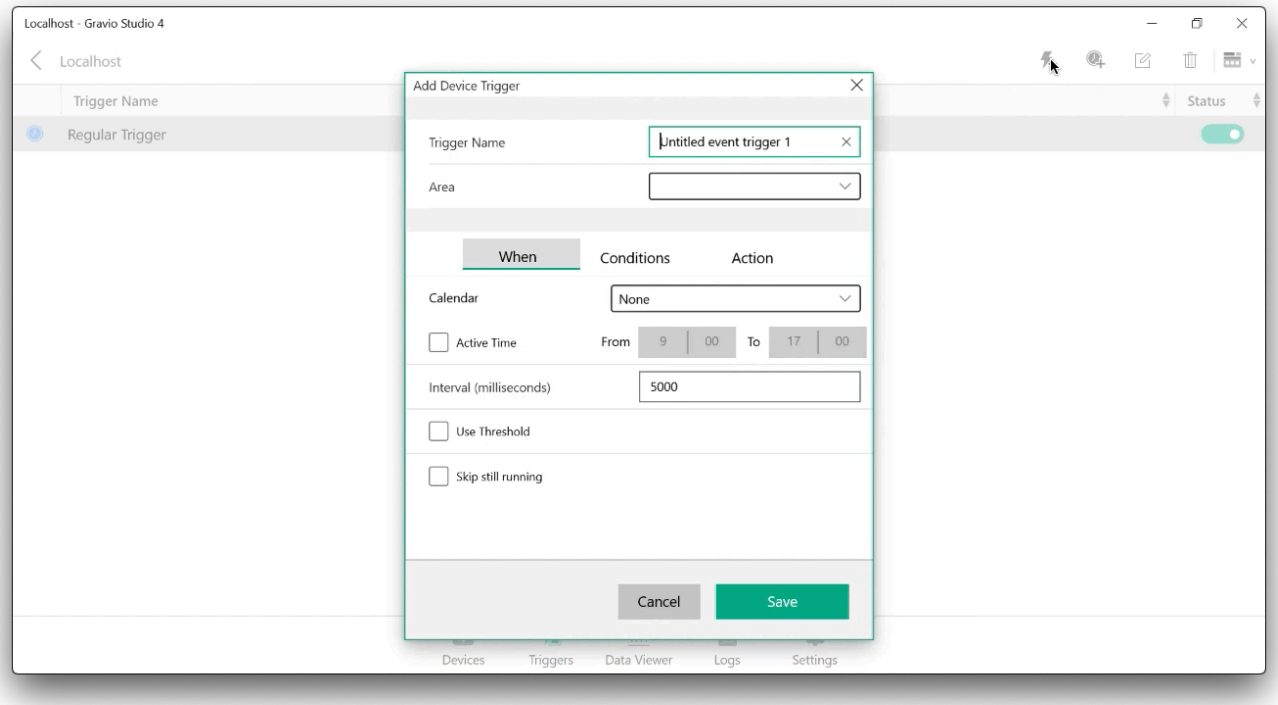
Give your trigger a sensible name, we recommend to name it after the condition that triggers the trigger.

Assign it to the correct area.

You have three tabs:

1. When: At which point in time the trigger should get triggered
2. Conditions: Under which conditions the trigger should get triggered
3. Action: Which action should be triggered and with which parameters

When



The Calendar function is reserved to enterprise customers who can create very detailed calendars in Gravio Coordinator. These calendars will then appear in this drop-down list.

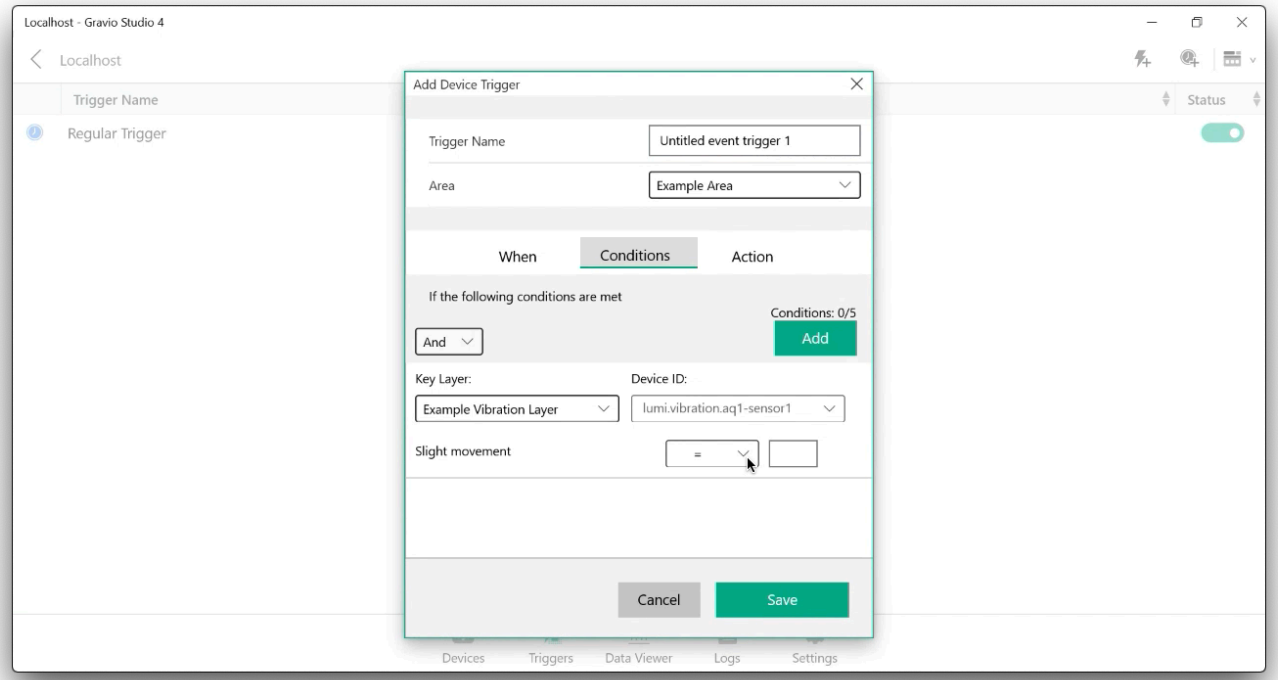
If you tick the “Active Time” the trigger will be active during the times specified here. If Active Time is not ticked, the trigger will trigger around the clock.

The interval is specified in seconds and determines how long the trigger should wait until it can be triggered again. This is useful if you for example don’t want a motion to be detected multiple times if the sensors send multiple signals in short succession.

The “Use Threshold” tickbox means that the trigger should only be triggered if the threshold of the sensor data is changed. For example, rather than triggering when reaching 20 degrees, you could tell the trigger to activate every time the temperature changes.

The tickbox “Skip still running” means that the trigger will only be activated if the associated action has finished.

Conditions

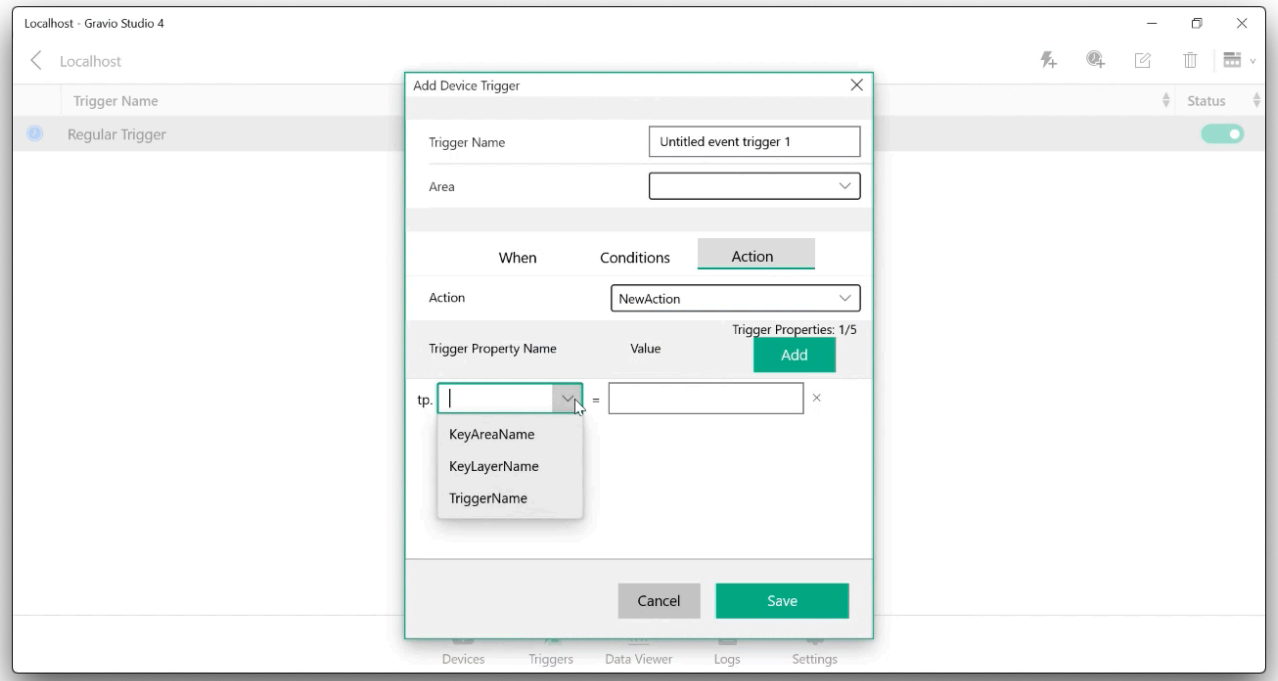


Under the Conditions tab you can set the conditions under which the trigger should be triggered. This is depending on the nature of the sensor layer type that you have picked. For example if it's a door sensor, you will have "open" and "close" as parameters. But if it's a temperature sensor you might have a temperature field and "more", "less", "equal", etc.

You can also combine conditions by choosing the "and" or "or" item in the dropdown and chaining the conditions up.

In the above example, it's a vibration sensor that is triggered.

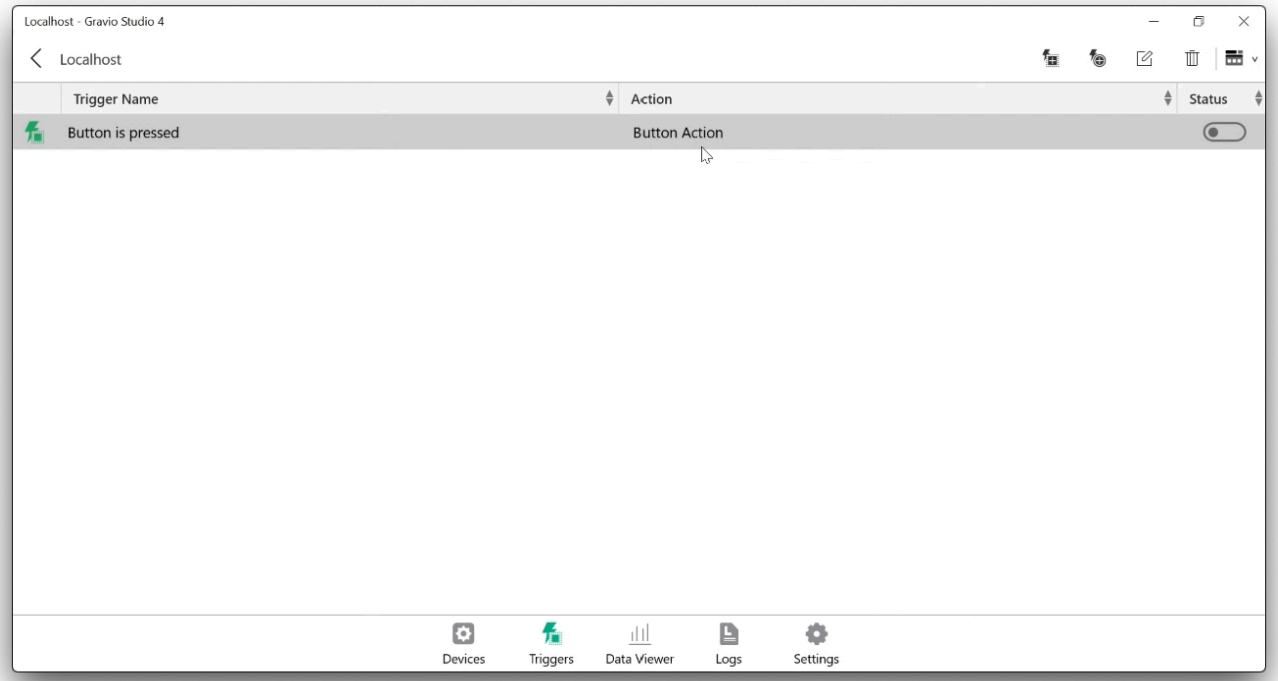
Action



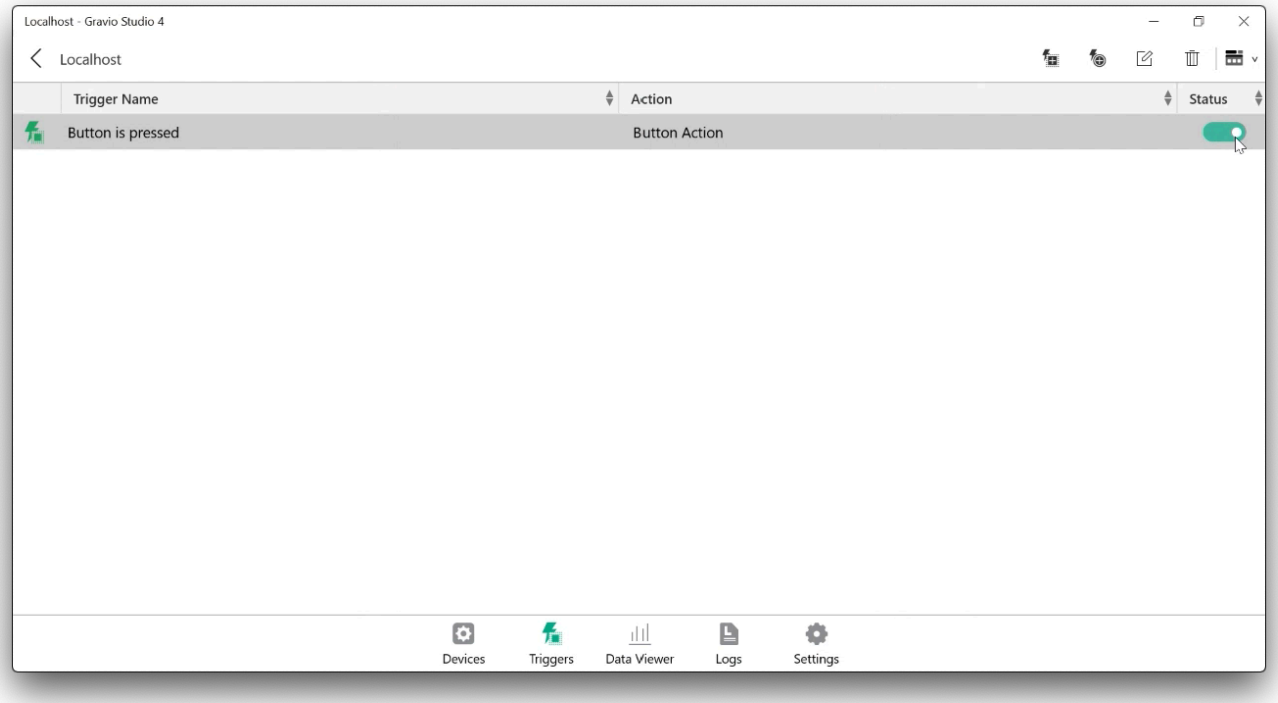
Finally, the “Action” section allows you to set the action that should be triggered. You need to have an action available for this. You can also pass arguments and values to those actions if required.

You can use trigger properties as parameters to pass on. You can find more details under [Trigger Variables and Properties]

After clicking the Save button, the trigger will appear in the triggers list:



Don’t forget to switch the trigger on on the far right side:



You’re done. Your sensors are now connected to the actions.

6.1.2.1.1. Trigger Parameter Examples

Setting the valid time period for triggers

Device triggers execute when data arrives, that matches the set conditions. You can specify an interval so that the next execution will not take place until this interval has elapsed from the previous execution time.
This is useful for example, if data arrives from the device once every 2 seconds, but you want the trigger to run every 10 seconds. In this case specify 9000ms (about 10% lower than 10 seconds) as the interval.

By doing so, the received being received every 2 seconds will be triggered at the following timings:

- Received data 1 (trigger execution)
- Received data 2 (no action)
- Received data 3 (no action)
- Received data 4 (no action)
- Received data 5 (no action)
- Received data 6 (trigger execution)
- Received data 7 (no action)
- …

Threshold

This parameter accommodates sensors that constantly transmit state data (e.g., temperature sensors, door/window open/close sensors, wireless switches).

Example 1: If a temperature sensor is set to trigger by threshold and the trigger is set to `>20°`, then after the sensor sends data above 20° and the trigger is executed, it will not be executed until the temperature drops below 20° and then above 20° again. If threshold triggering is off, the trigger will always be executed when the sensor sends data above 20°.

If the sensor sends data at 21°, 22°, 18°, and 23°:

- Threshold trigger is set, it will be executed at: 21° and 23°
- If threshold trigger is not set, it will executed at: 21°, 22°, and 23°

Example 2: If the threshold triggering setting is on for a door/window sensor and the trigger is set to `close`, the sensor will send close data and the trigger will not be executed until it is opened once. If threshold is set to off, the trigger will always be executed when the sensor sends close data.

If the sensor sends close, close, open, close:

- Threshold trigger is set, it will be executed at: close, close
- If threshold trigger is not set, it will executed at: close, close, close

Example 3: If threshold triggering is on for a wireless button, and the trigger is set to `click`, the trigger will only be executed if the state changes. If the threshold is set to off, the trigger will always execute

upon a receipt of a click.

If the sensor sends click, click, double click, click:

- Threshold trigger is set, it will be executed at: click, click
- If threshold trigger is not set, it will executed at: click, click, click

Setting trigger execution conditions

Key Layer and Layers

The condition in a Key Layer needs to be specified. When the condition is met, the respective action is executed. You can have additional layers and conditions that can be connected via AND or OR condition.

Example 1: When the trigger condition of the door/window sensor is set to `close` in the key layer

1. Receive data `open` from door/window sensor.
2. Receive the data `close` of the door/window sensor << **Trigger is executed here**
3. Receive data `open` of the door/window sensor
4. Receive data `close` from door/window sensor << **Trigger is executed here**

Example 2: When the key layer is set to `close` for the door/window sensor and the layer is set to `20° or higher` for the temperature sensor (if you have selected to use threshold triggering):

1. Receive data `open` from door/window open/close sensor
2. When the temperature sensor condition is set to `20°` or higher (when trigger by threshold is selected)
3. Temperature sensor data `20.0°` received
4. Receive door/window sensor data `open`
5. Receive data `19.8°` from temperature sensor
6. Receive data `close` from door/window sensor (trigger not executed)
7. Receive data `open` from door/window sensor
8. Temperature sensor data `20.1°` received
9. Receive data `close` of door/window sensor << **Trigger is executed here**
10. Receive temperature sensor data `20.2°`

Important Note: the two sensor data packets need to be received within the set interval time

Example 3: If the key layer is set to `close` for the door/window sensor and the layer is set to `20° or higher` for the temperature sensor (if you have not selected to use threshold triggering)

- Door/window sensor data `open` is received
- Door/window sensor data `close` is received (trigger is not executed)
- Temperature sensor data `20.0°` received
- Receive data `open` from door/window sensor
- Receive data `20.1°` from temperature sensor

- Receive data `close` from door/window sensor << **Trigger is executed here**
- Receive data `open` from door/window sensor
- Temperature sensor data `20.2°` received
- Receive data `close` from door/window sensor << **Trigger is executed here**
- Temperature sensor data `20.1°` received

Example 4: If the door/window sensor condition is set to `close` in the key layer, the temperature sensor condition is set to `20° or higher` in the layer, and the humidity sensor condition is set to `30% or higher` in the layer
(When **AND** is selected):

- Receive data `open` from door/window sensor.
- Receive data `close` from door/window open/close sensor (trigger is not executed)
- Temperature sensor data `20.0°` received
- Receive data `open` from door/window sensor
- Receive data `20.1°` from temperature sensor
- Receive data `close` from door/window sensor (no trigger is executed)
- Receive data `open` from door/window sensor
- Humidity sensor data `35%` received
- Receive data `close` of door/window sensor << **Trigger is executed here**
- Receive data `20.1°` from temperature sensor

Example 5: If the door/window sensor condition is set to `close` in the key layer, the temperature sensor condition is set to `20° or higher` in the layer, and the humidity sensor condition is set to `30% or higher` in the layer
(When **OR** is selected)

- Receive data `open` from door/window sensor
- Receive data `close` from door/window sensor (trigger is not executed)
- Temperature sensor data `20.0°` received
- Receive data `open` from door/window sensor
- Receive data `20.1°` from temperature sensor
- Receive data `close` from door/window sensor << **Trigger is executed here**
- Receive data `open` from door/window sensor
- Humidity sensor data `35%` received (no trigger is executed as door/window, the key layer, is still `open`)
- Receive data `close` from door/window sensor << **Trigger is executed here**
- Temperature sensor data `20.1°` received

Example Equations

You can also set an expression which has to be met in order to execute the trigger.

Example 1: When the temperature of the temperature sensor is greater than 25°.

Expression: `tv.Data >= 25`

Example 2: When the temperature of the temperature sensor is between 20 and 25 degrees Celsius

Expression: `tv.Data >= 20 && tv.Data <= 25`

Example 3: IVAR face recognition receives FR_UNKNOWN (For Gorilla Face Recognition module)

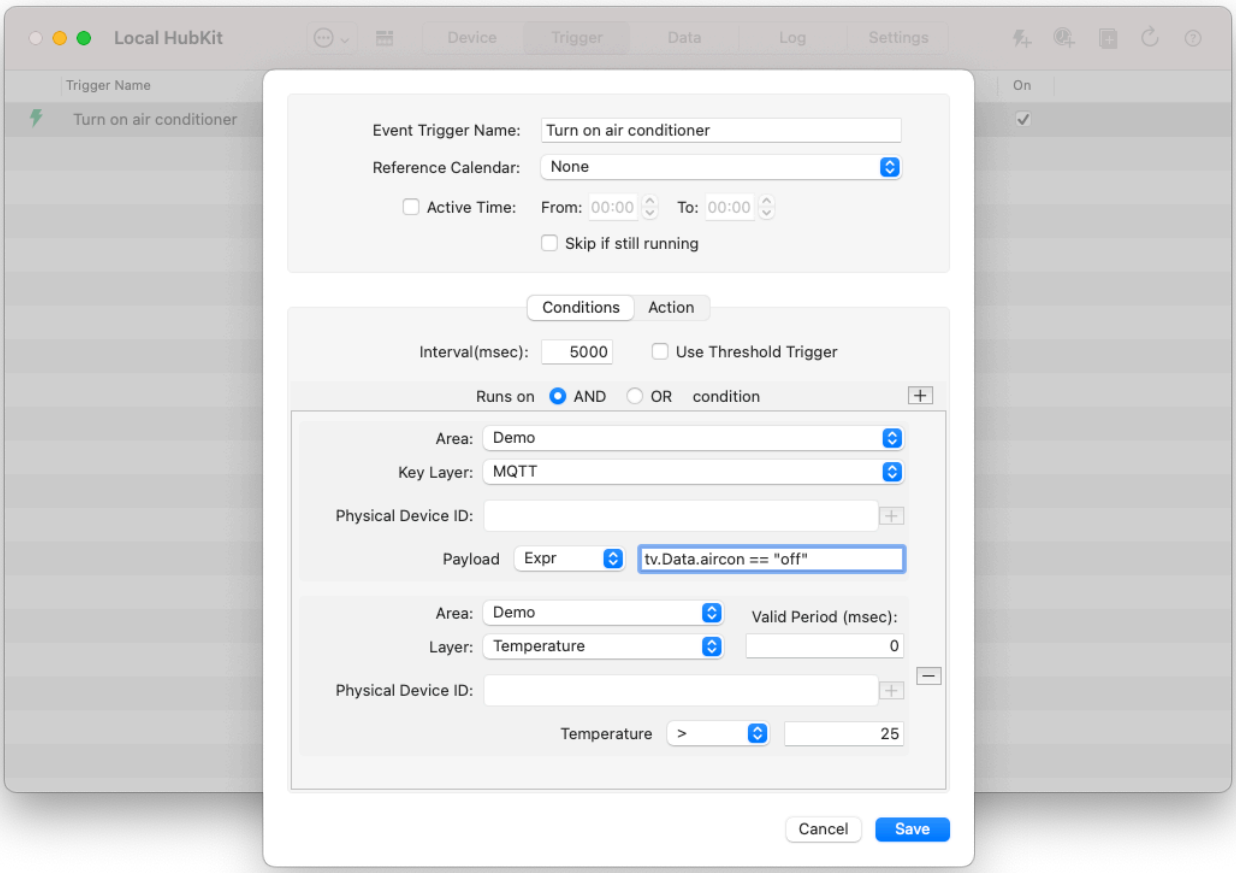
Event type: Any event
Expression: `tv.Data.common.type == "FR_UNKNOWN"`

Example 4: IVAR face recognition receives unmasked data

Event type: Any event
Expression: `tv.Data.iod.mask == "NONE"`

Example 4: If you have arbitrary data coming from MQTT (e.g. “aircon=off”), you can have that as a matching condition:

Event type: Any event
Expression: `tv.Data.aircon == "off"`



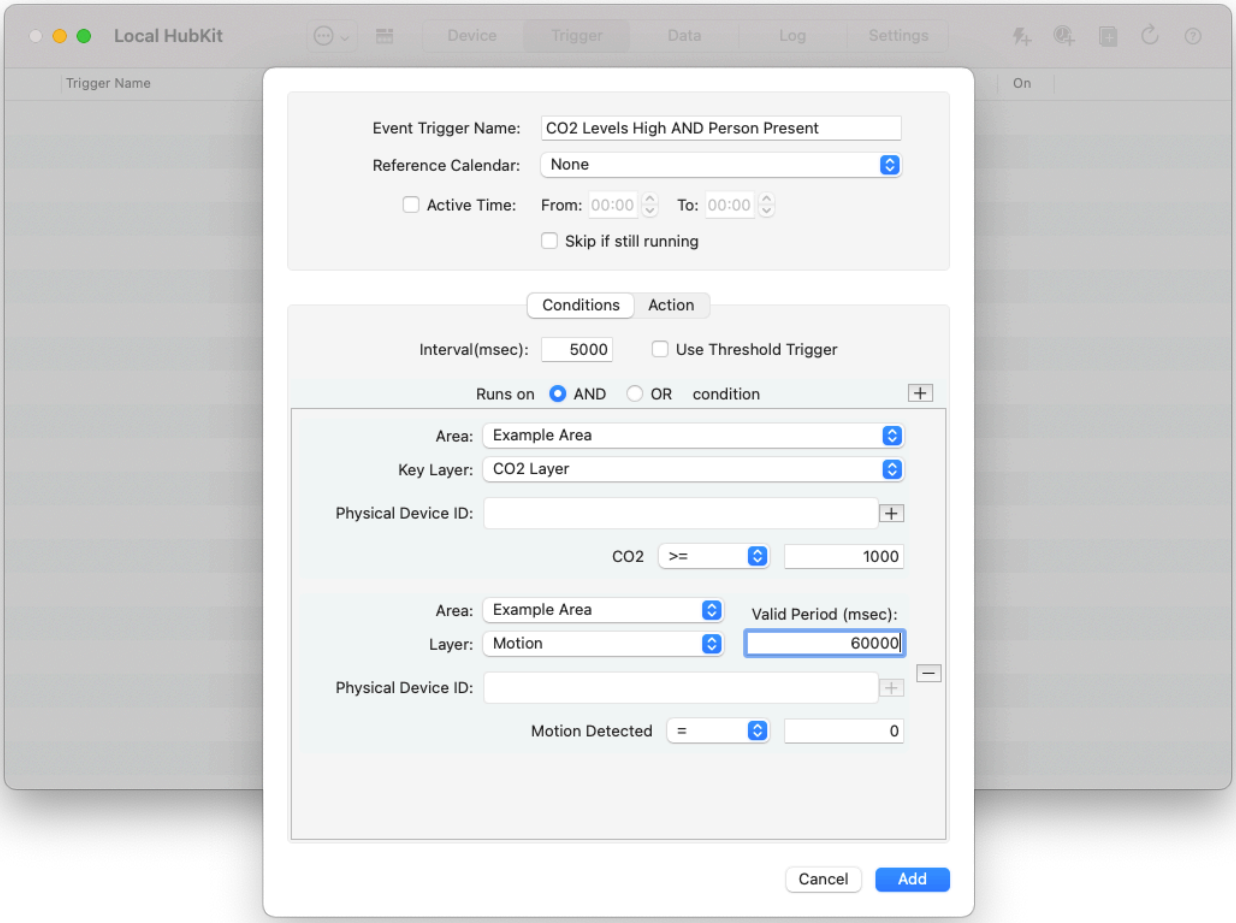
In this case, the trigger gets executed if the aircon data from MQTT is `off` and the temperature is higher than 25 degrees celsius.

Example of trigger conditions for key layers and layers with additions

Example: If you want to trigger only when CO2 exceeds 1000ppm **and** when there is a person in the room using the motion sensor.

- In the key layer, specify the condition when the CO2 layer value exceeds 1000ppm
- Add the motion sensor layer
- Since the motion sensor does not send any data when nobody is present, we need to limit the validity of the last value. In this example we set the “Validity Period” to 60,000 milliseconds (1 minute).
- This results in the trigger not being executed, even if CO2 exceeds 1000ppm, if no motion has been detected within the last minute.

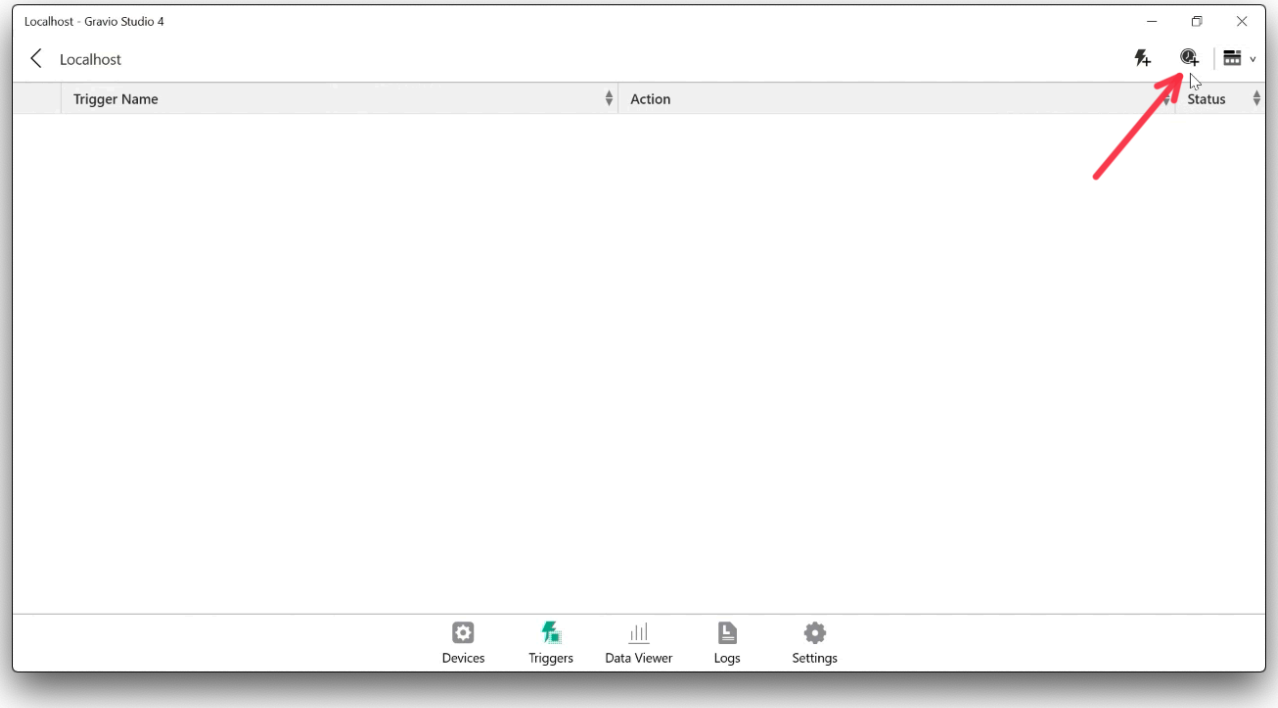
The “Validity Period” can also be set to `0`. In this case, the conditional judgement will be made based on the most recent value of the sensor.



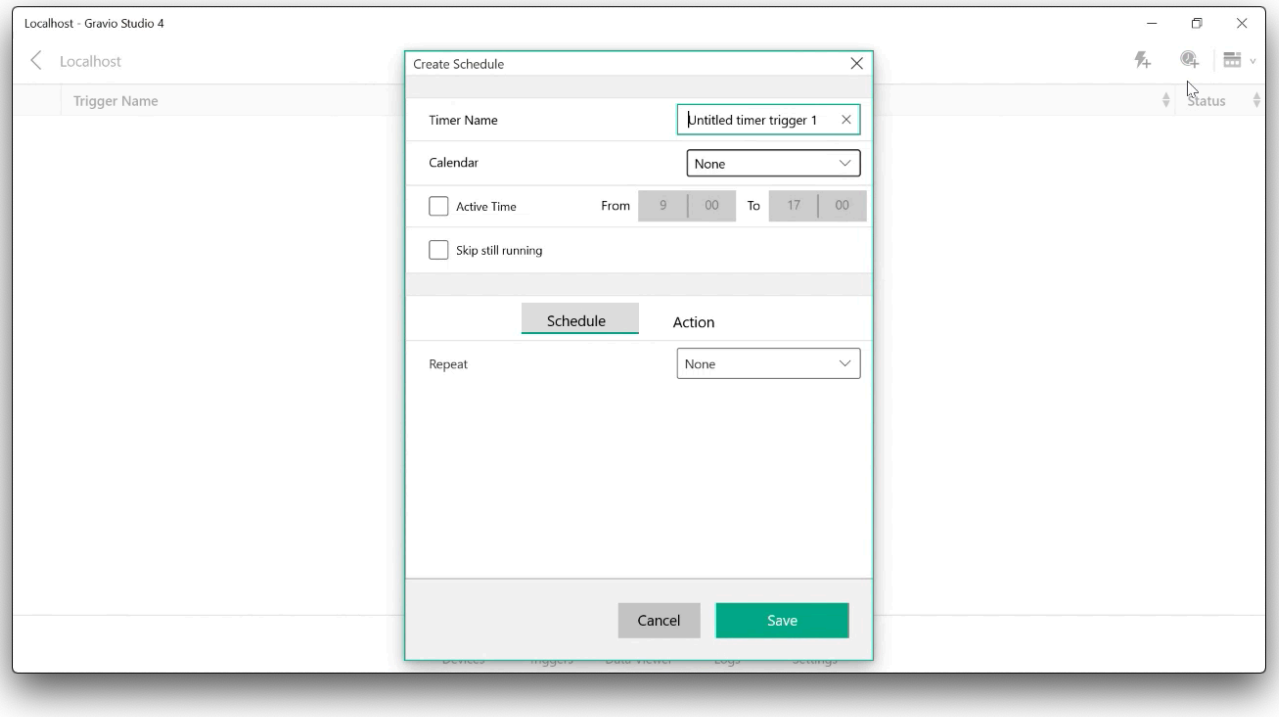
6.1.2.2. Time Based Triggers

You can create triggers that are based on times on time-related parameters such as intervals, weekdays, day-of-month, etc.

To create a time-based trigger, open the +-icon with the clock on the trigger tab:



You will see the following dialogue box:



Give your trigger a meaningful name.

If you are an enterprise customer, you can choose a calendar which you would like to use. You can create calendars in your Gravio Coordinator dashboard.

Select within which timeframe the trigger should be active.

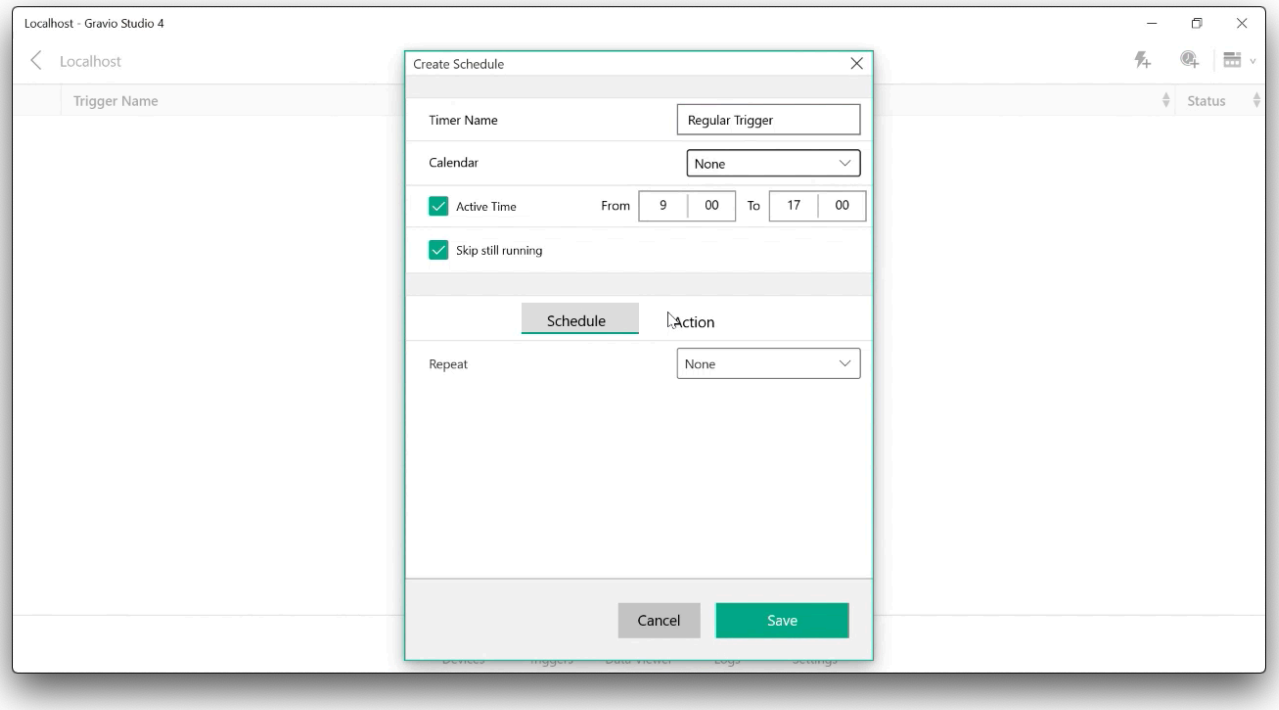
Tick the “Skip” tickbox if you want the trigger to skip if the trigger action is still running from a previous trigger.

You will see two Tabs: Schedule and Action

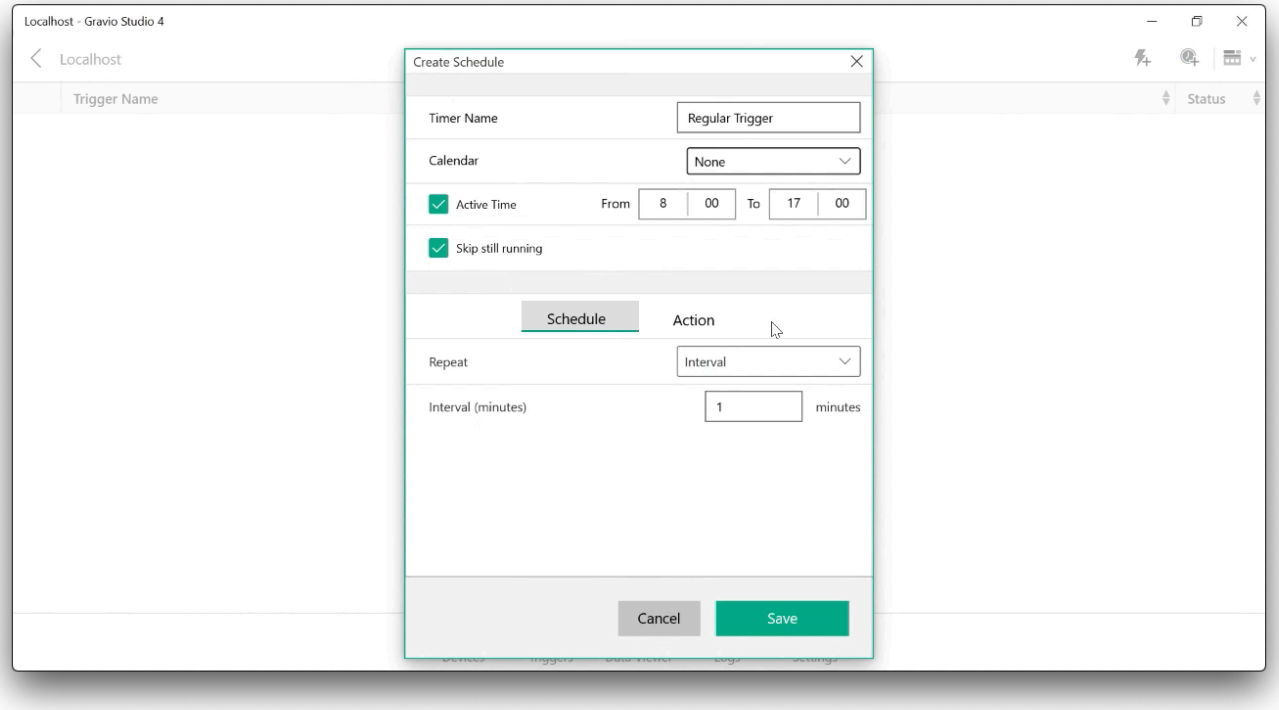
Schedule Tab

In the dropdown menu of Schedule, you have the following options:

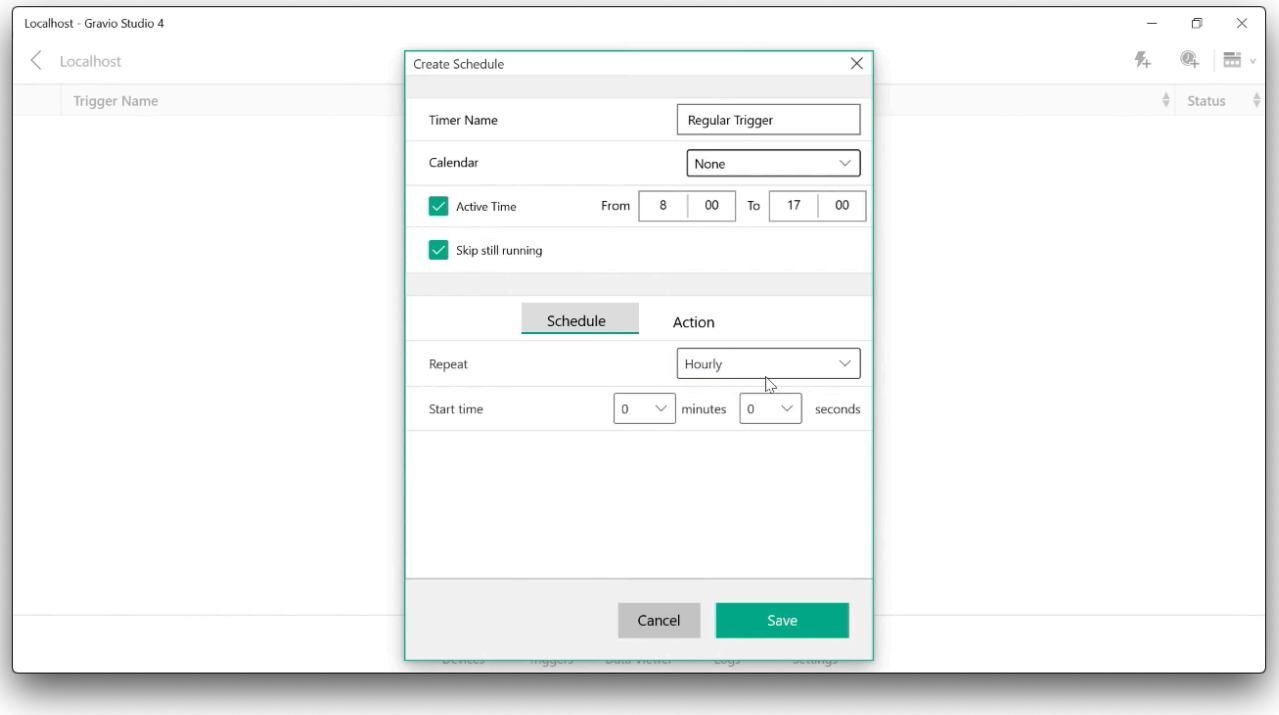
- None: If you there is no schedule



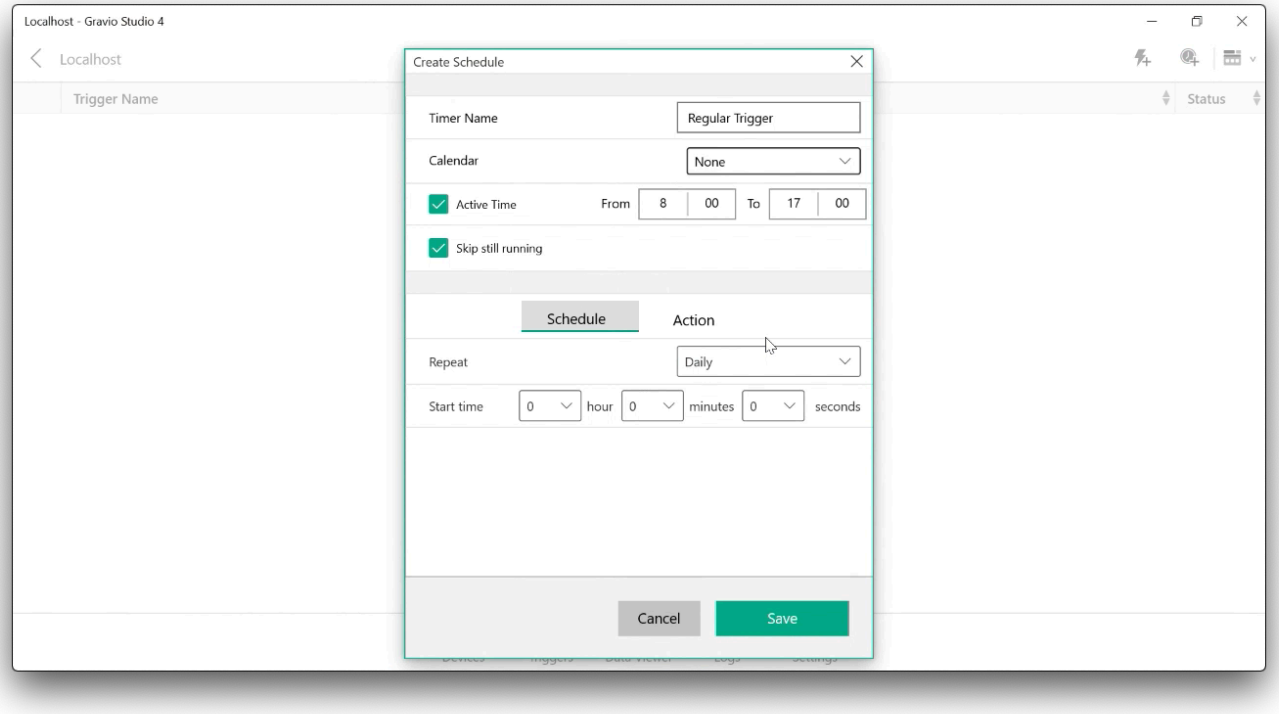
- Interval: if you want the trigger to fire in a certain interval (for example every 10 minutes)



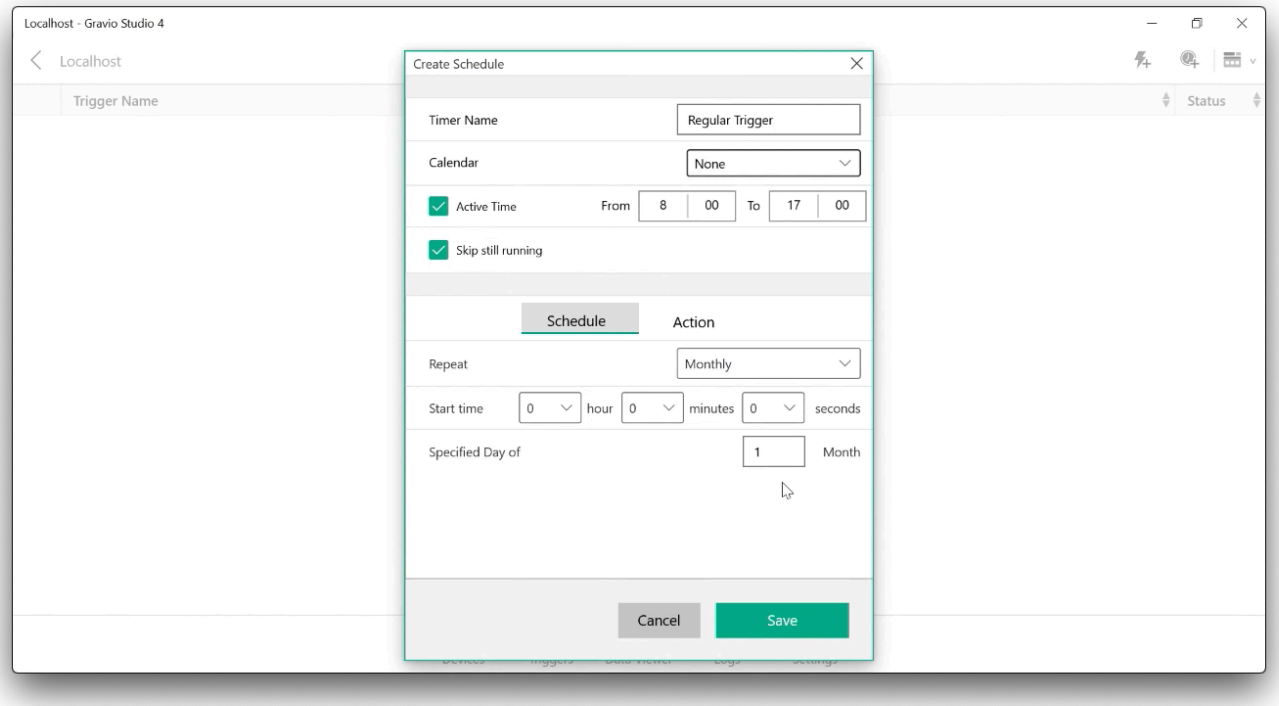
- Hourly: If you want the trigger to fire every hour on a specific minutes/seconds setting



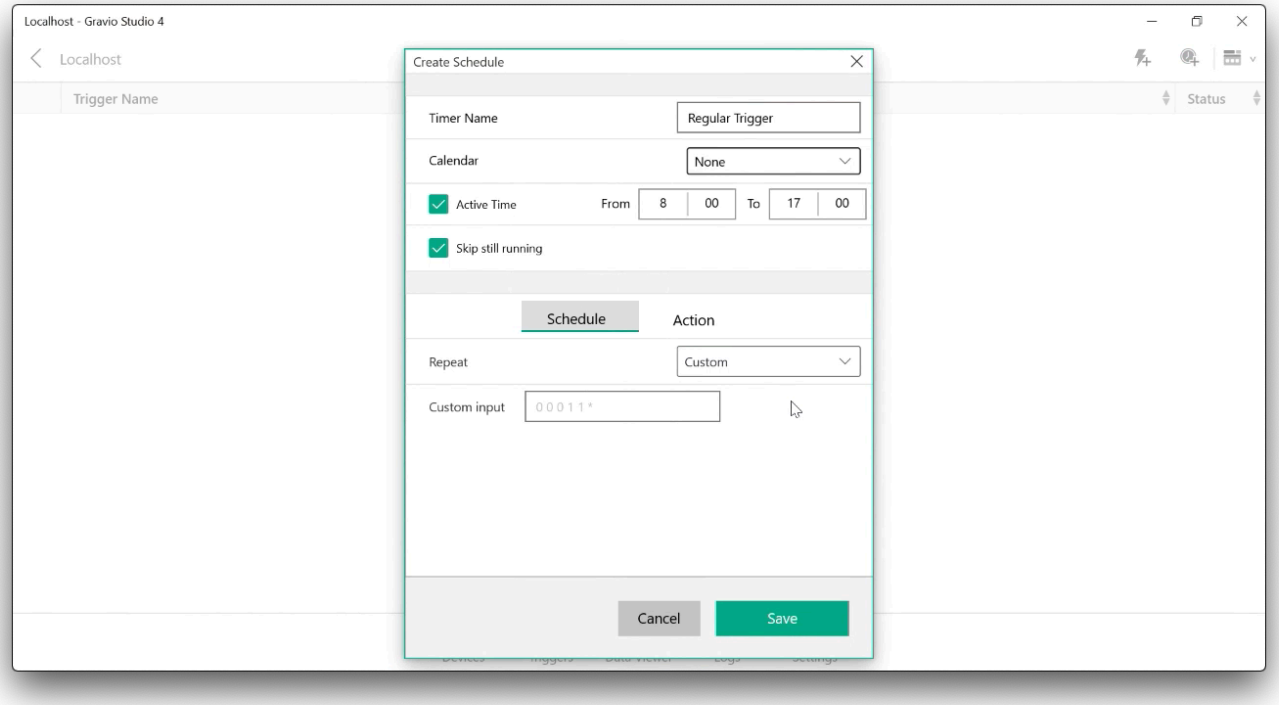
- Daily: If you want the trigger to fire on specific weekdays at specific times



- Monthly: If you want the trigger to fire on specific days of a month at specific times



- Custom: Use a syntax that is similar to crontab to fire a trigger at specific times



The positions are:

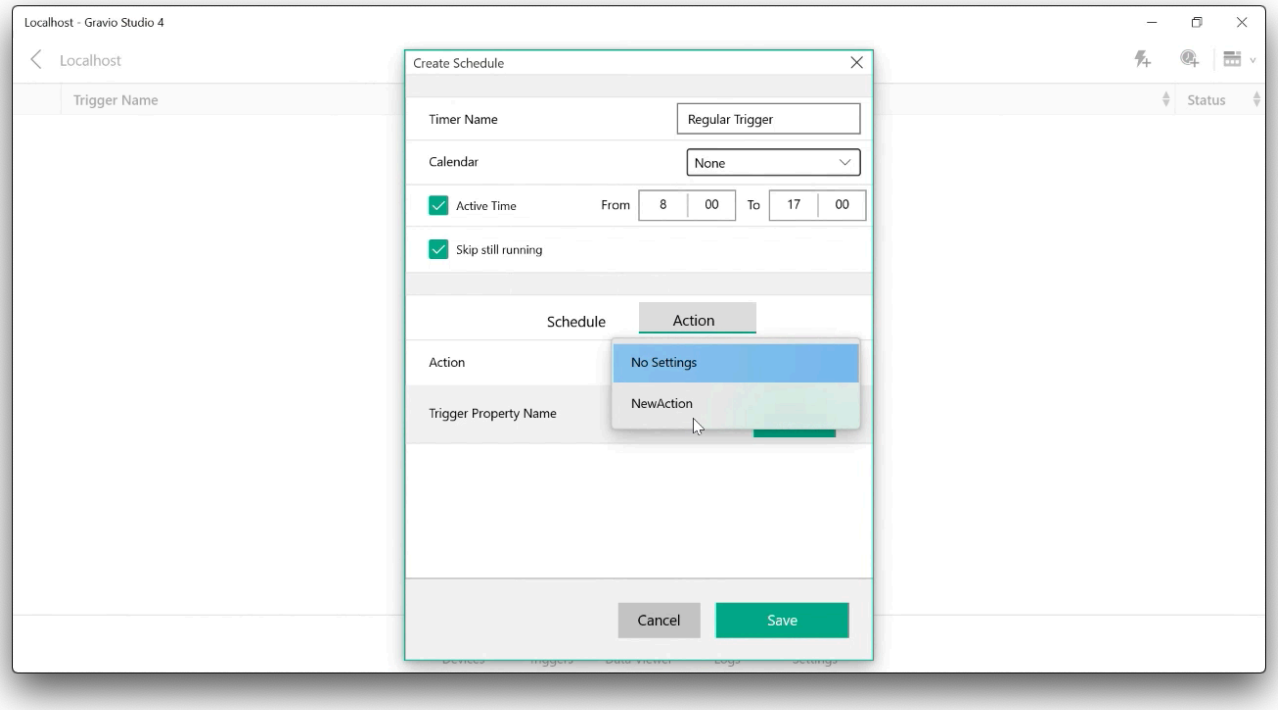
Unit	Required	Acceptable Values	Available special characters (see cron syntax)
Seconds	required	0-59	• / , -
Minutes	required	0-59	• / , -

Hour	required	0-23	• / , -
Day	required	1-31	• / , – ?
Month	required	1-12 or JAN-DEC	• / , -
Day of the week	required	0-6 or SUN-SAT	• / , – ?

Example: Once a year February 4th 14:23:51 “51 23 14 4 2 *”

Action Tab

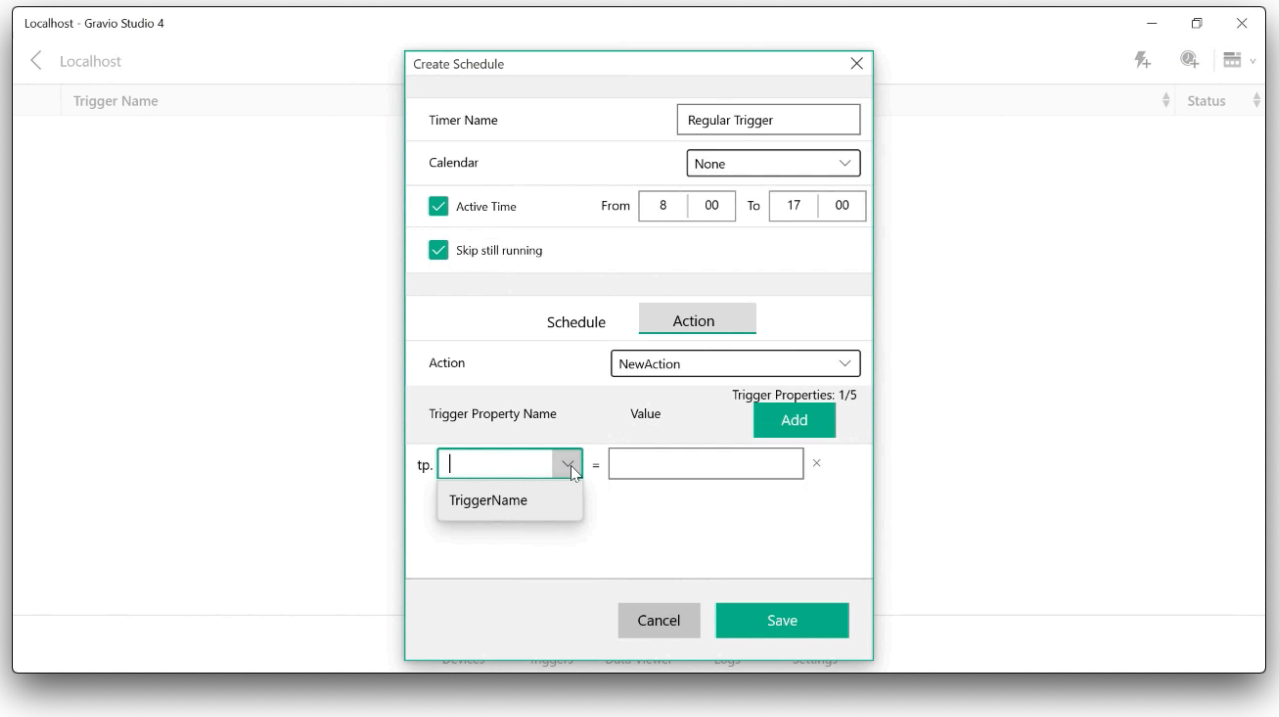
In the Action Tab you can pick which Action the trigger should fire. Please make sure you have built at least one action before you create a trigger, so you can choose your action in the dropdown menu:



You can also set up to 5 trigger properties as arguments when executing the action:

Trigger Properties can be set to the following three predefined properties and any property name:

`tp.KeyAreaName`: Sets the selected area name.
`tp.KeyLayerName`: Sets the first key layer name specified in the condition.
`tp.TriggerName`: Sets the name of the trigger to be executed.



By specifying an arbitrary property name and value other than the above three, that value can also be referenced without action.

Don’t forget to enable the new trigger by toggling the switch to green.

6.1.2.3. Trigger Variables and Properties

Device / timer triggers allow you to pass trigger properties and trigger variables as arguments when performing an action.

Trigger variable

The trigger variable is a variable in `tv.`, which is passed to the action as `tv.` In an event-driven trigger by receiving sensor data.

Trigger properties

The trigger property `tp.` Can be defined in the trigger settings dialog, and even if the same action is called, the behavior can be changed by using conditional branching by the value of `tp.`

Trigger properties can be set to the following three predefined properties and any property name.

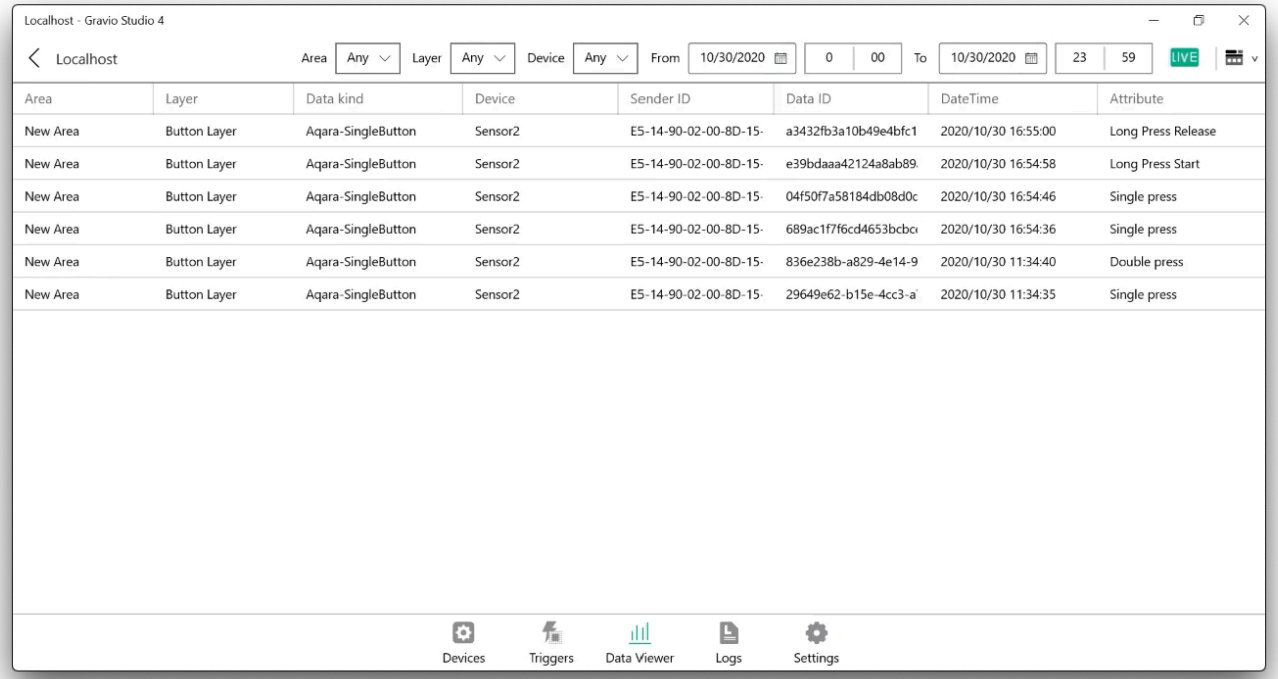
`tp.KeyAreaName`: Sets the selected area name.
`tp.KeyLayerName`: Sets the first key layer name specified in the condition.
`tp.TriggerName`: Sets the name of the trigger to be executed.

6.1.3. Data Viewer Tab

The Data Viewer Tab allows you to investigate incoming sensor data in real-time (press the “live” button on the top right for real-time live data). This is useful if you want to set up the system and test its functionality. Please click on the “Live” button on the top right to see the live incoming data. The view is updated every 5 seconds.

This view is also useful to identify the sensor’s IDs in the Sensor ID column.

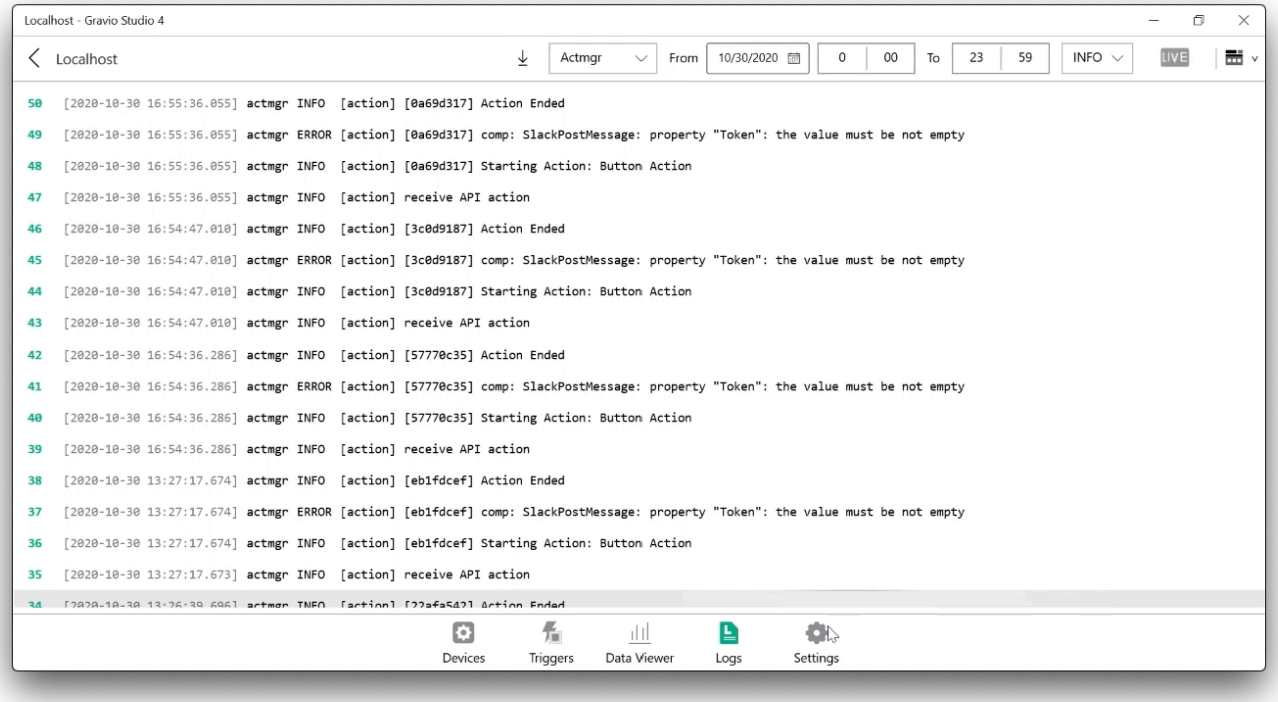
You can filter the data for various aspects such as timeframes, areas, layers or devices.



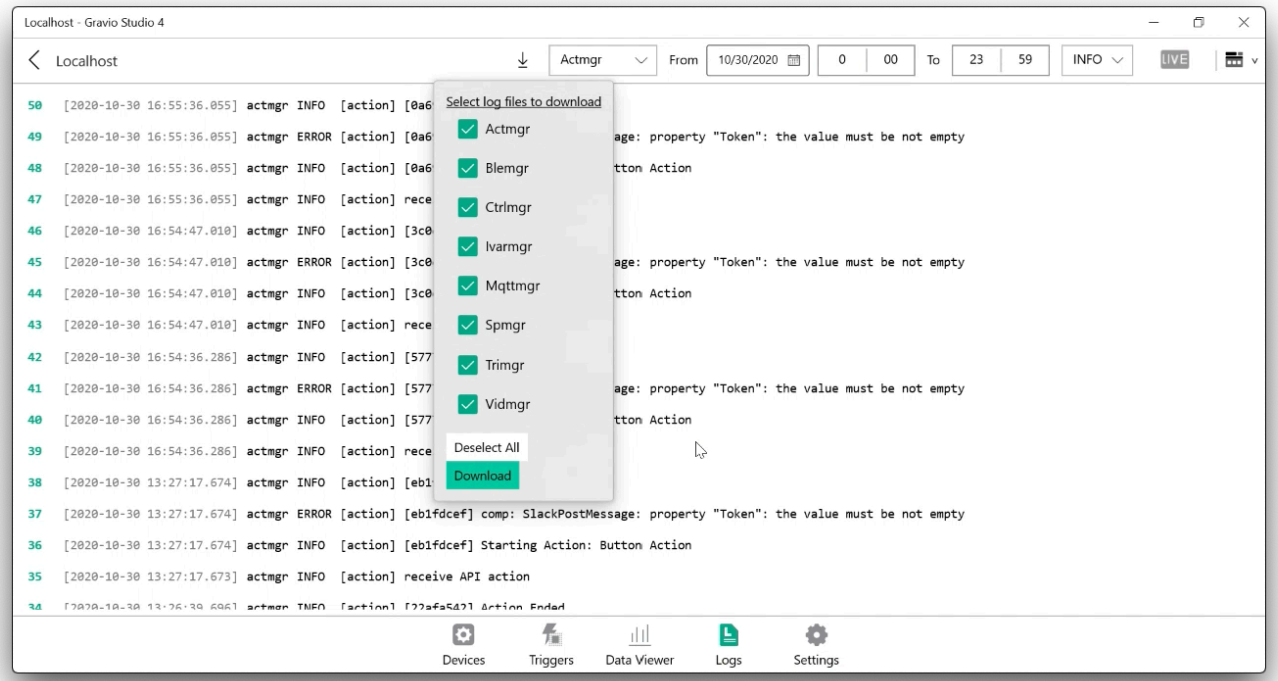
6.1.4. Log Viewer Tab

In the Log Viewer tab you can access or download the log files that Gravio produces. They are mainly used for debugging. The logs include:

- **Actmgr** – containing any logs the Actions and Steps produce
- **Blemgr** – Bluetooth related logs
- **Ctrlmgr** – Control manager related logs such as sensor management or maintenance tasks
- **Ivarmgr** – IVAR, the third party visual detection engine related logs
- **Mqttmgr** – Log events related to MQTT functionalities of Gravio
- **Spmgr** – Serial Port related log files
- **Trimgr** – Trigger related logs
- **Vidmgr** – Logs produced by the video manager



You can also download the logs by clicking on the downwards arrow.



You can also find the log files under

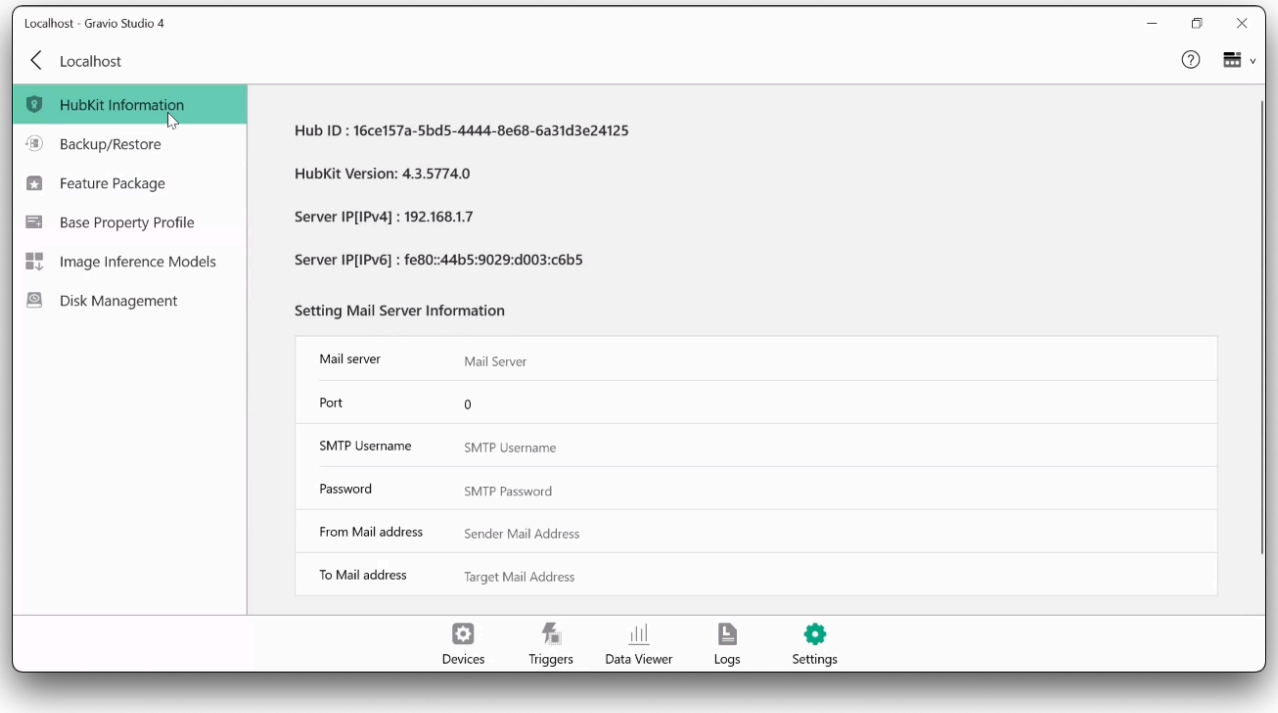
`/Library/Logs/HubKit/` on Mac

6.1.5. Settings Tab

In the Settings tab you can manage various settings aspects such as view the hubkit's versions, backup or restore, set the base property profiles, deploy image inference models or view the disk usage space. If you have the enterprise version, you can also connect to your Gravio Coordinator here and set up and manage the blockchain functionalities.

HubKit Information

Here you can see the various aspects of the HubKit that you are connected to. This information is also important when reporting bugs so the Asteria team know which version you are operating.

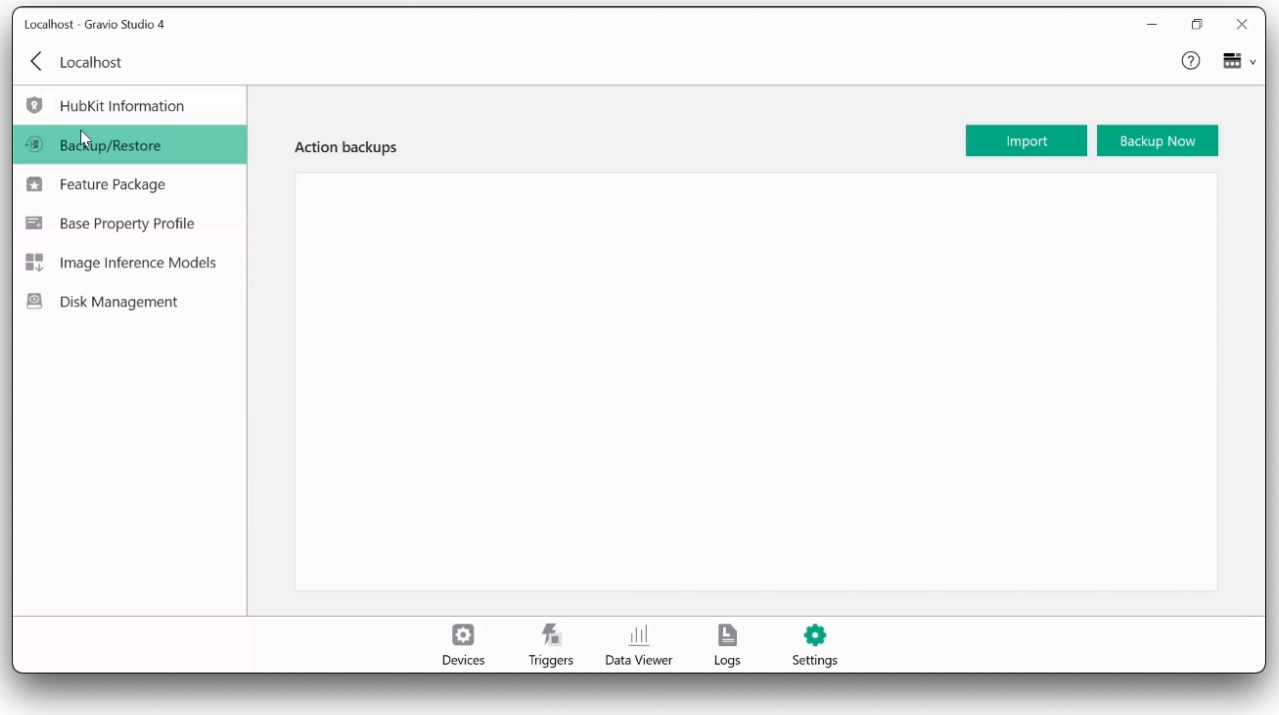


At the bottom you can also set the mail server parameters that will be used to send you notifications via e-mail. These notifications can be info about low disk space or issues with the hubkit.

[Gravio 4.5 or higher] – If periodic communication from the sensor of a Zigbee device (about 60 minutes) is not received, and if data is not received for a period of time in that state (90 minutes), the system will notify the email address set in the notification settings that communication from the device has been lost. CO2 is sent out when no signal is received for 1 minute.

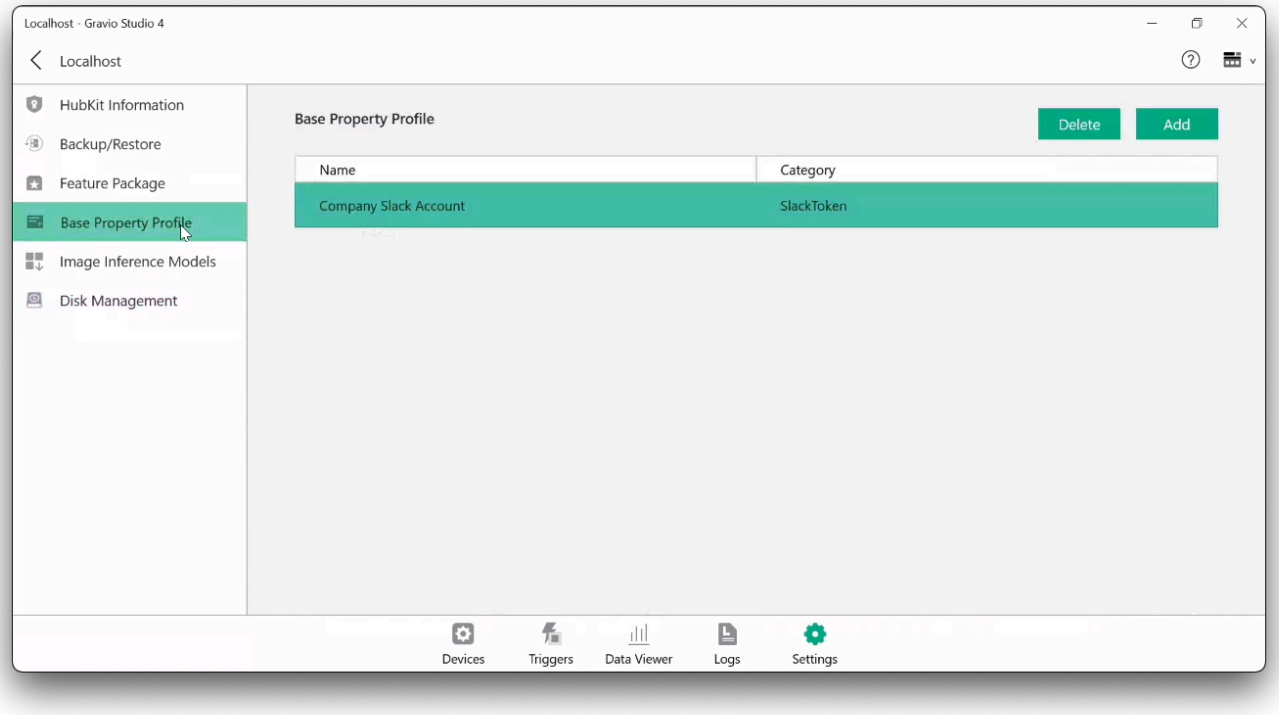
Backup & Restore

You can backup device configuration settings to your Gravio Cloud account, whether it's the Asteria hosted one or the one hosted on your own local Gravio Coordinator. From there you can restore these settings back into your home folder on your Gravio account.



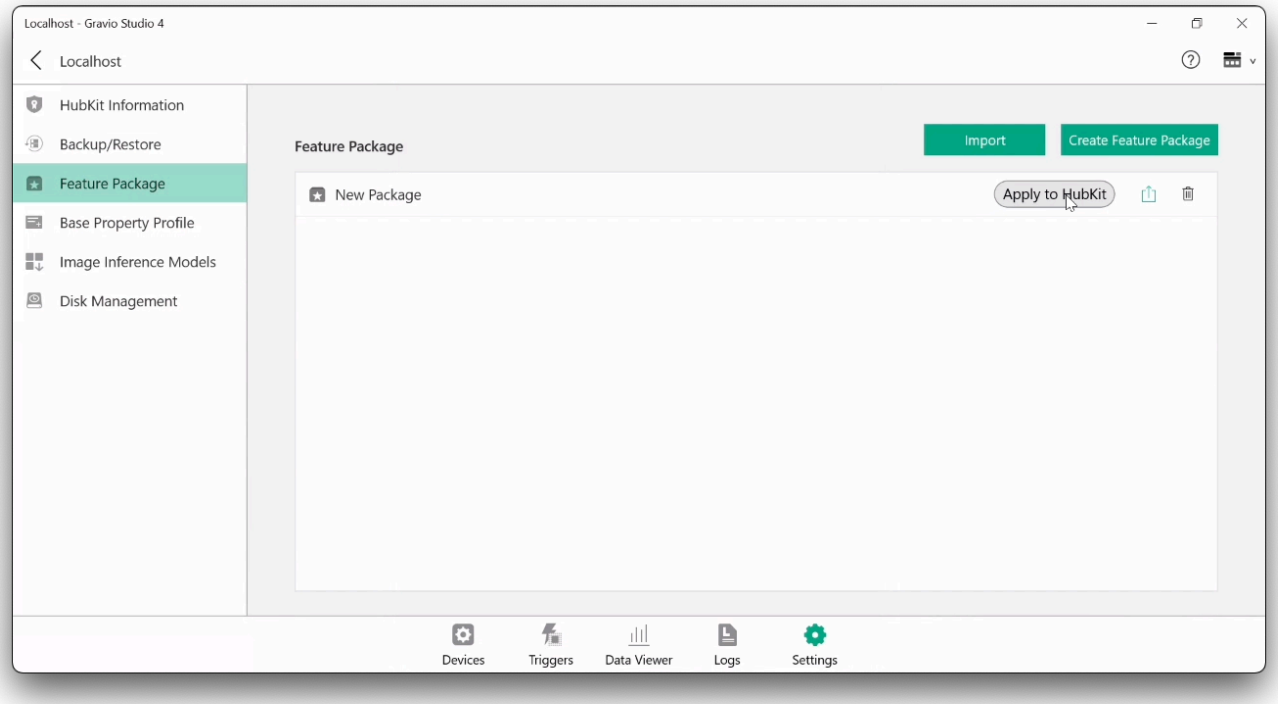
Base Property Profile

These are re-usable settings that will not be included in actions or backups. Examples for such settings are authentication tokens, passwords, login details, secret keys, mail server settings etc. This allows you to re-use them in your actions where needed.



Feature Package

A feature package is a set of configurations including areas, layers, triggers, actions that you can export as a Zip file to be imported on either your cloud account, or, if you have the Enterprise edition of Gravio, to your coordinator for distribution to your edge nodes.



Attention: that we recommend that you put all confidential data such as tokens, usernames, passwords for third party services into a base property profile (see above) so those details are not exported included in the package, if you tick the appropriate checkbox during the export process.

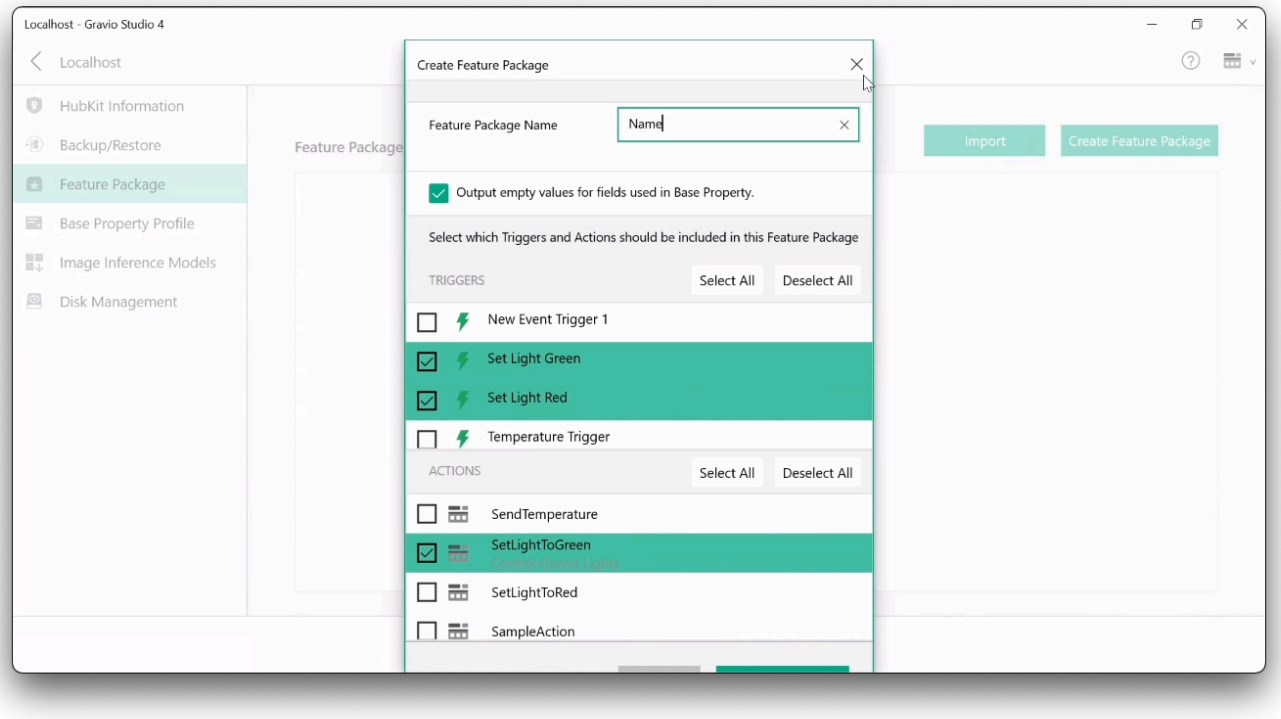


Image Inference Models

In this section, you can deploy the available image inference models to your local HubKit for use. There are two ways of deploying computer vision models:

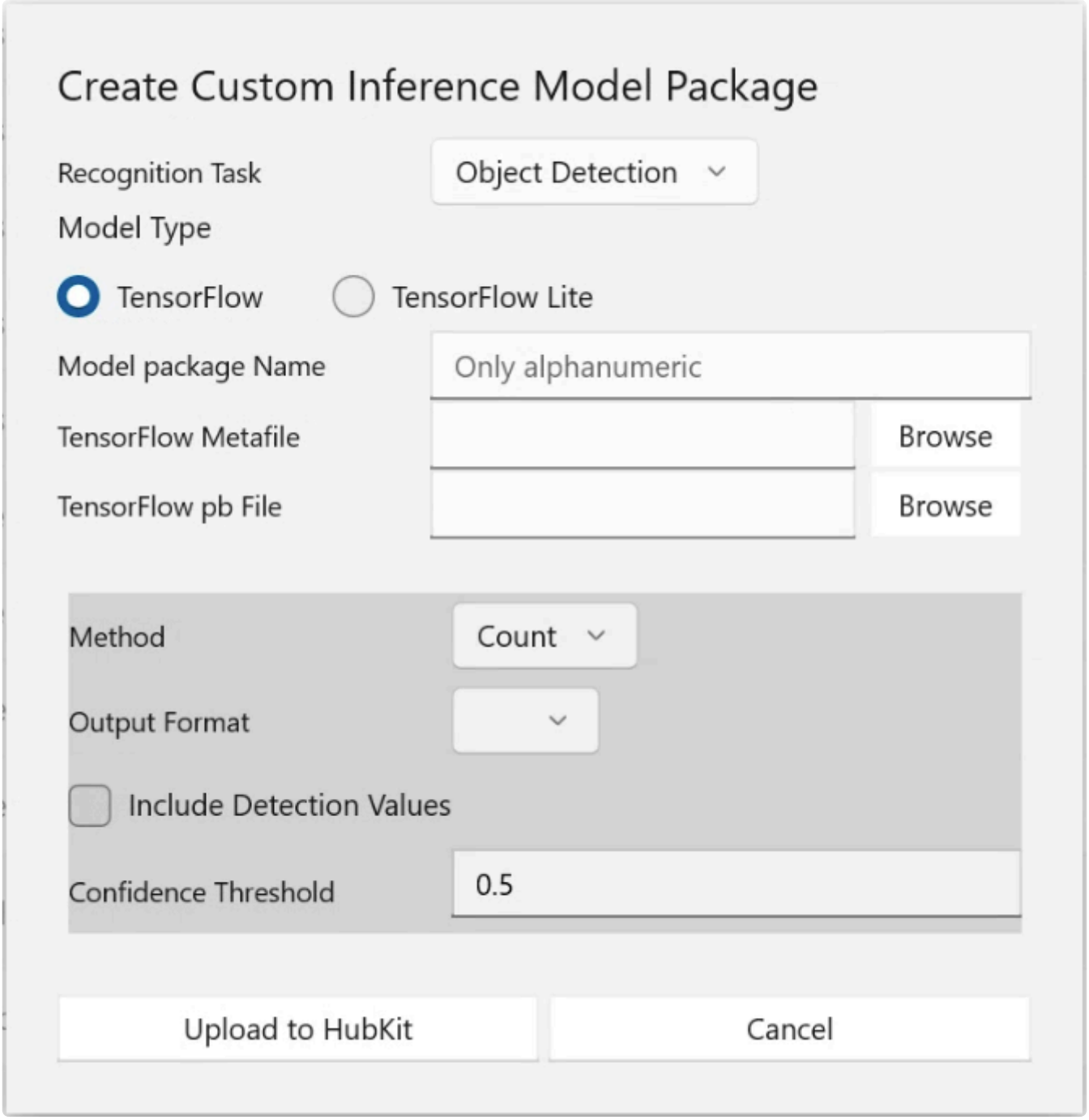
1. Locally, via Gravio Studio (which is available for all versions including Free, Basic and Standard)
2. Via the Gravio Coordinator backend (which is only available for Enterprise licenses)

There are also a few standard models that come from our Gravio Cloud. They include a pre-trained people counting model.

To upload a new computer vision model locally via Gravio Studio, press the “Create” button, select your `.tflite` file package containing and upload it.

Alternatively, click the “Import” button and import a previously exported `zip` file.

Once you click the “Create” button, you will see the following dialogue box:



Select the Recognition Task you would like to conduct. You can either focus on detecting objects or to classify an image. For Image Classification you will need to use TensorFlow Lite.

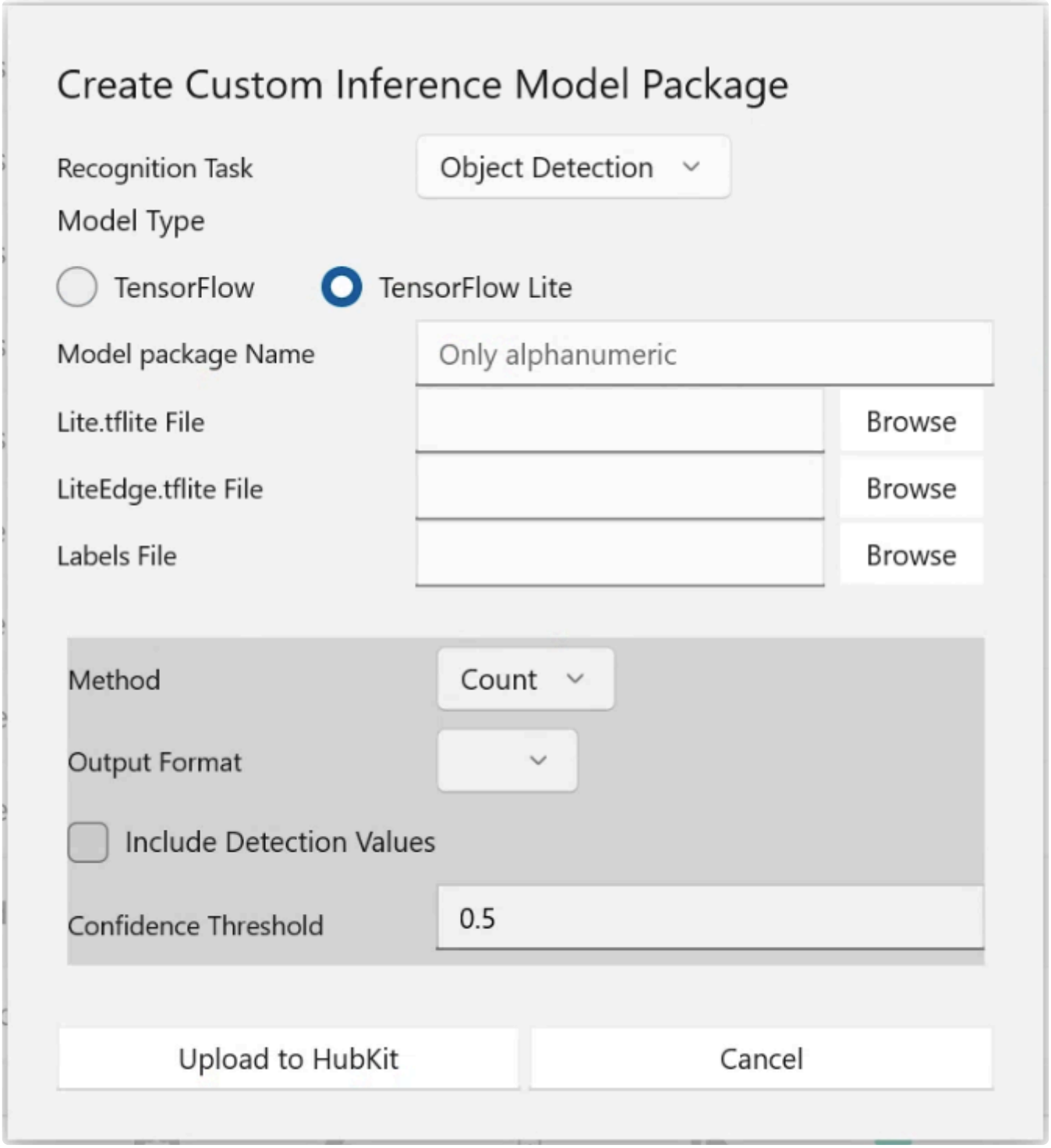
For TensorFlow prepare the following files:

File
TensorFlow meta file
TensorFlow pb file

In the dialogue box:

Form Field	Explanation
Model Package Name	Use alphanumeric characters to describe your package. This name will appear as “Sensor”.
TensorFlow metafile	Select the prepared file
TensorFlow pb file	Select prepared file
Method	Select **Count** to output a number or **Group By** to return a string
Output format	Select **JSON** to output detailed information, or select **Value** to output only the value.
Include detected values	Select this checkbox to include detected values
confidence threshold	Set between 0.0 and 1.0 to define at which threshold level the “sensor” should be fired

*If you pick the TensorFlow Lite method, you will see the following screen instead. *



In this case prepare the following files:

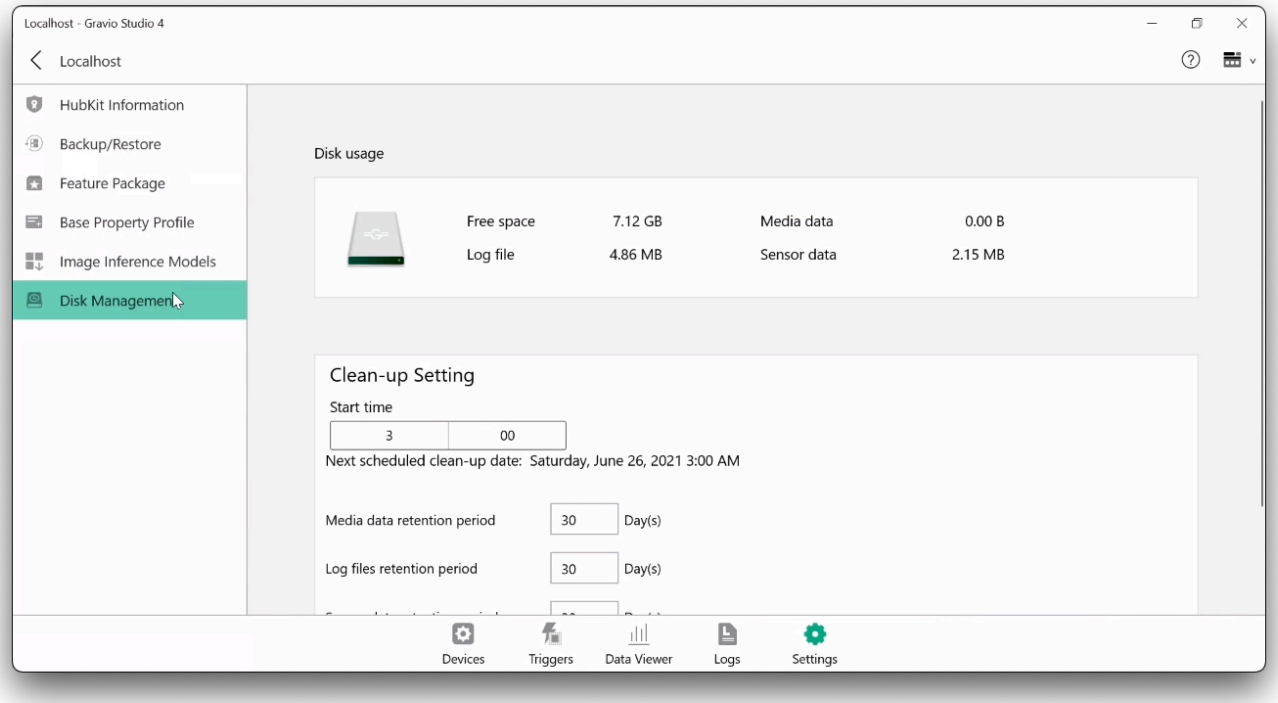
File
`Lite.tflite` file
`LiteEdge.tflite` file
`labels` file

In the dialogue box:

Form Field	Explanation
Model Package Name	Use alphanumeric characters to describe your package. This name will appear as “Sensor”.
Lite.tflite file	Select the prepared file
LiteEdge.tflite file	Select prepared file
Labels File	The file containing the labels to be detected
Method	Select **Count** to output a number or **Group By** to return a string (Note: this cannot be specified if “Image Classification” is selected.
Output format	Select **JSON** to output detailed information, or select **Value** (or **Name** of you selected “Image Classification) to output only the value.
Include detected values	Select this checkbox to include detected values
confidence threshold	Set between 0.0 and 1.0 to define at which threshold level the “sensor” should be fired

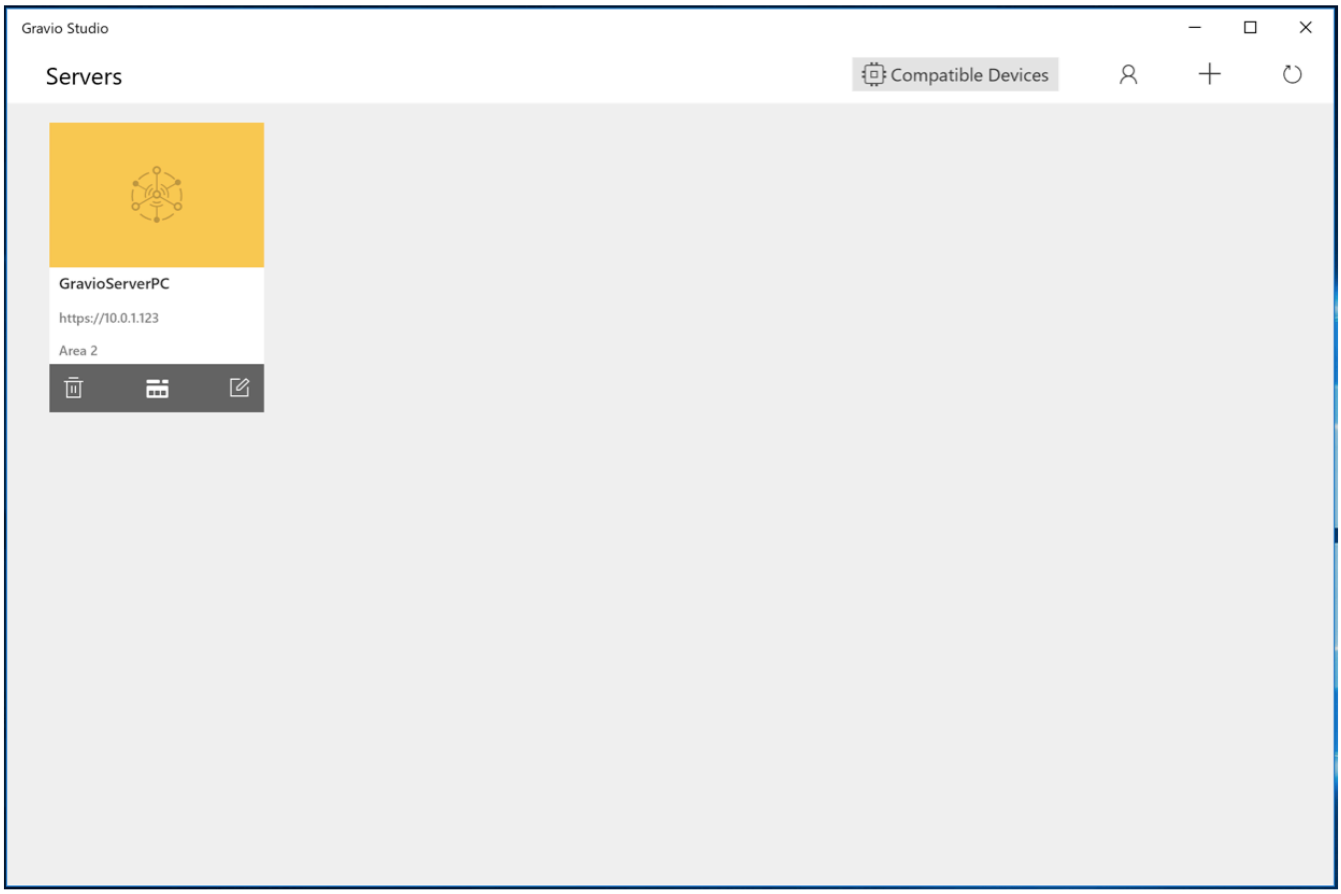
Disk Management

In Disk Management you can view the health status of your disk and set the various parameters to keep the disk space used optimally.



6.2. Gravio HubKits Overview

The Gravio's HubKit Lists view in Gravio gives you an overview of all connected Edge HubKits available. These can be software servers from Linux, Windows or Macs where you have installed the Gravio HubKit, or they can be Gravio Hub devices. For information on how to set up a server, please see Installation and Setup.



Each Server has three views:

1. The **Server Configuration** view, where you configure and connect the sensors
2. The **Actions Configuration** view, where you configure the actions and steps that are triggered by the sensors
3. The **Server Settings** view, where you configure the server, the IP address or the image displayed

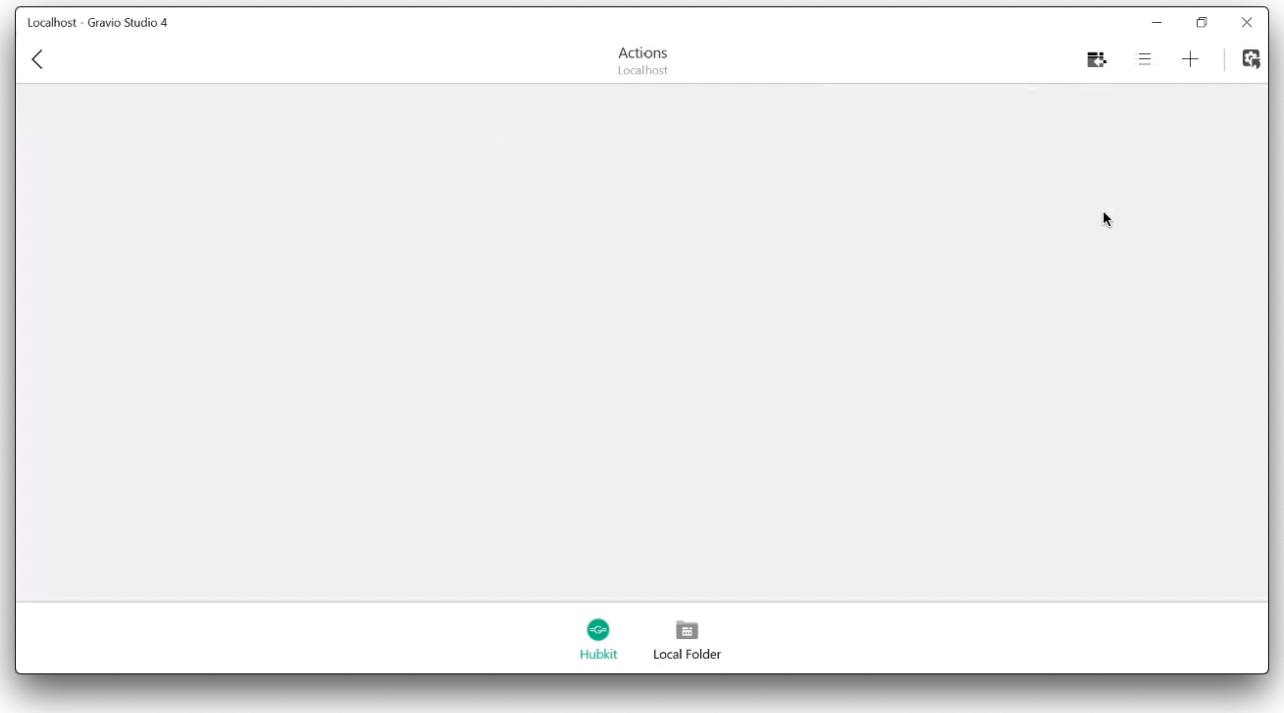
- To enter the Server Configuration View, double click on the top half, i.e. the main image of the card.
- To enter the Actions Configuration View, double click on the middle actions icon at the bottom edge of the card.
- To enter the Server Settings View, double click on the right side pencil icon at the bottom edge of the card.

6.3. Action Configuration View

The Gravio HubKits can trigger Actions. You can think of an Action like a small and simple computer program, that's made up of Steps. Actions can be created in the Action Editor which can be opened by clicking on the Action Editor icon:

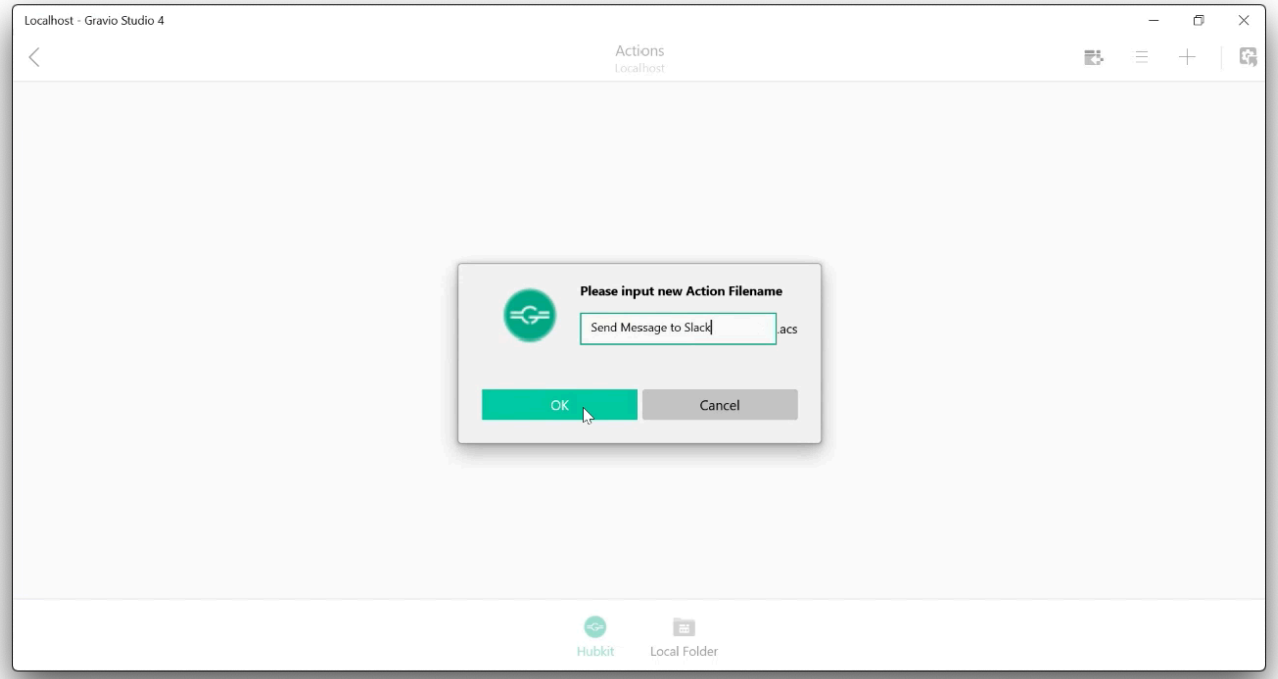
You can find these icons in various places, including the HubKit overview, the Devices view and on macOS also in the trigger view for example.

If you click on these, you will see the overview of available actions of this HubKit:

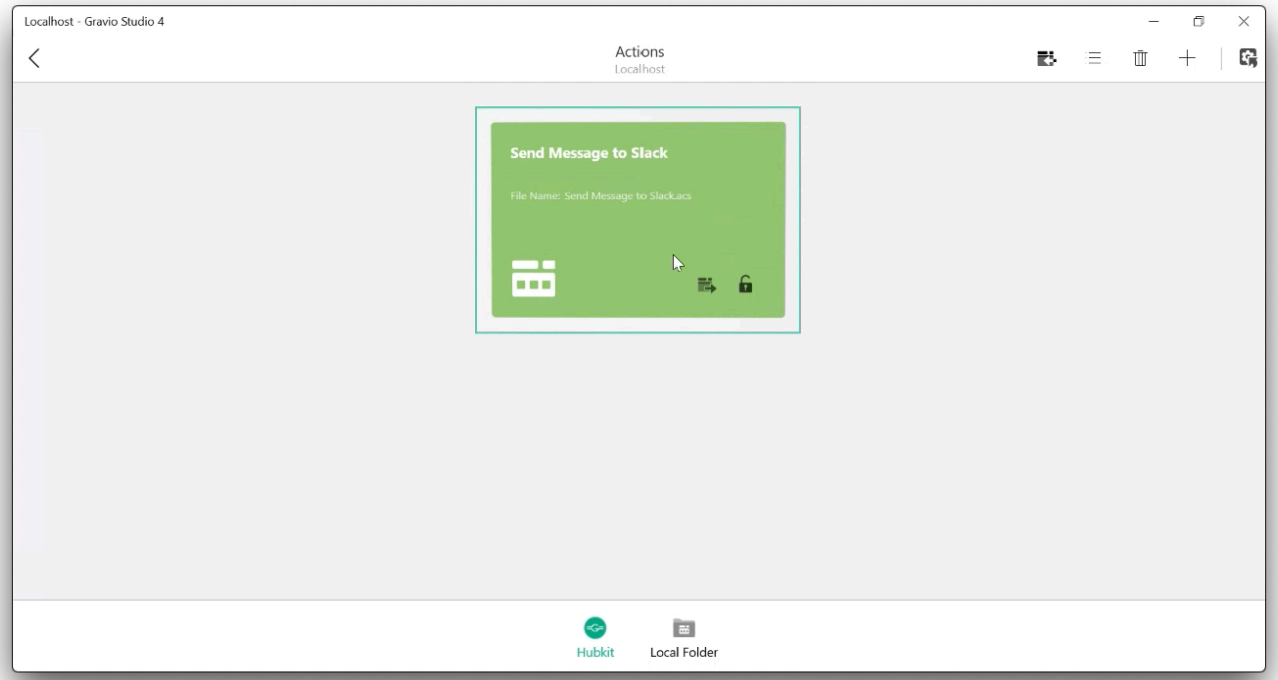


At the bottom, you can choose between HubKit, which are the actions installed on that HubKit, or the Local Folder, which are actions that you can edit on your local computer to be deployed or used in other HubKits. Note that local actions will run on the HubKit you are currently connected to, without them being deployed on the HubKit.

Click the plus sign at the top right to create a new action. A name field will pop up where you can give it a name. Note, action files can be shared and they have the `.acs` suffix. We recommend giving the action the name of the expected outcome (rather than what triggers the action), so you know always know what it is supposed to be doing.



Once created, the actions will appear in the Action Overview and now they can be triggered from the Triggers:



Double-Click the action to edit it.

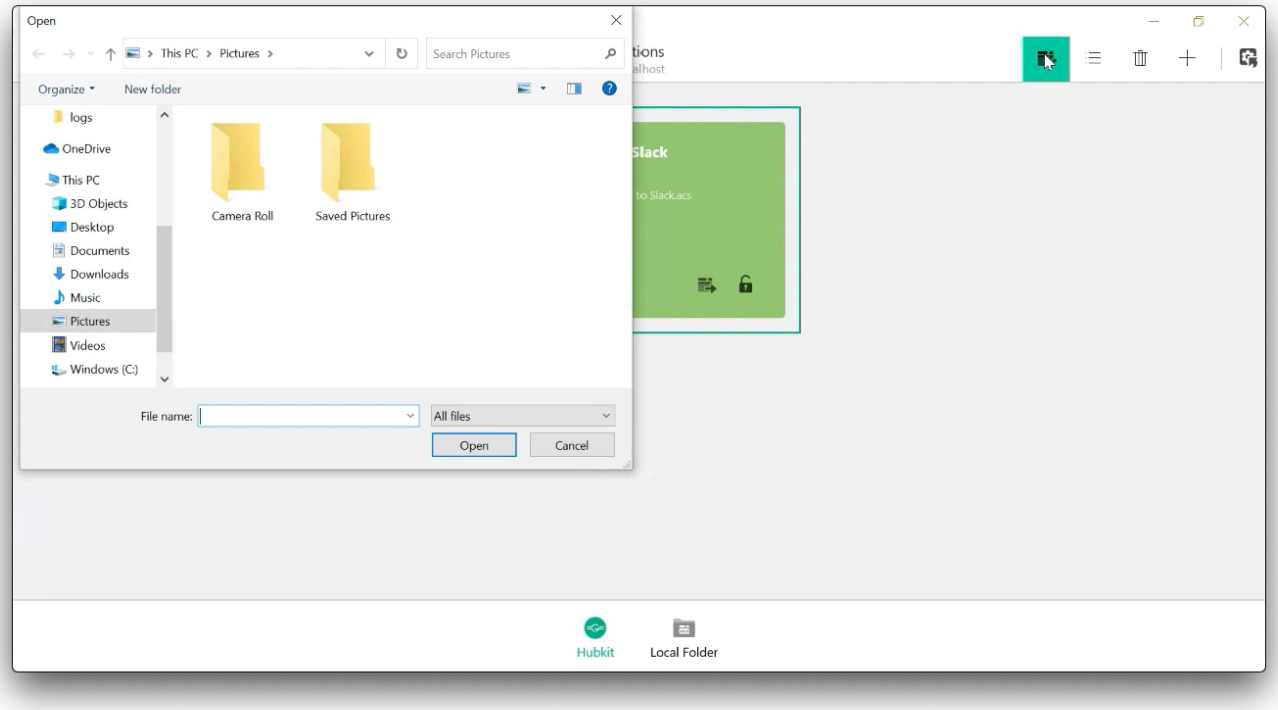
Each Action also has 2 buttons on it:

1. One button is to export the action to a file, ready to be uploaded to Github or shared with

somebody

2. The other button is to lock the action so it can't be edited accidentally.

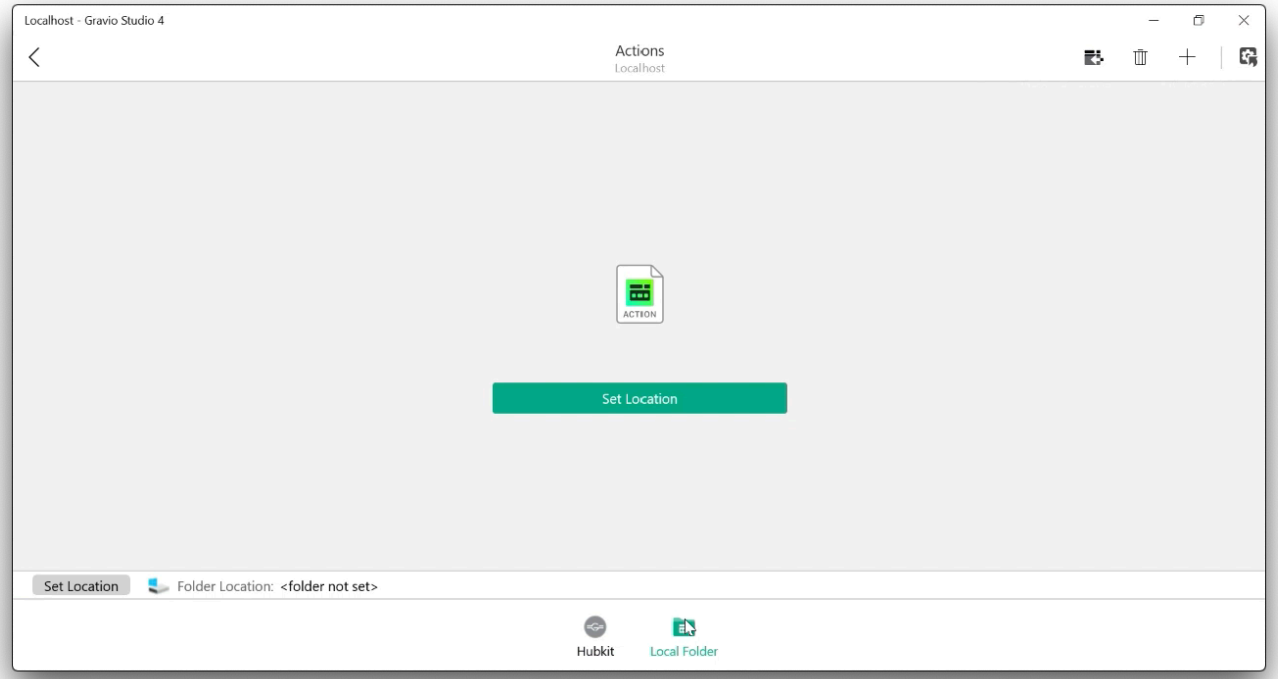
The button at the top allows you to import action files with the `.acs` ending:



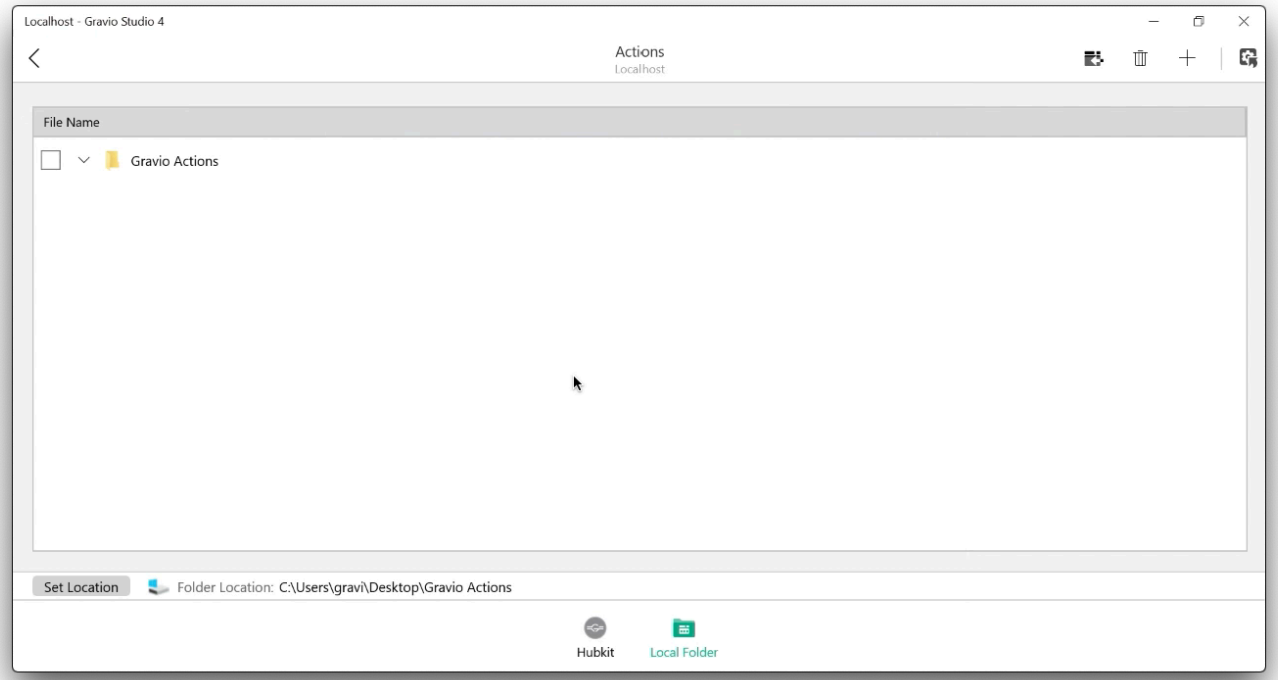
This will open the local browser file where you can select your `.acs` file.

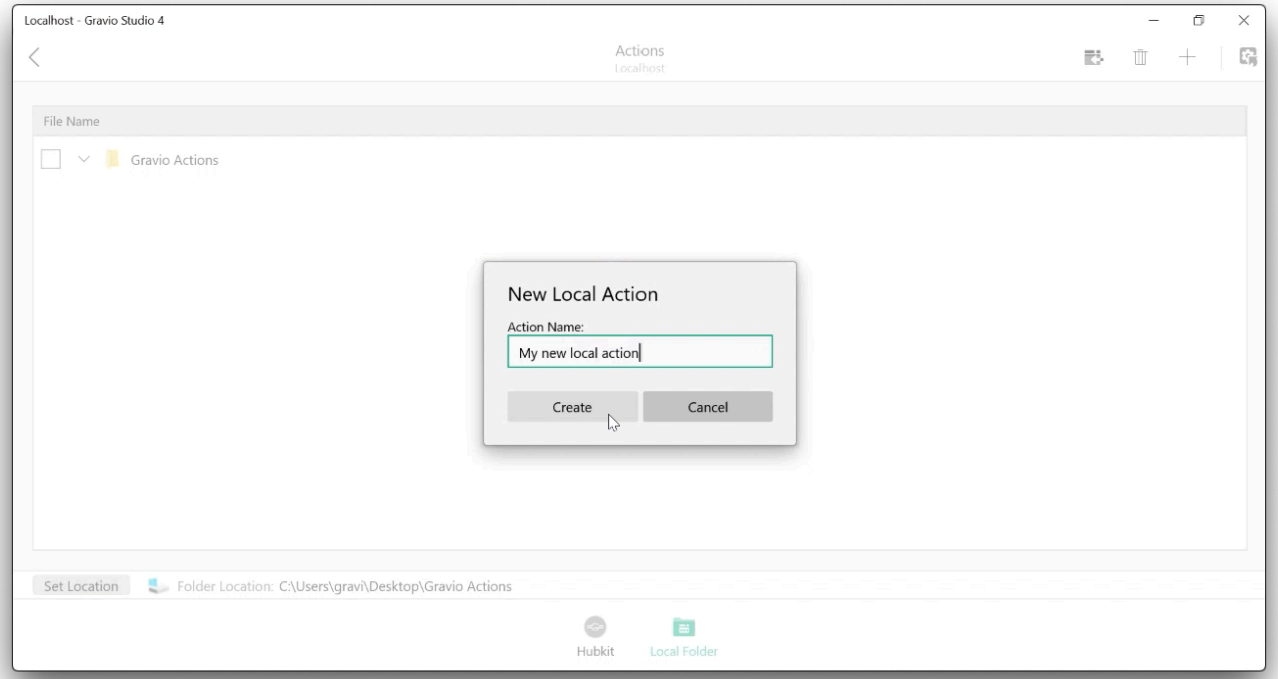
Local Folder

The Local Folder tab allows you to set a local folder where you can create your own local actions without having to deploy them to the Gravio HubKit. The first time you open the local folder, you need to set the location of the folder. Choose a folder where you will be keeping your Gravio Actions.

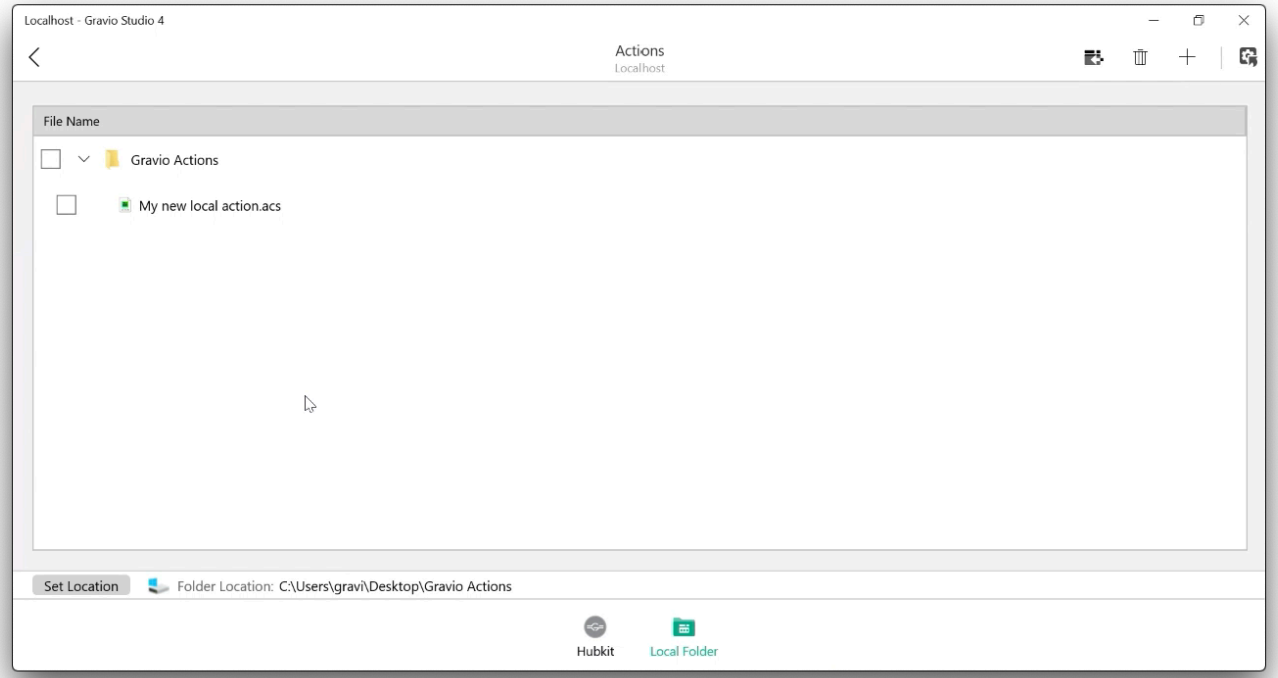


Once you've chosen your folder, you will see this view, and clicking the + symbol on the top right will create a local action:





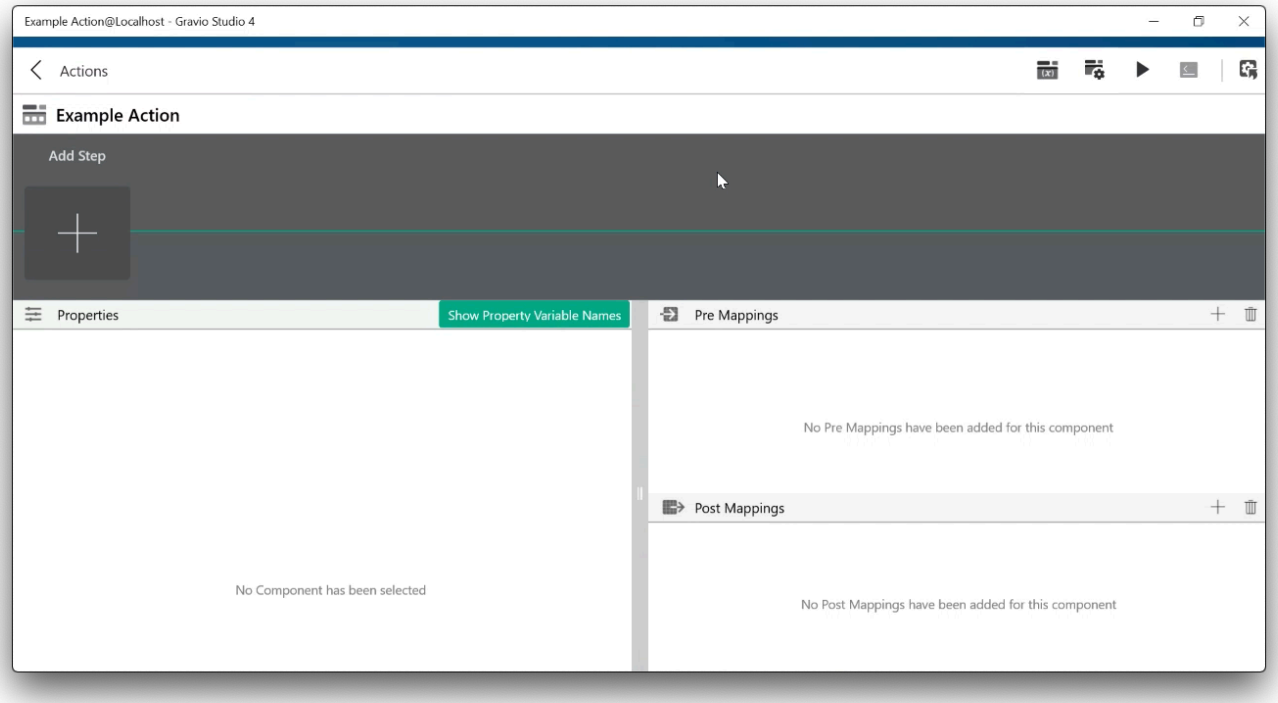
Double-click on the action file in order to edit the action locally.



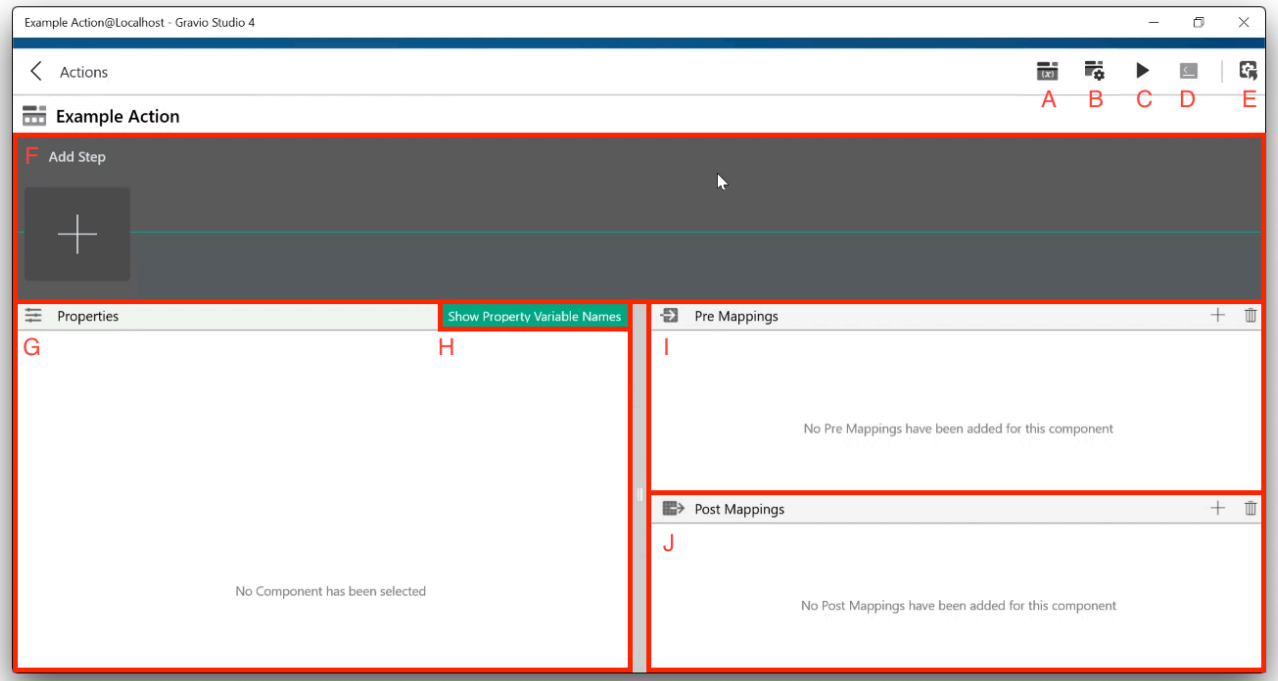
6.3.1. Writing an Action

Double-clicking on your Action in the Actions Overview will open your Action Editor. This is where you put your actions together. Actions consist of separate steps which you can string together, create “decision trees” and create inputs and outputs.

The general view of an Action Step looks like this:



We can split that view into separate parts which you will recognise in the step editor:

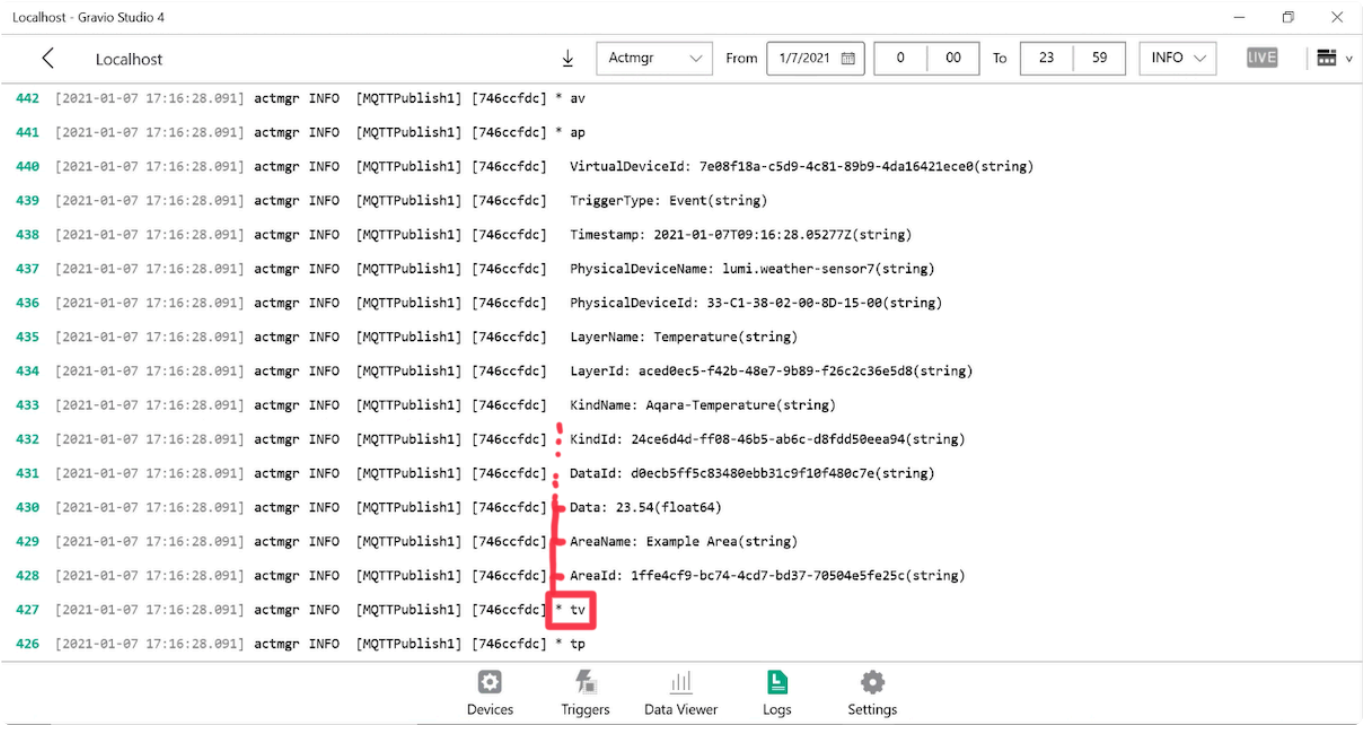


A	This button will open the available Action Variables, which are prefixed with `.ap`. More details about various variable and property types can be found under Available Variables in Actions
B	This button will open the property inspector. Properties are prefixed with `.cp` and specific to properties. More details about various variable and property types can be found under Available Variables in Actions
C	This is the button to run the Action manually. Note, not all actions may run without any sensor input.
D	This button opens or closes the console, where you can see debug information while running the Action.
E	This button takes you back to the device editor where you can set and configure devices and sensors.
F	This is where you can add the Steps (consisting of components) to your Action. You can also drag and drop the components back and forth inside this row. Steps will be processed from left to right.
G	The properties part shows the settings of the properties in the currently selected step component. These vary significantly between different components.
H	With this button you can toggle the various component property variable names. More details about various variable and property types can be found under Available Variables in Actions
I	Pre-mappings are happening before the component runs. They represent and process the data input.
J	Post-mappings are happening after the component has run. They represent and process the data output. Please ensure that the input of the next step is always in line with the outputs of the previous step.

6.3.2. Available Variables in Actions

Think of the Action Steps as connectable parts that have an input and an output pipe. Each Step waits for incoming objects (in `cv.Payload`) it can then process and pass on through the output pipe to the next Step. Before processing the Step, you can Pre-Map and after processing the Step you can Post-Map the `cv.Payload` variable.

To inspect the action variables coming through, you can enable debugging (see below) and inspect the `Actmgr` logfile. In the example here, you can see the `tv.` variables outputted by a temperature sensor:



You can therefore access the values of these variables using `tv.AreaId`, `tv.AreaName`, `tv.Data`, `tv.KindId` etc. read from down to up chronologically.

There are 6 broad categories of Variables in Gravio:

Action Property

Referred to by adding a prefix of `ap.`
Action Properties can be defined as a property passed from the outside the Action when the Action is executed.

Action Variable

Referred to by prefixing `av.`
Action Variables are variables that can be referenced within an action. This could be for example Sensor Data.

Component Property

Referred to by prefixing `cp.`
A Component Property is a value that determines the behaviour of a component. They differ vastly between different types of components.
For example, the POST component might have an URL as property, whereas the Gravio Light component has some color and illumination settings of the Gravio Light.

Component Variable

Referred to by prefixing `cv.`
Some variables are implicitly declared in the component.
For example, the `cv.StatusCode` contains the HTTP status code and response headers of the HTTPRequest Component.

To find out what the respective Component's `cv.` values are, please refer to the documentation of the respective Component.

Trigger Property

Trigger properties set by event triggers and timer triggers can be referenced in actions.
Referenced by prefixing it with `tp.`
For example `tp.KeyAreaName`

Trigger Variable

Referenced by adding a prefix of `tv.`
The data sent by the sensor can be referenced in the Action program with the following trigger variables

Trigger Variable	Description
tv.AreaId	Area ID
tv.LayerId	Layer ID
tv.KindId	Data Kind ID
tv.PhysicalDeviceId	Device ID of the sensor
tv.Timestamp	The time when the data was output from the sensor. The format is RFC3339Nano.
tv.DataId	This sensor data unique ID
tv.Data	The value of the sensor data. `tv.Data` will be the same as the first component's `cv.Payload`

Value (type)

Variable Type	Description
Int	Integer
Float	floating point number
String	String , Strings can be concatenated with `+`, for example `"The value from the sensor is " + tv.Data`
Bool	true/false
DateTime	Date Time
Byte sequence	Byte sequence
Array	Array
Object	Object

Type	Example
literal	`[1, 2, 3]`
Array	`["abc", "def", "hij"] ["abc", 123, cv.variable]`
Object	`{"key1":123, "key2":"abc", "key3":cv.variable, "key4":[1, 2, 3], "key5":{"subkey1":"def"}}`

Type Conversion

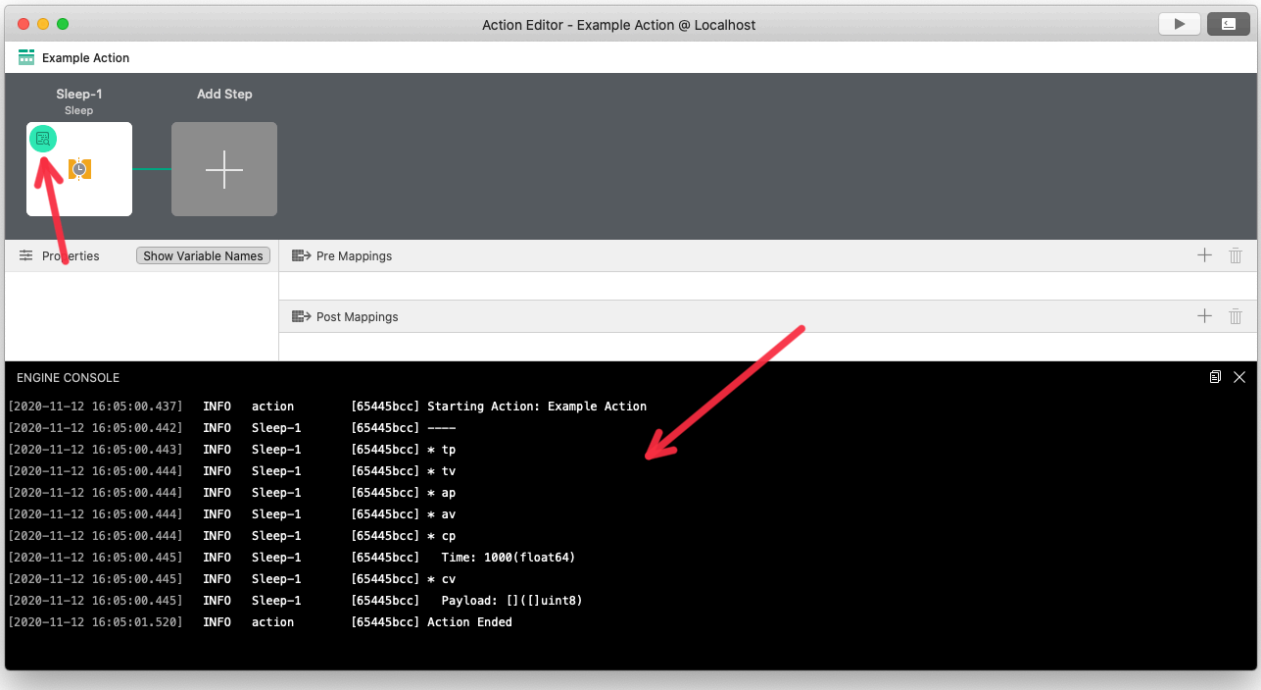
It is possible to convert from each type to each type.

Type	Example
Boolean	• The numeric system is false when it is 0 and true otherwise. • If the date and time is 0 in Unix time, it is false, otherwise it is true. • True if the string is “true”/“1”, false otherwise. • Byte strings follow string rules after being converted to strings • false for any other type.
Int	• Boolean is 1 when it is true, 0 otherwise. • Floating decimals are a Go language casting rule • At the time of the date and time, the Unix time is a number. • For strings, follow the rules of strconv.ParseInt • Byte strings follow the string rules after being converted to strings. The other types are 0
Float	• Boolean is 1.0 if it’s true, 0.0 otherwise

	• Integer is a Go language casting rule • At the time of the date and time, the Unix time is a number. • In the case of a string, the rules of strconv.ParseFloat are followed • Byte strings follow string rules after being converted to strings • 0.0 for all other types.
String	• Boolean is “true” if it is true, otherwise it is “false”. • Convert integers to strings using strconv.FormatInt and floats to strings using strconv.FormatFloat • If it is a date and time, fix it as a string in time.RFC3339Nano format. • Byte sequence is regarded as a string of utf-8 bytes. If there is an error, an empty string • Array and Object are made into a string by multiplying by json.Marshall
DateTime	• Numeric systems and Boolean are converted to integers by a rule that converts them to integers, and then converted to date and time by a rule that converts them to Unix time • For the string, if the following formats are tried in order, they are considered to be date and time, otherwise they are 0 (0 for date and time is 1 January 1, 00:00:00). time.RFC3339Nano, “2006-01-02T15:04:05.999 MST”, “2006-01-02T15:04:05.999”, “20060102T150405.999 MST”, “20060102T150405.999”, • The same rules as for strings • Otherwise, zero.
byte sequence	• Boolean, numeric system and date/time are changed to strings and made into a utf-8 byte string • The string should be a utf-8 byte string • Array, Object can write json.Marshal and make the string a byte string • Array, Object numeric system and the date and time should be an array containing one of its values • Unmarshal the string as JSON string. In case of an error, an array of size 0 • Byte strings are rewritten as strings, same rules as strings The array and objects are still in the • Any other type is an array of size 0

Debugging

You can easily dump the value of all variables by activating the “debug” button on the top left corner of the step. If you enable it, Gravio will dump all variables to the console at the bottom:



The variable values are also dumped in the `actmgr` logfile, which you can inspect in the logfile area of Gravio.

The variable handling in Gravio follows the schema according to github.com/PaesslerAG/gval

6.3.3. Referring to the value of a Object variable in an Action

If you like to refer to the value of a variable of type Object available in an action, you can access the element with a selector or square brackets connected to the variable with a dot.

As an example of an Object type variable, if you use IVAR's face recognition feature and execute an action in the Trigger Manager, you will receive the following JSON data in your action. (Partial Data)

This data can also be referenced by any component as `tv.Data` (TriggerVariable.Data) when the action is executed. It can also be referenced as `cv.Payload` in the first component, as it is also the Payload of the first component.


```
{
 "id":"XXXXXX",
   "source":{
      "channel":4,
      "bacId" : "YYYYYY",
      "taskId" : "ZZZZZZ",
      "ivarId" : "000000"
   },
   "common":{
      "description":"Face Recognition",
      "presentImageTime":"2019-06-17T07:53:22.986Z",
      "time":"2019-06-17T07:53:22.986Z",
      "type":"FR"
   },
   "iod":{
      "age":"MIDDLE",
      "ageRange":[
         38,
         53
      ],
      "faceAvailable":true,
      "frSuitable":"NA",
      "gender":"MALE",
      "glasses":"REGULAR",
      "groupId":0,
      "liveness":"NO_RESULT",
      "livenessProbability":"0.0000",
      "mask":"NA",
      "objectId":649,
      "objectType":"FACE",
      "occlusion":"NA",
      "race":"NA"
```

```
   },
   "fr":{
      "candidates":[
         {
            "customId":"00001",
            "displayName":"Person B",
            "id":"6",
            "indentityGroup":"Company A",
            "similiarityScore":"0.7272"
         }
      ],
      "fdrVersion":"60032084",
      "indentityGroup":"Set2",
      "mmsReturn":"OK"
   }
}
```

For example if you are referring to: fr -> candidates -> displayName

```
tv.Data.fr.candidates[0].displayName
```

It is written as

IVAR JSON is a simple key name, so you can refer to it using the dot-connected selector, but if a space or hyphen is used in the key name, you can also refer to it as follows (if the display name contains a space). fr and candidates can be mixed and matched by connecting them with dots.

```
tv.Data["fr"]["candidates"][0]["display Name"]
```

```
tv.Data.fr.candidates[0]["display Name"]
```

The above is only for referencing and retrieving values, but it can be used in the same way to insert conditional expressions in Filter components.

```
cv.Payload.fr.candidates[0].displayName == "Person B"
```

These features can be used in addition to actions, where you can write trigger conditions and expressions.
You can also enter an expression in the IVAR condition of a trigger, and if you want to trigger an action only for men, you would write the following expression

```
tv.Data.iod.gender == "MALE"
```

6.3.4. Action Component Behavior

The action component has two component variables (`cv.`): the input payload (`cv.Payload`) and the output payload (`cv.Payload`).
You can use it as the input payload (`cv.Payload`) when used in Pre-Mapping and as the output payload (`cv.Payload`) when used in Post-Mapping.

Input Payload

The “Input Payload” is the same as the previous component’s “output payload” (`cv.Payload`).
The first component placed in the action has a (color-blue) trigger variable (`tv.Data`) passed from the trigger that is the input payload (cv.Payload).

Output Payload

The “Output Payload” is the input payload of the next component.
The output payload is defined in each component as it is created by the operation of each component.
For a description of what the “output payload” looks like, see the individual action component descriptions, see Action Components .

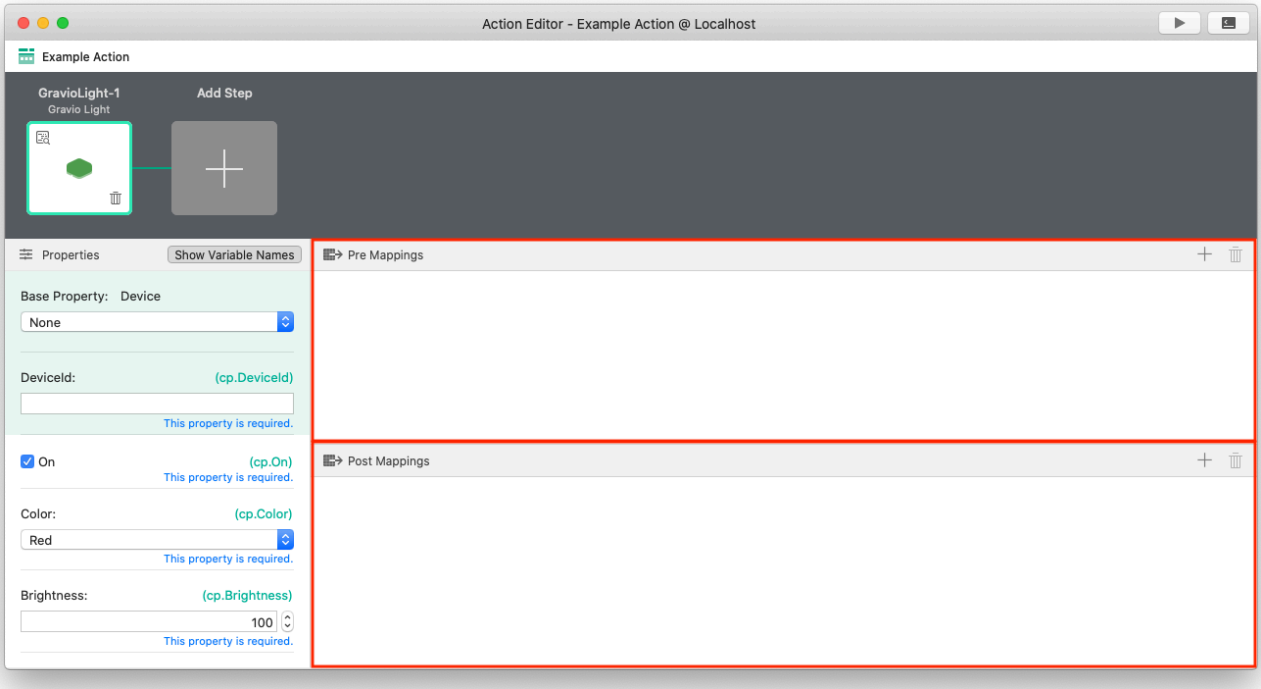
Payload

The Payload of the actions are always a JSON document format and can contain one or more JSON arrays or JSON objects. The next component will be executed for each JSON object individually in `cv.Payload`.

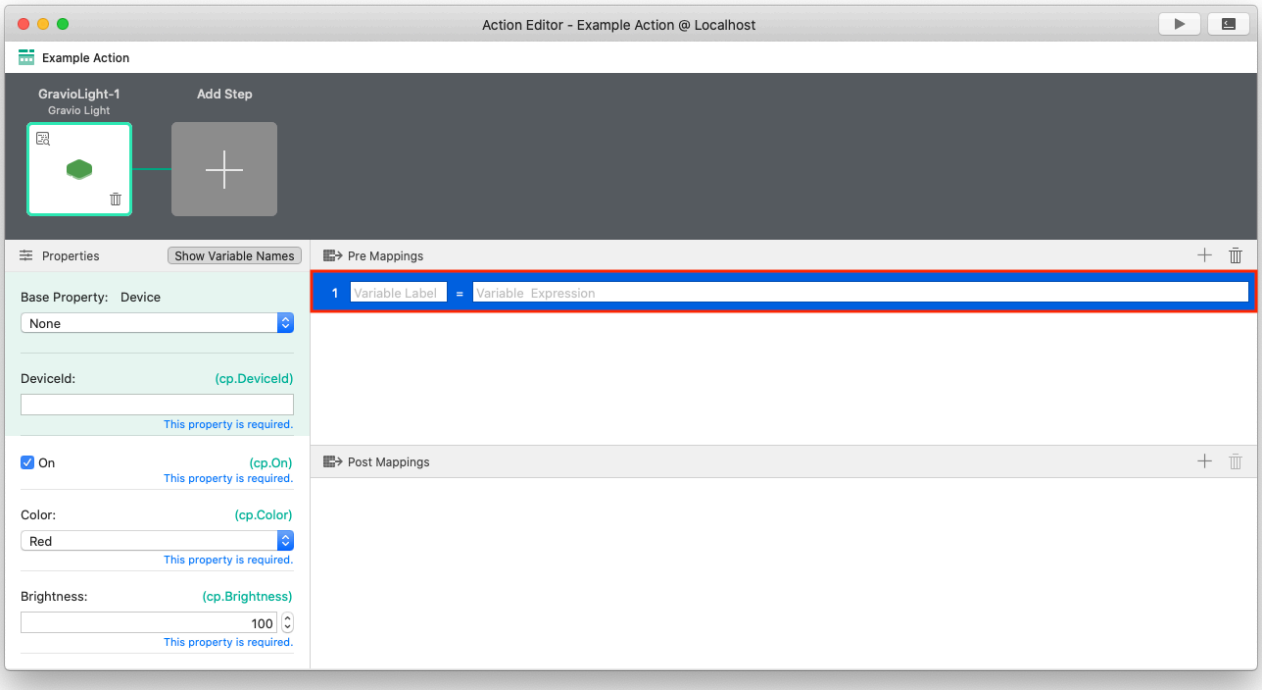
Some components like Filter and Join typically have more `cv.Payload` inputs than outputs. And others such as ReadCSV are likely to have more outputs than inputs (if any).

6.3.5. How to write expressions in Pre Mappings / Post Mappings

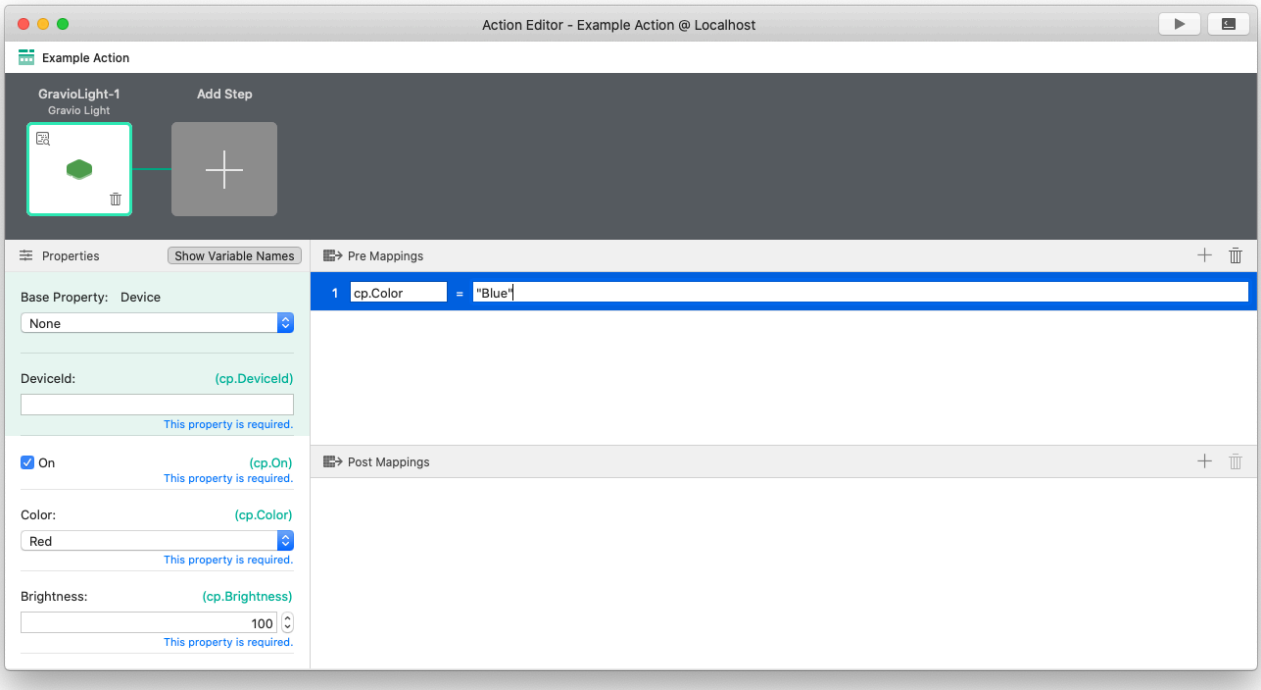
Each action component in the Action Editor has a pre Pre Mappings / Post Mappings area on the right side of the screen:



Press the “+” button to add a new line to either the Pre-Mappings or the Post-Mappings. Pre Mappings are executed before the main function of the component is conducted. Accordingly, Post Mappings are executed thereafter. Clicking on the + icon will add a new line:



You can write the name of the variable on the left side and the expression or value to be assigned to the variable on the right side. In this example, we are in the Gravio Light component and assign a color to the light:



Note the `cp.` prefix which refers to the Component Parameter variable. By clicking on the “Show Variable Names” button you can see how the variables in the component are being referred to.

Note that the sequence of execution is from top to bottom

Expression Description

The expressions on the right-hand side can be written as quadratic operations (`+` / `-` / `*` / `/`) or functions. You can also describe variables with prefixes such as `tp.` / `tv.` / `ap.` / `av.` / `cp.` / `cv.` as variables.

When you assign the evaluation result of a right-hand expression to the left-hand side, the type of the right-hand value (`Int` , `Double` , `String` , `Bool` , `DateTime`) is converted to the type of the left-hand side.
If you write an expression that is not an assignment (e.g., `cv.Payload > 10`), it will return a value of type Bool.
To return a Bool value, you can perform a comparison operation on a number or string.

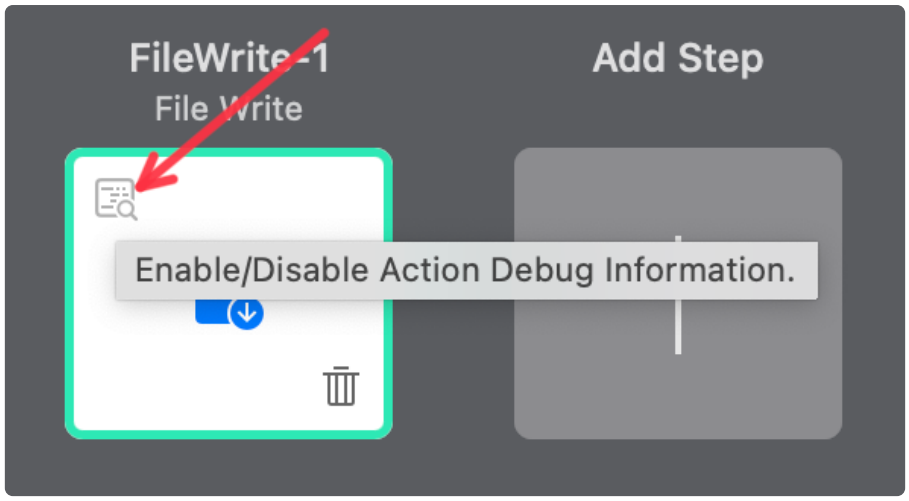
Example formula
`cv.Payload > 10`
`cv.Payload == "abc"`
`true`

Ternary Operators

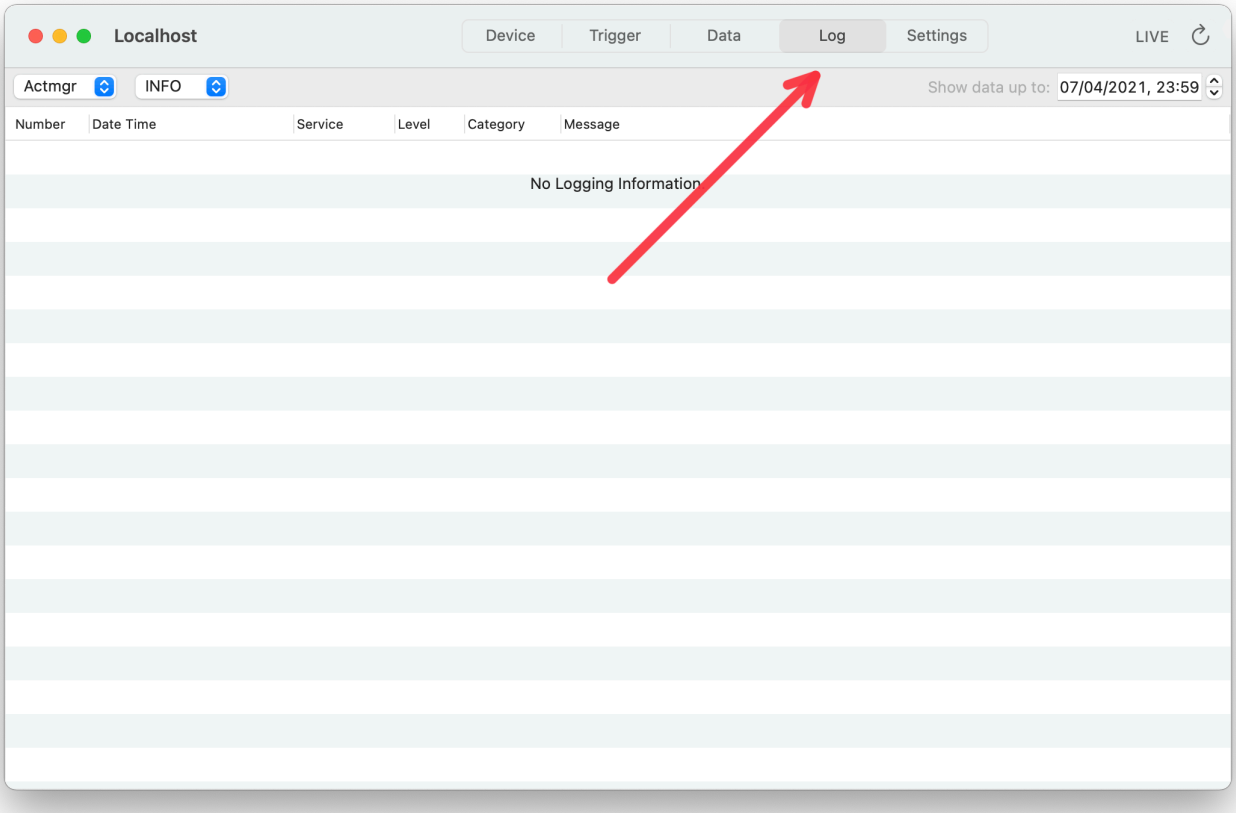
You can also write conditional or ternary operators to set values, such as `tv.Data == 0 ? 'up' : 'down'`

Debugging

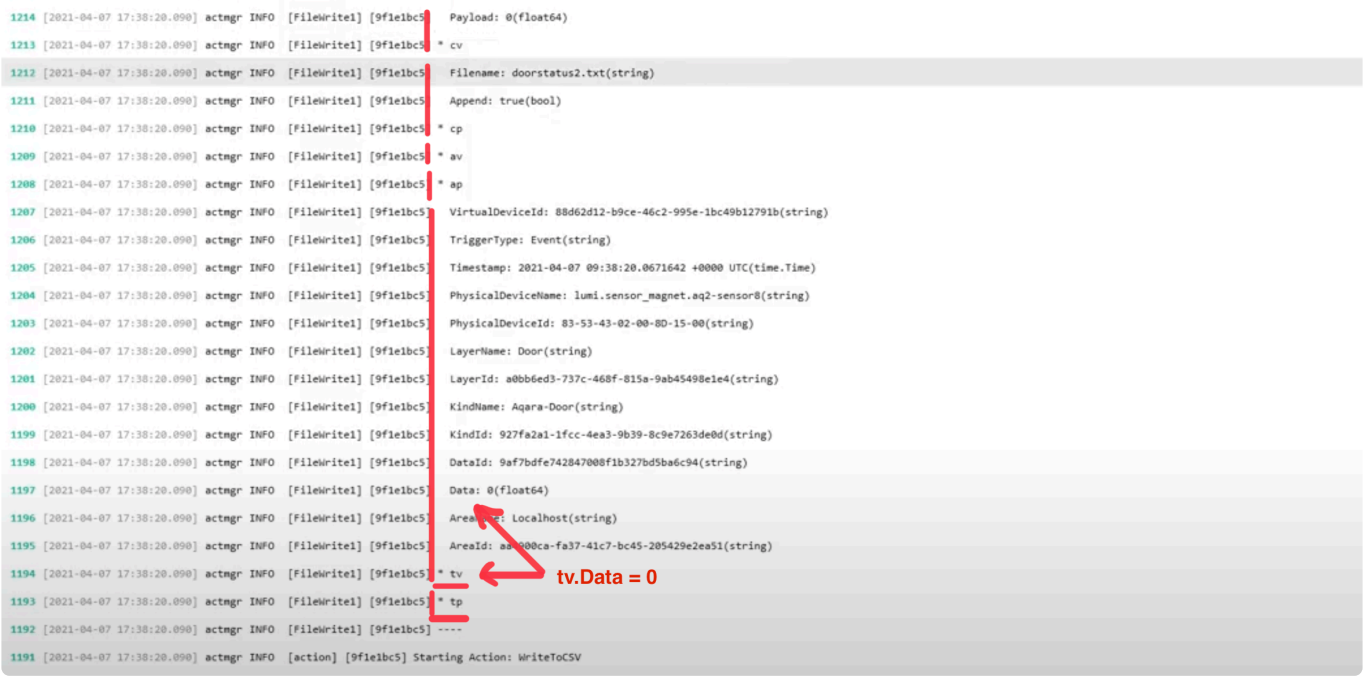
You may want to find out, what certain variable's values are, especially if they are coming from sensors. You can do that by enabling the icon "debug" toggle icon in your step:



This will show you the variables and the respective values in the Action Manager Log:



You will see the values of the available variables there, reading upside down (as new logfile content is appended at the top):

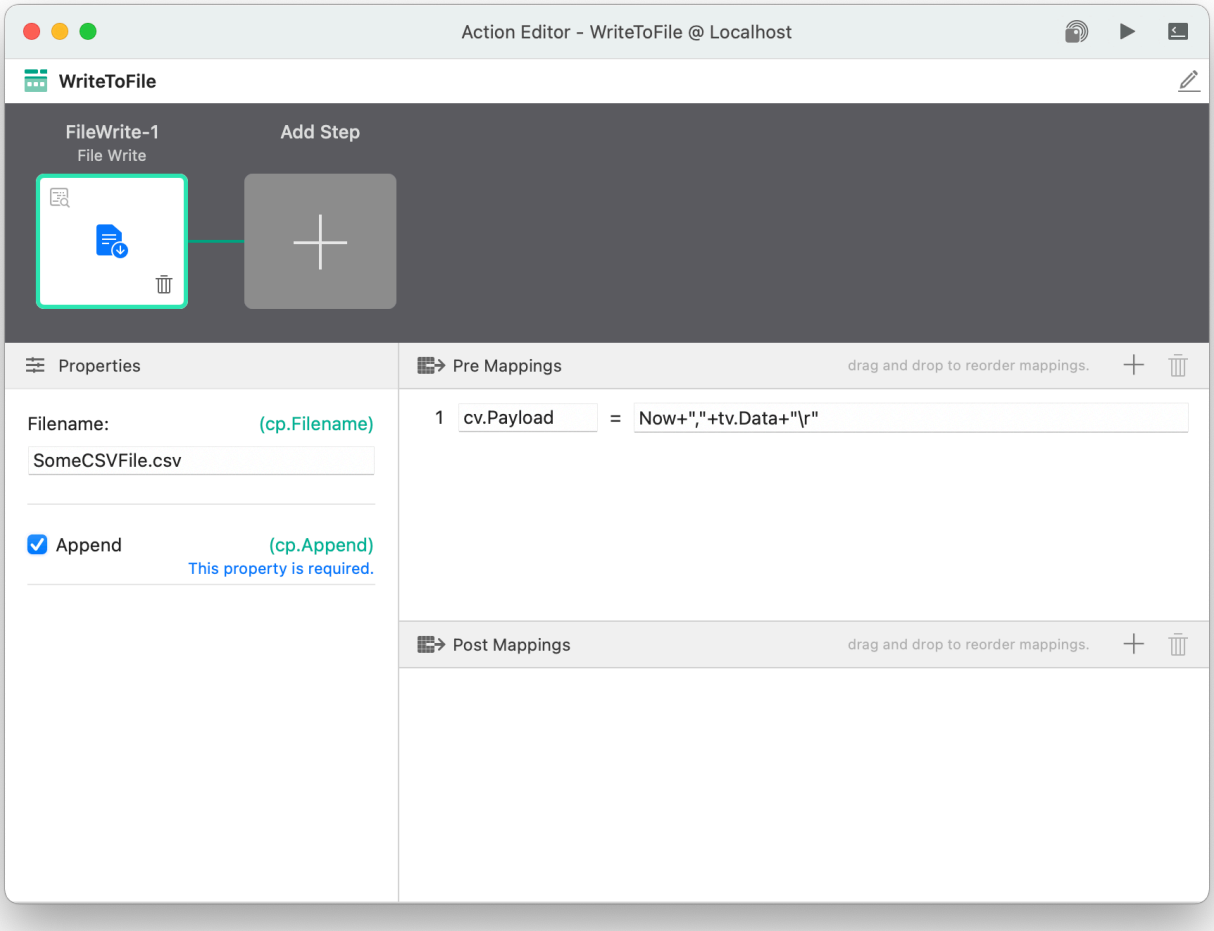


This shows you, the variables prefixed with tv (trigger variable) are:

`tv.AreaId` is `98ca-fa37-41c7-bc45-285429e2ea51(string)`
`tv.AreaName` is `Localhost(string)`

`tv.Data` is `0 (float64)`
... and so on.

Therefore you can access this data in the Gravio action for example like in this File Write step here:



6.3.6. Functions available in the Pre Mappings / Post Mappings expressions

You can use the following functions on the right side of the expressions in Pre Mappings / Post Mappings.
See Functions for a detailed explanation of each function.

String

`Len` `Contains` `RuneCount` `HasPrefix` `HasSuffix` `Index` `Join` `LastIndex` `Repeat` `Replace` `Split` `ToLower` `ToUpper` `Trim` `TrimLeft` `TrimRight` `TrimPrefix` `TrimSuffix`

Type conversion

`ToBool` `ToInt` `ToFloat` `ToString` `ToDate` `ToBinary` `ToJSON`

Encoding

`URLPathEscape` `URLPathUnescape` `URLQueryEscape` `URLQueryUnescape` `MD5` `SHA1` `SHA256` `BASE64` `DecodeBASE64` `BASE64URL` `DecodeBASE64URL`

Date Time

`Now` `Year` `Month` `Day` `Hour` `Minute` `Second` `Weekday` `ToLocal` `ToUTC` `ToTimezone` `DateFormat` `DateParse`

Environment Variable

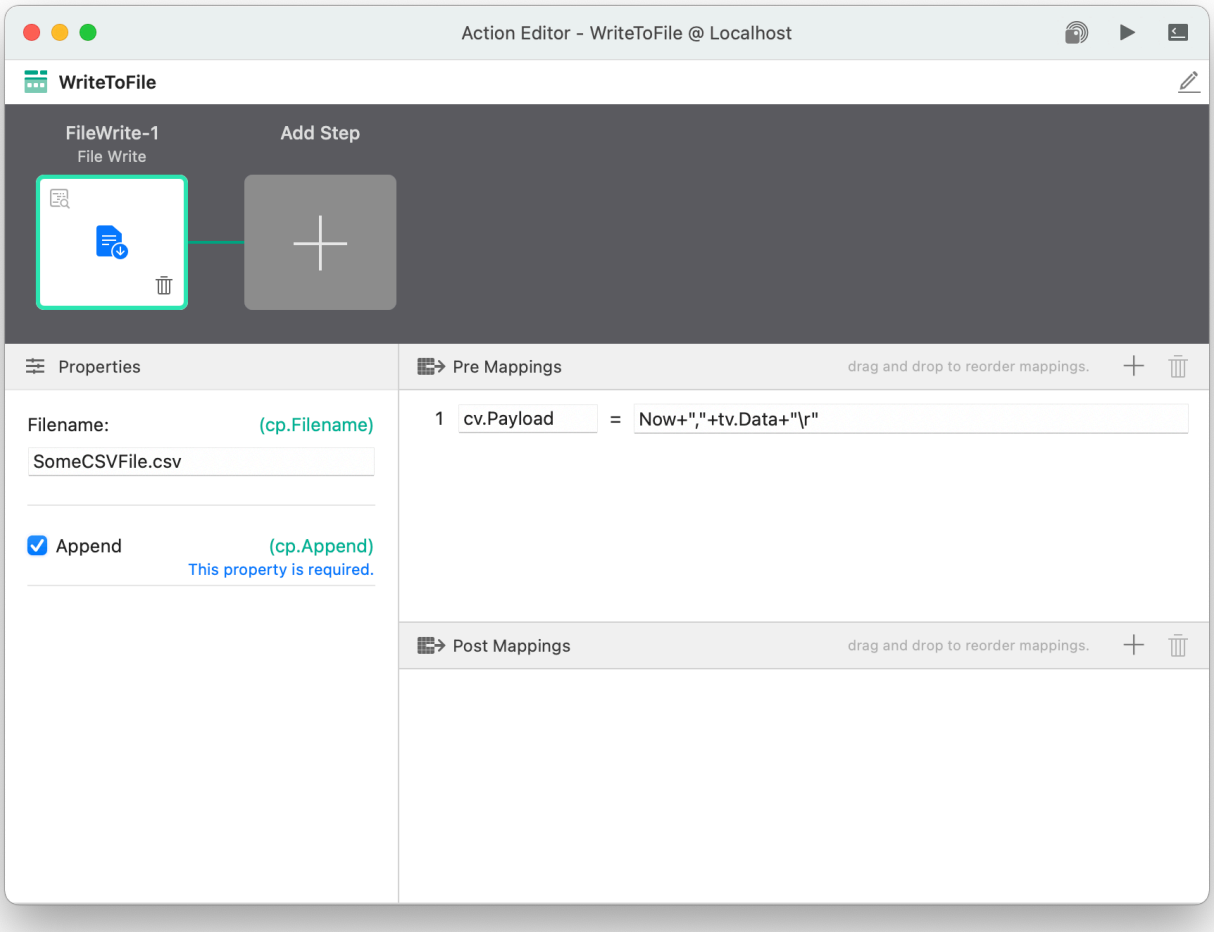
`Env`

Regular Expression

`RegExpMatch` `RegExpFind` `RegExpFindSubmatch` `RegExpFindAll` `RegExpReplace` `UUID`

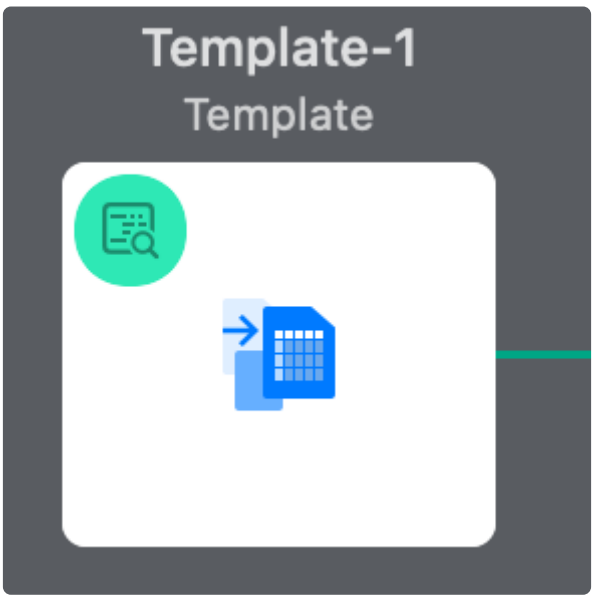
Example

An example usage of the `Now` function looks like this:


Action Editor - WriteToFile @ Localhost
WriteToFile
FileWrite-1
File Write
Add Step
Properties
Filename:
(cp.Filename)
SomeCSVFile.csv
Append
(cp.Append)
This property is required.
Pre Mappings
drag and drop to reorder mappings.
1
cv.Payload
=
Now+","+tv.Data+"\r"
Post Mappings
drag and drop to reorder mappings.

6.3.7. Debug Log

Press the button at the top left of the component to output the debug log.



By nature, components run independently and in parallel, but when executing an action while outputting debug logs, they are synchronized within the action engine.

The debug log outputs the properties and variables of `tp.` (trigger properties) / `tv.` (trigger variables) / `ap.` (action properties) / `av.` (action variables) / `cp.` (component properties) / `cv.` (component variables) in that order.

The action program is processed in the following order.

1. Get the component variables including Payload from the queue
2. Assignment is performed if there is a value of PostMappings passed from a previous component.
3. PreMappings expression evaluation and value assignment are conducted.
4. The Component's main functionalities are executed according to the set properties.
5. A the component variable `cv.Payload` is updated to the new content, depending on the result of the component's output.
6. PostMappings is evaluated only as an expression and the resulting value is stored in a component variable containing the new `cv.Payload`.
7. The newly filled `cv.Payload` is sent to the next component's input queue.
8. For the next component, the process starts over from number 1 above.

If you turn on the debug log function, in the processing procedure of the above-mentioned component, the content of the variable after the expression evaluation and assignment of PreMappings of 3 is printed in the console log.

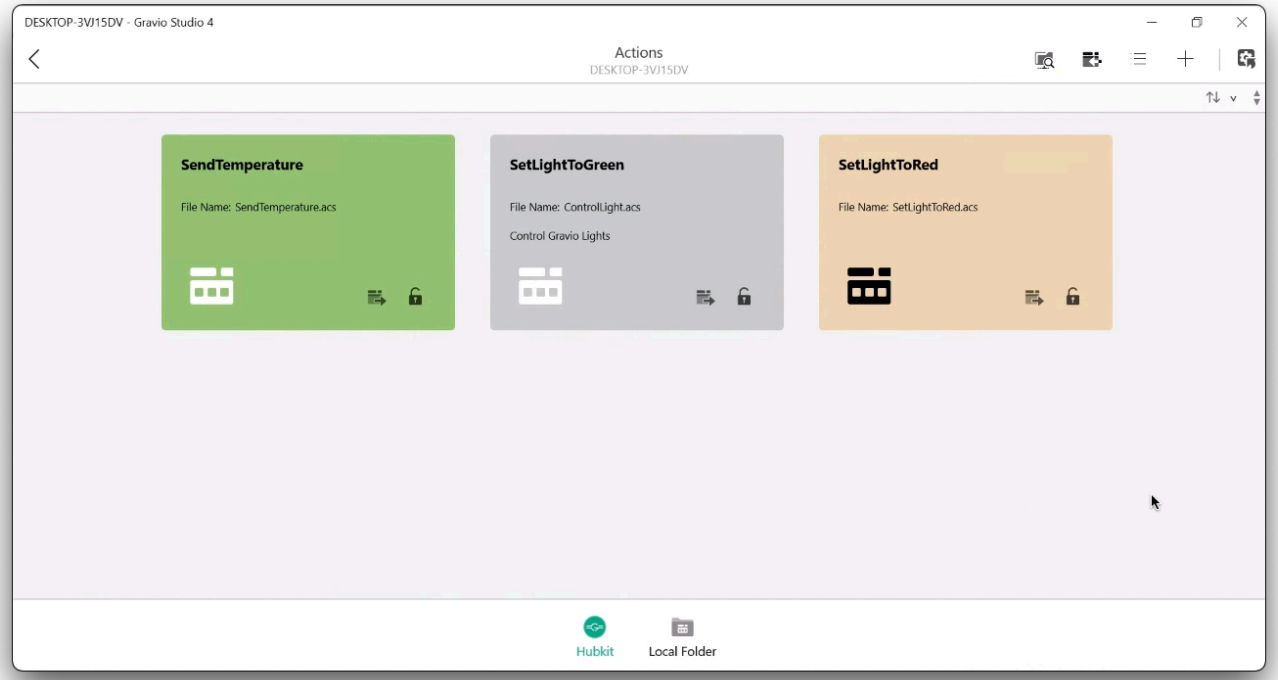
6.3.8. Using the File Viewer

Some action components in HubKit are able to read files in the actmgr/data directory and save files in that directory.
Files stored in this directory can be retrieved by Gravio Studio, and local files can be uploaded to the directory by Gravio Studio.

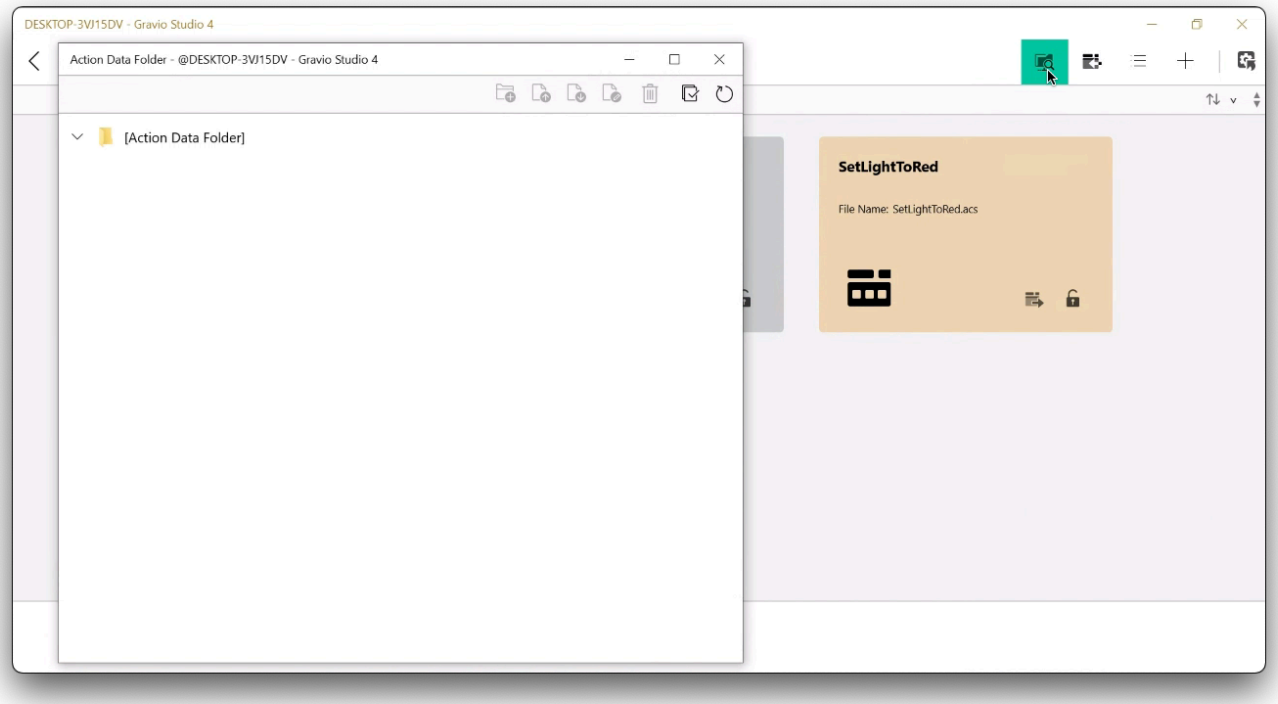
Use the following UI to do this.

Opening the Finder View

Open the Actions view:



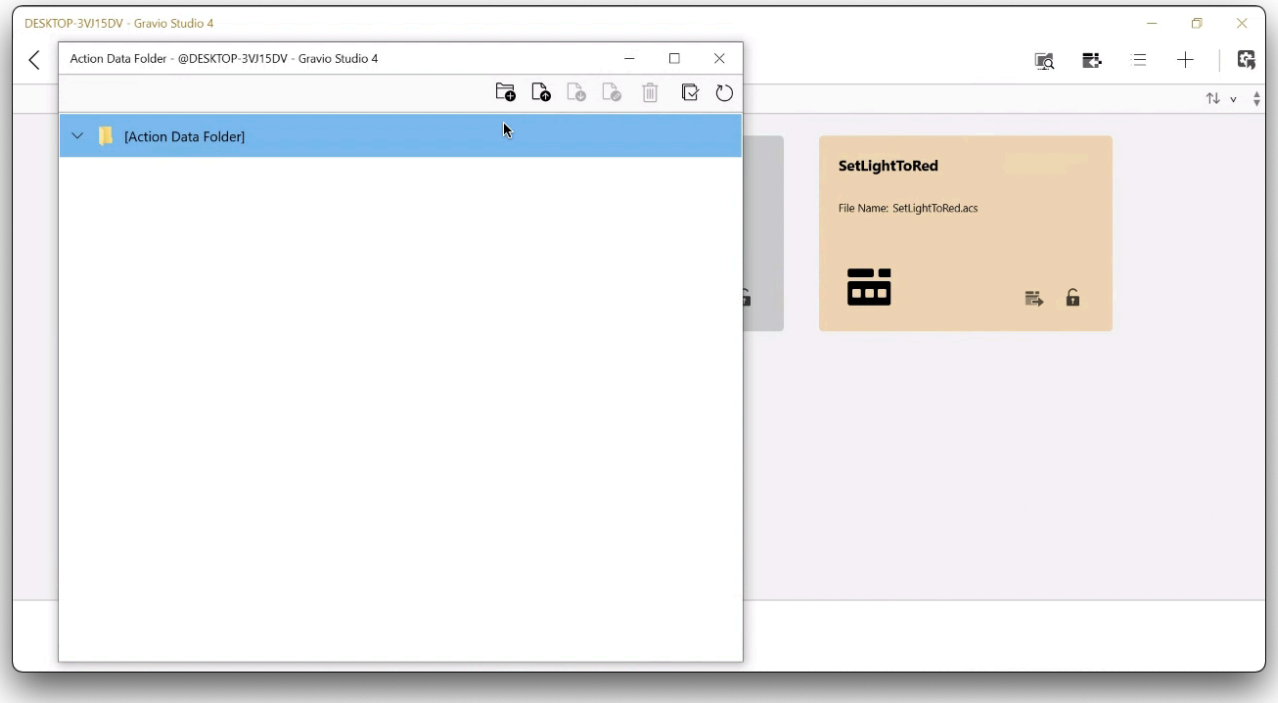
Once in there, open the file viewer at the top right:



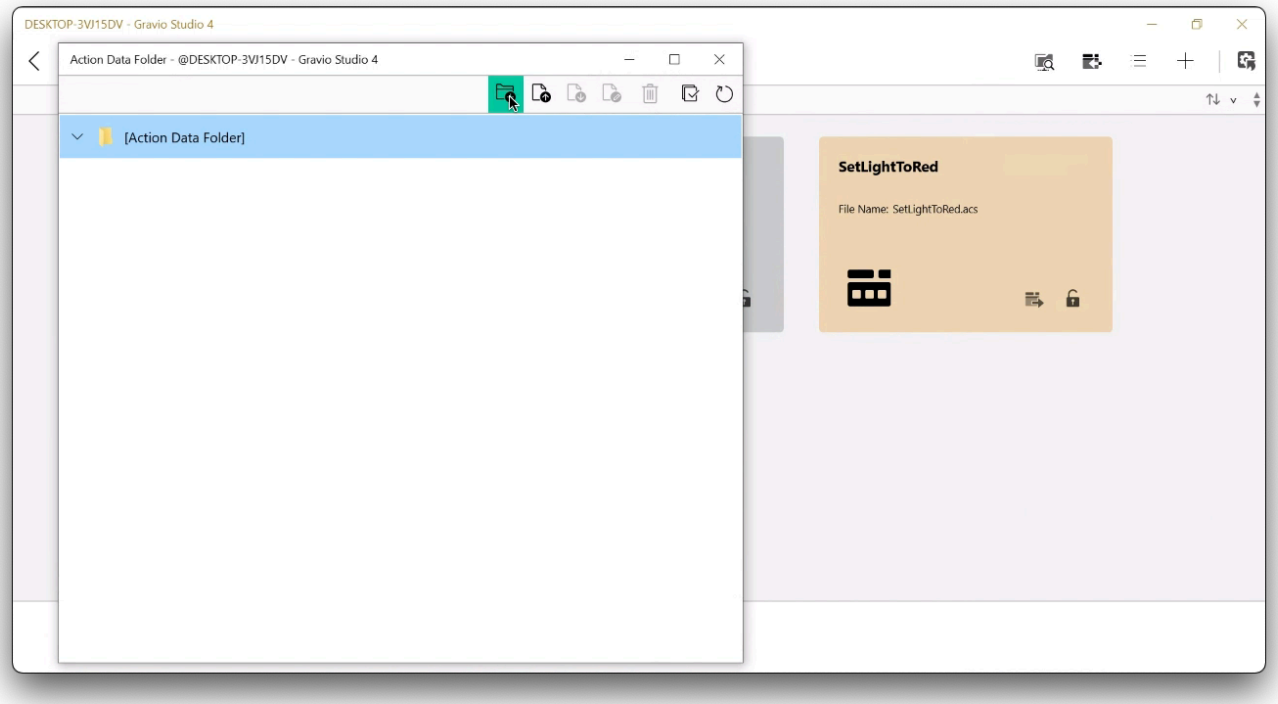
A new window will open. This represents the files located on your HubKit in your `actmgr/data` folder in tree format.

Create Folders

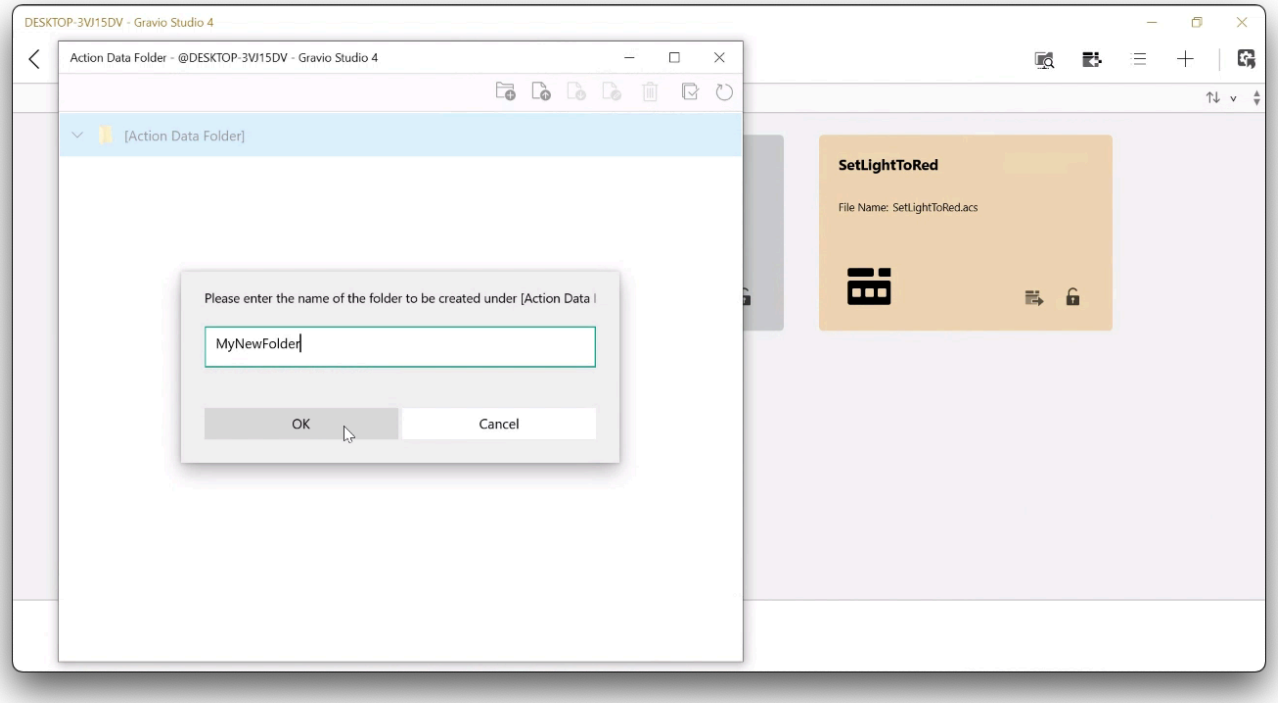
To create a subfolder, select the parent folder:



And click on the “Create Folder” icon:



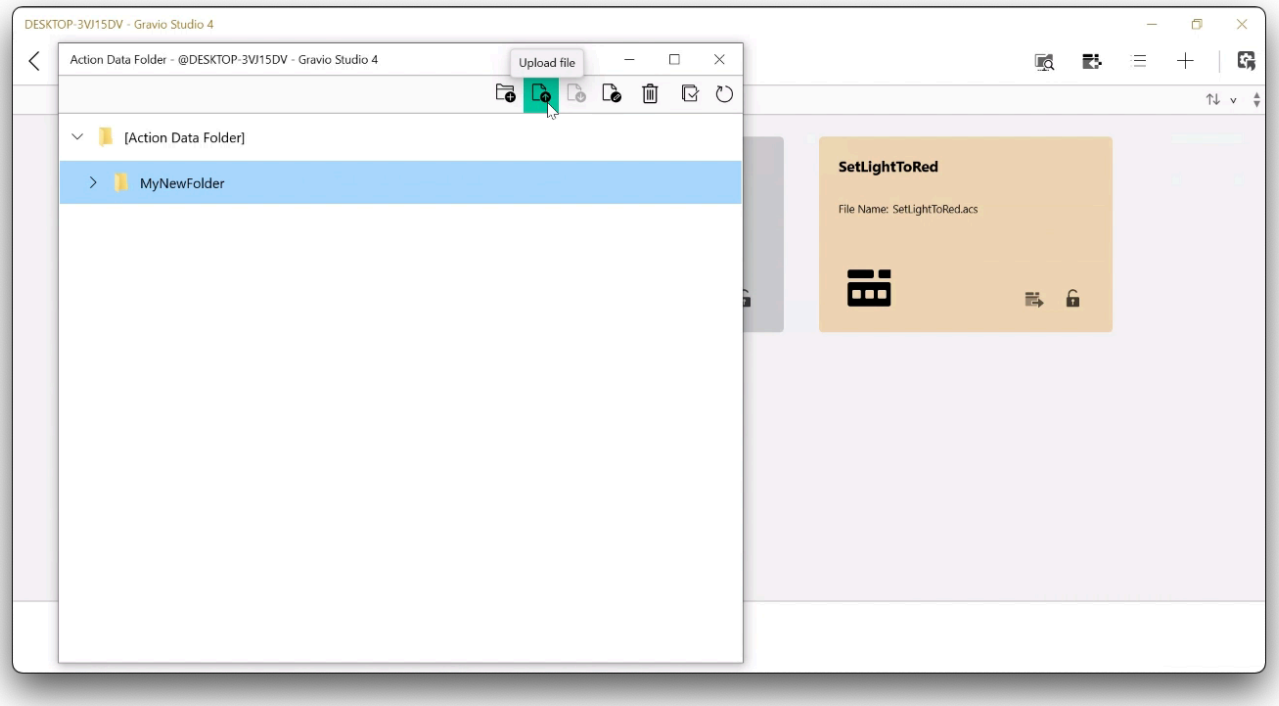
Pick a good name for your folder



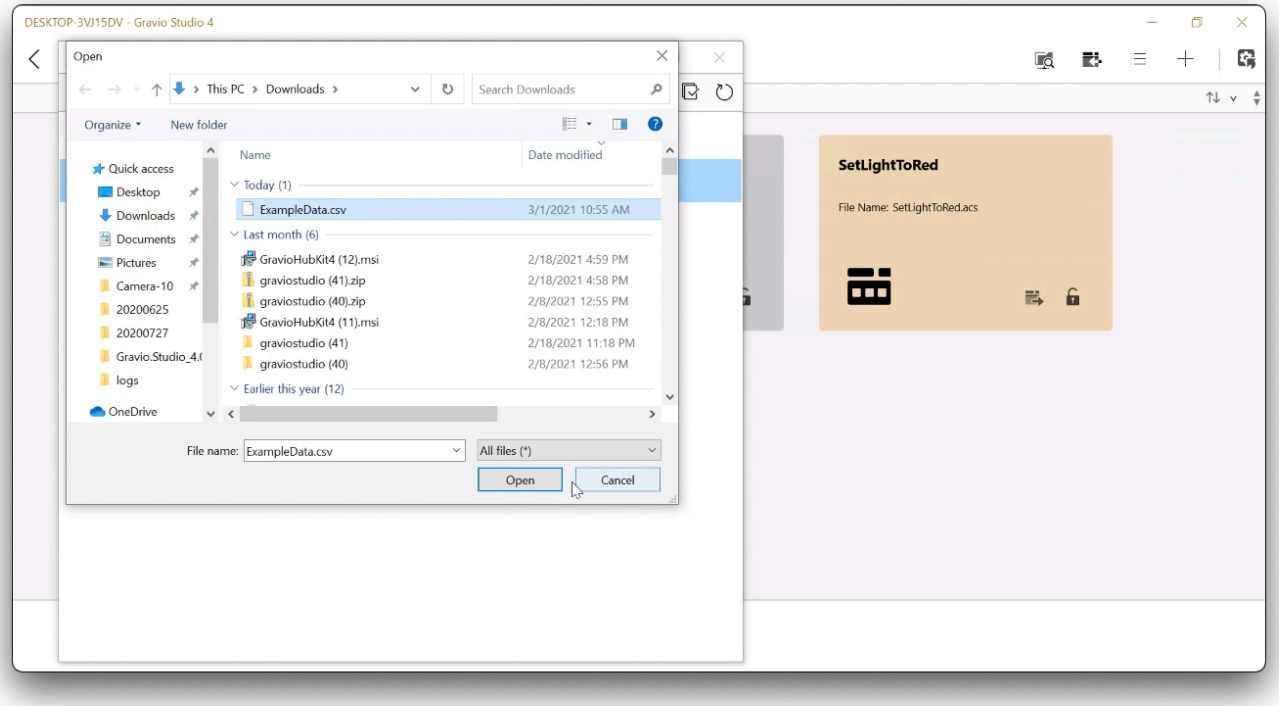
It will be created in your parent folder.

Uploading a file

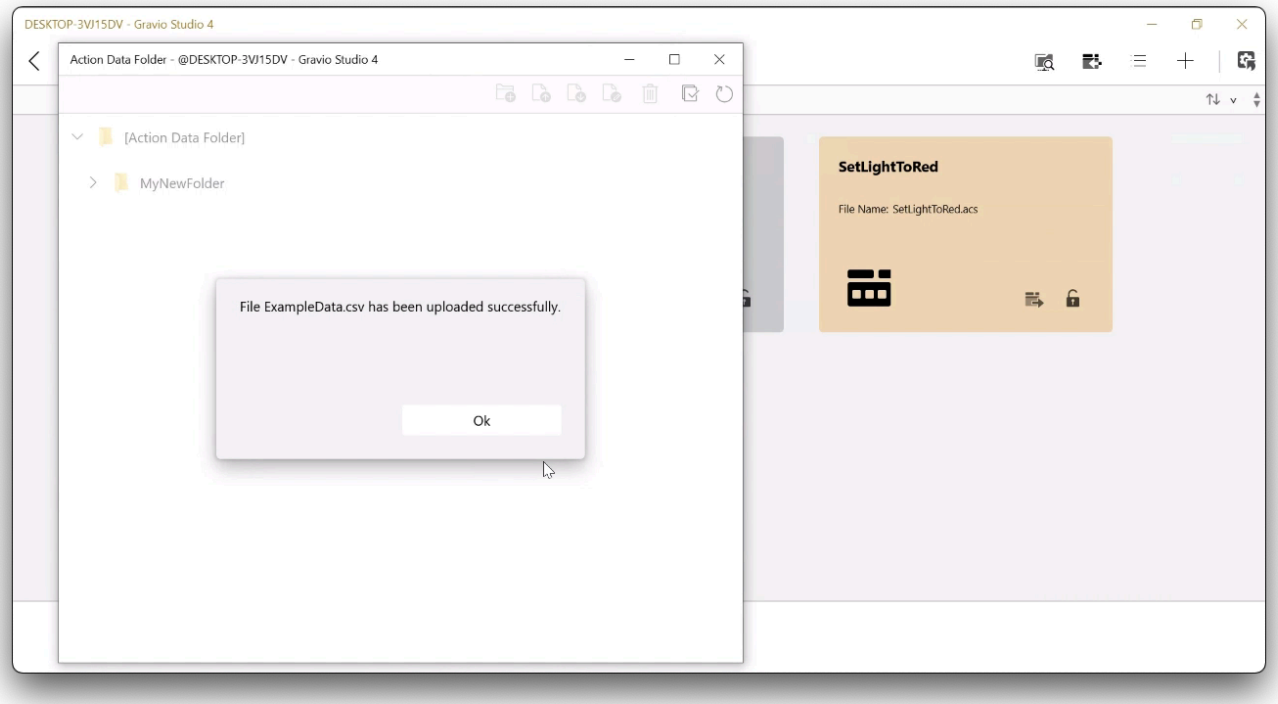
Select the folder you like to upload a file into, and click the “file upload” button:



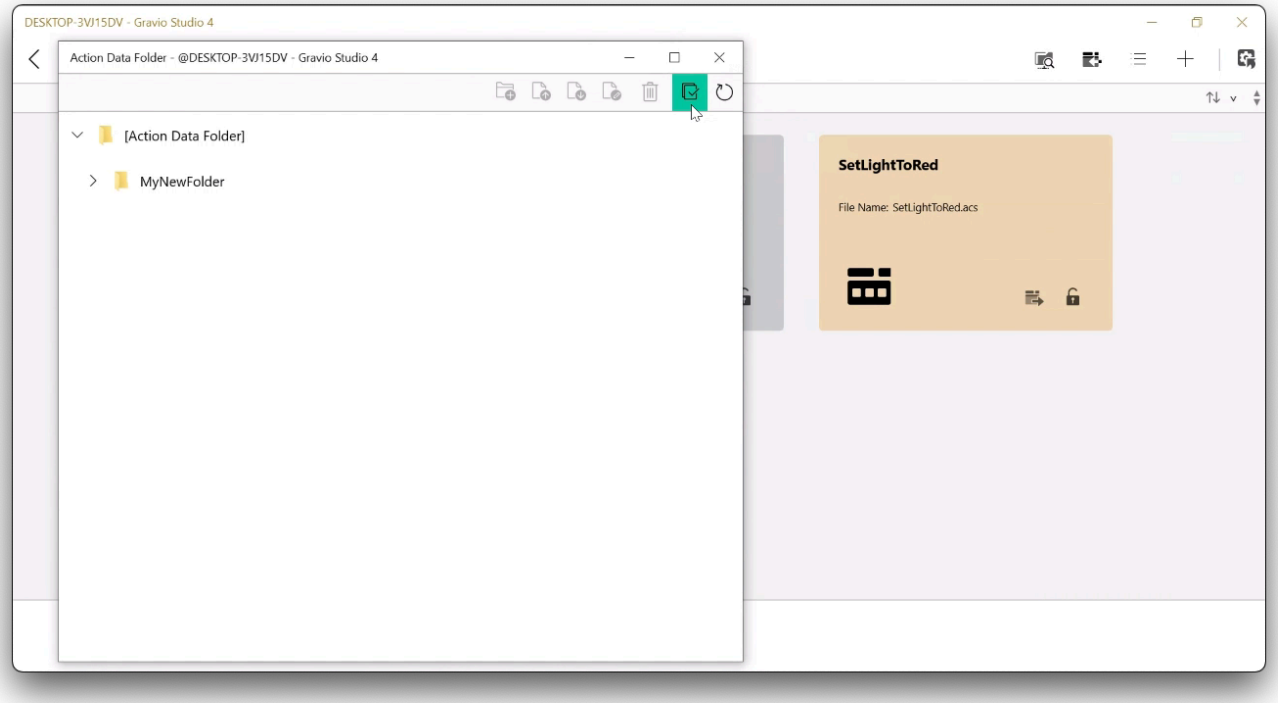
Pick your file on your local machine:

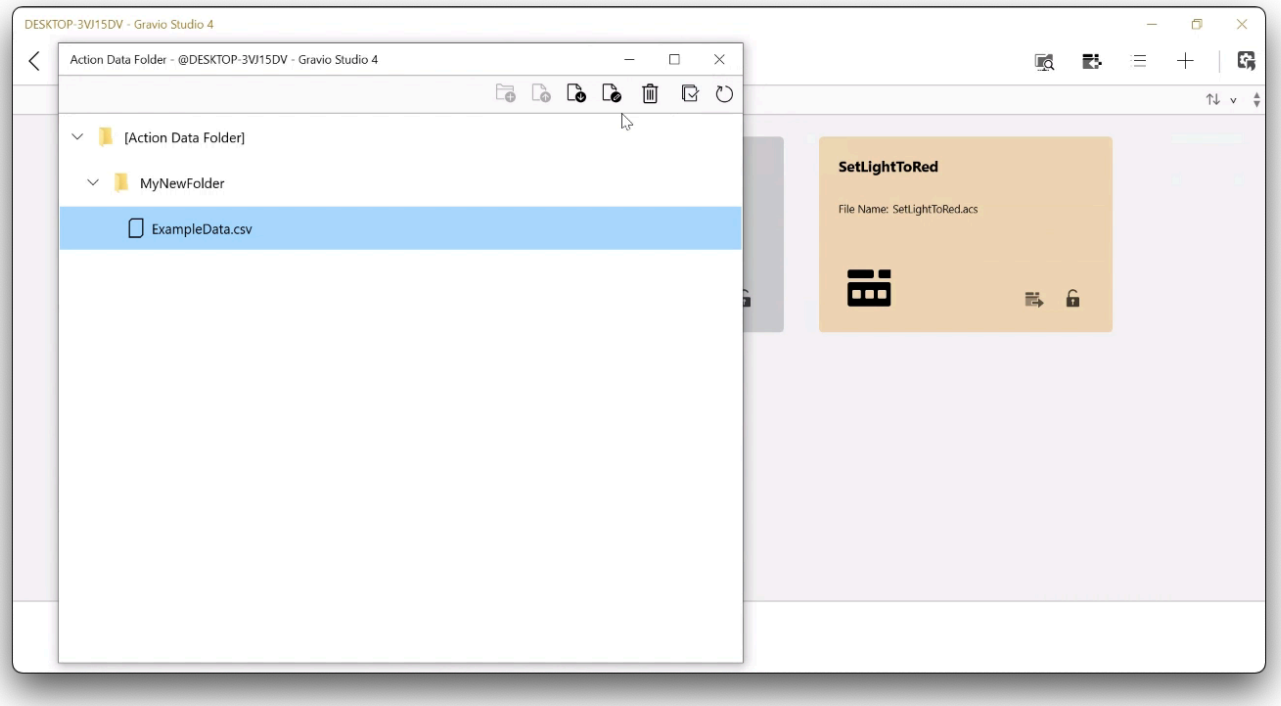


After confirmation you will see a message:



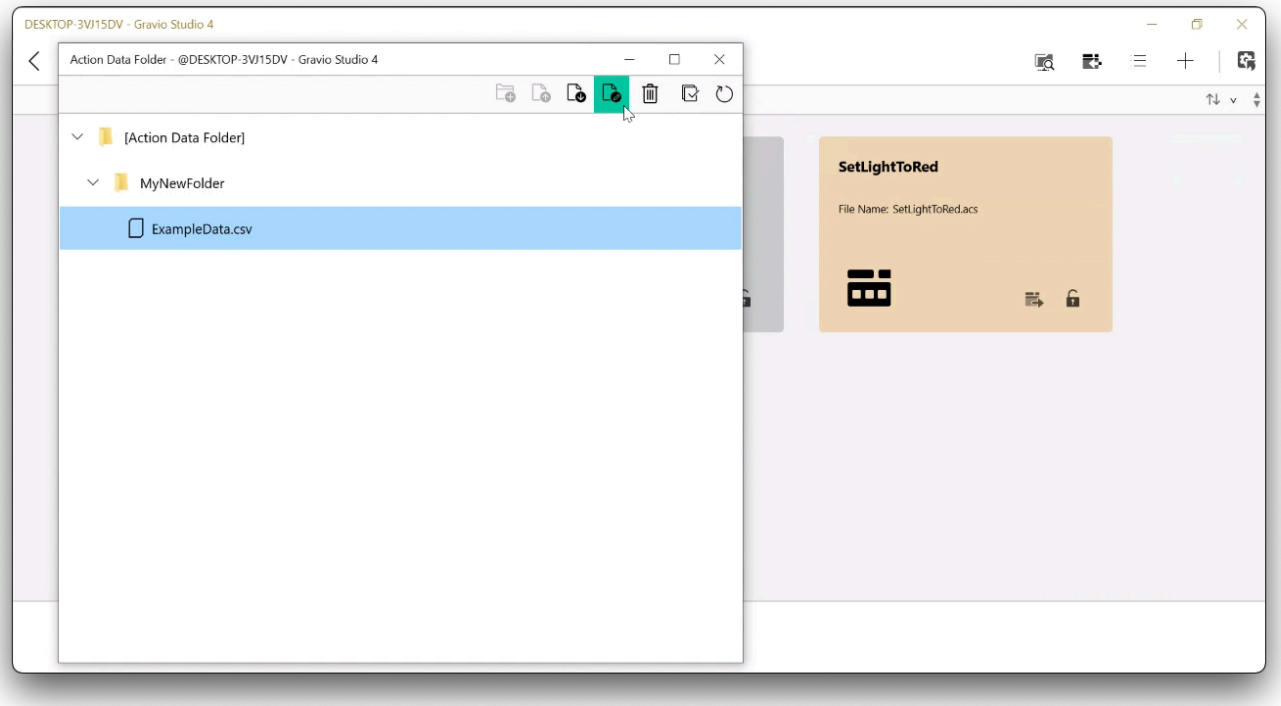
Click on the “Expand all” button or on the expand arrow next to your folder to see the file:



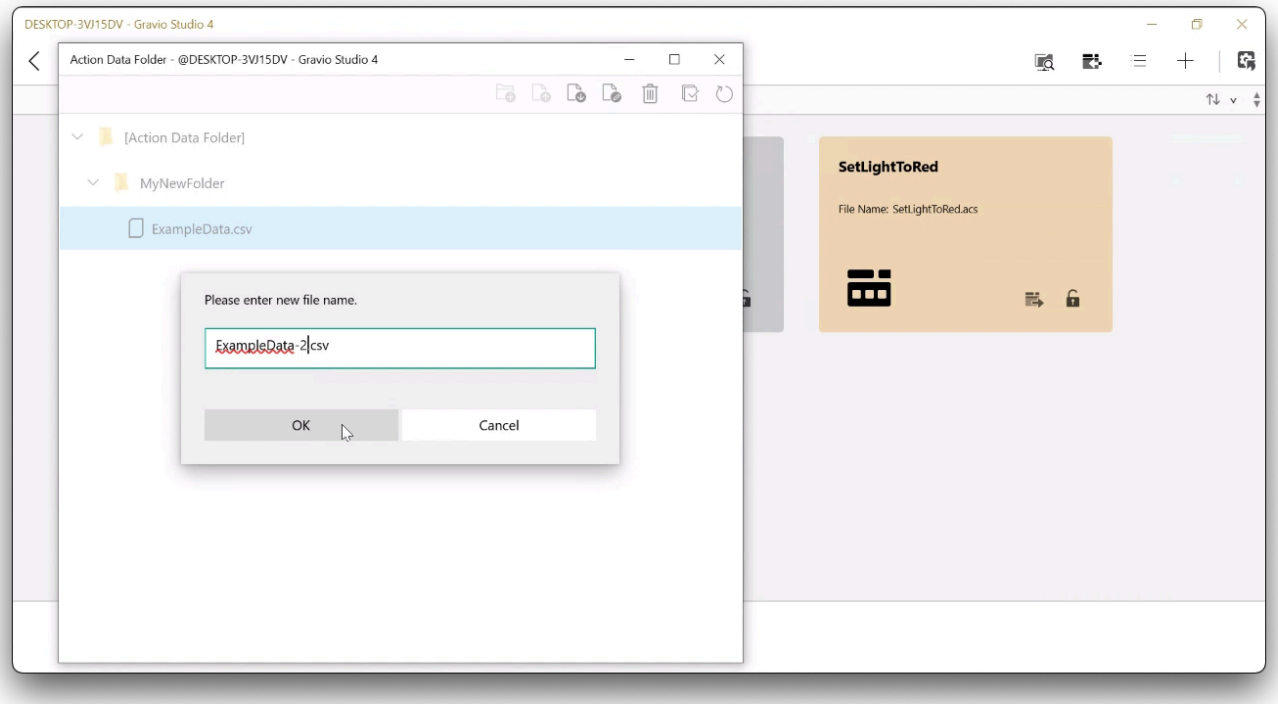


Renaming a folder or file

You can rename a folder or file by selecting it and clicking on the rename button:

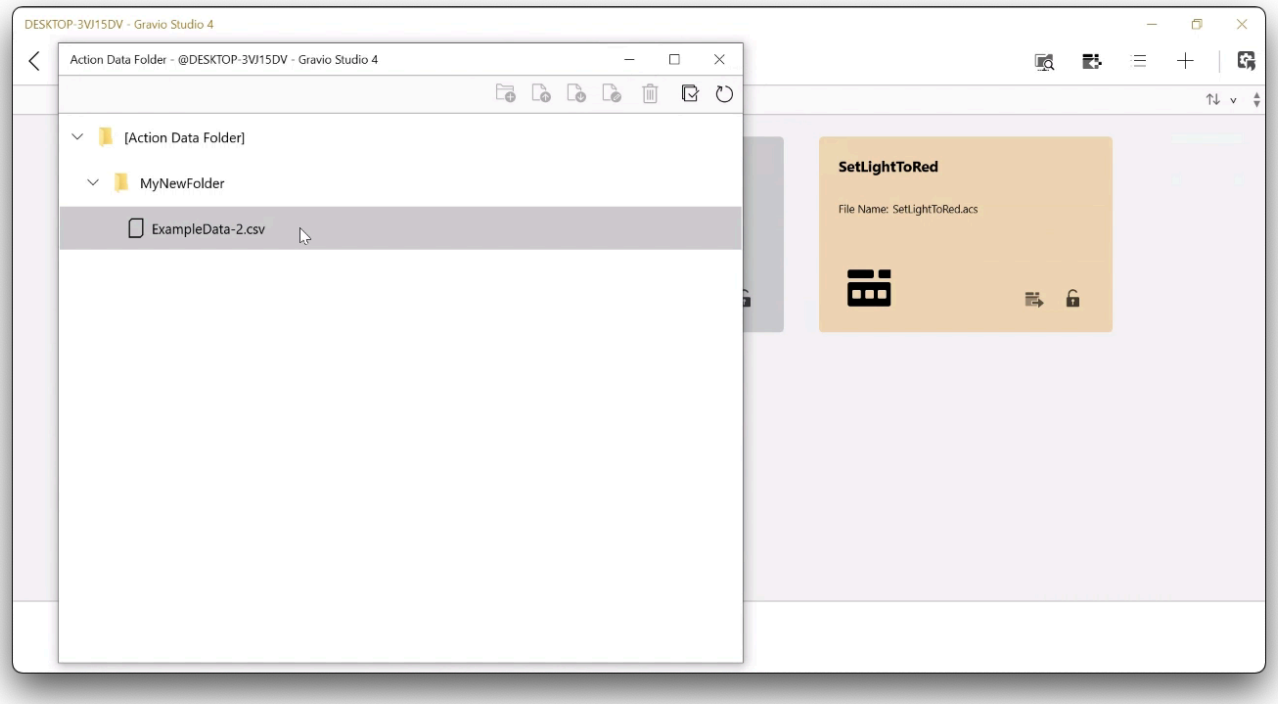


Make sure you give the file a meaningful name:

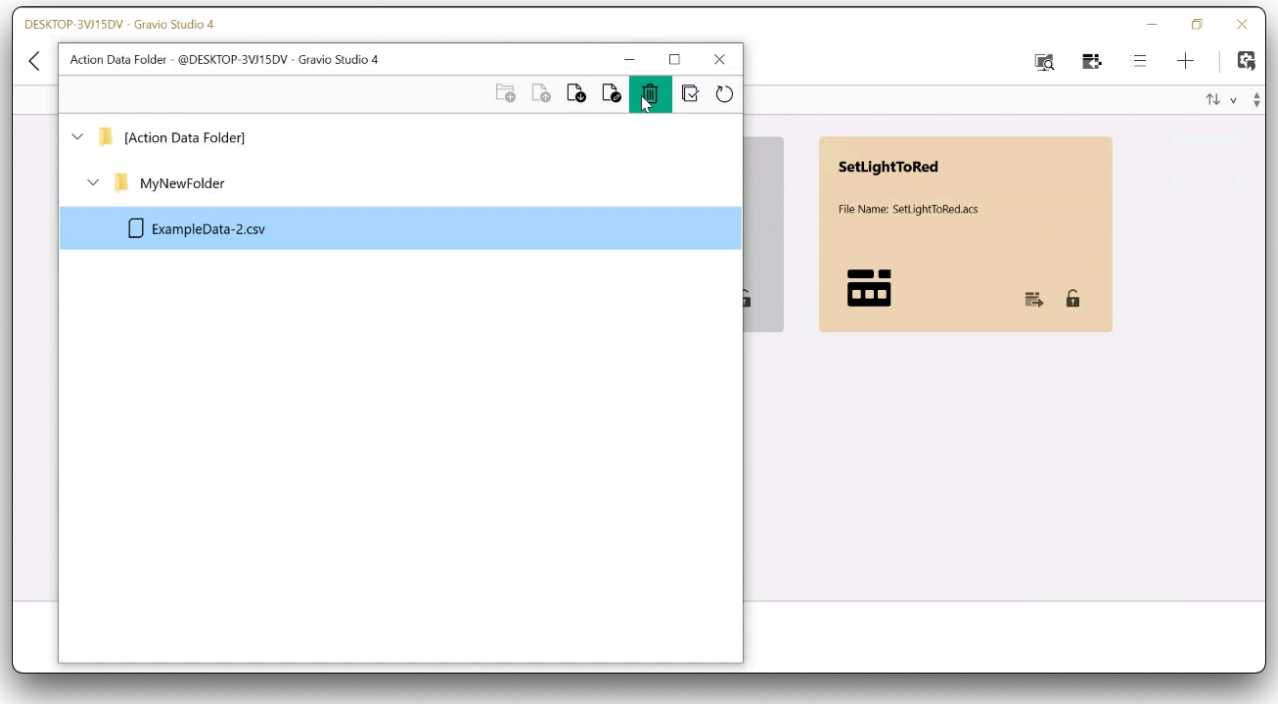


Deleting a file or folder

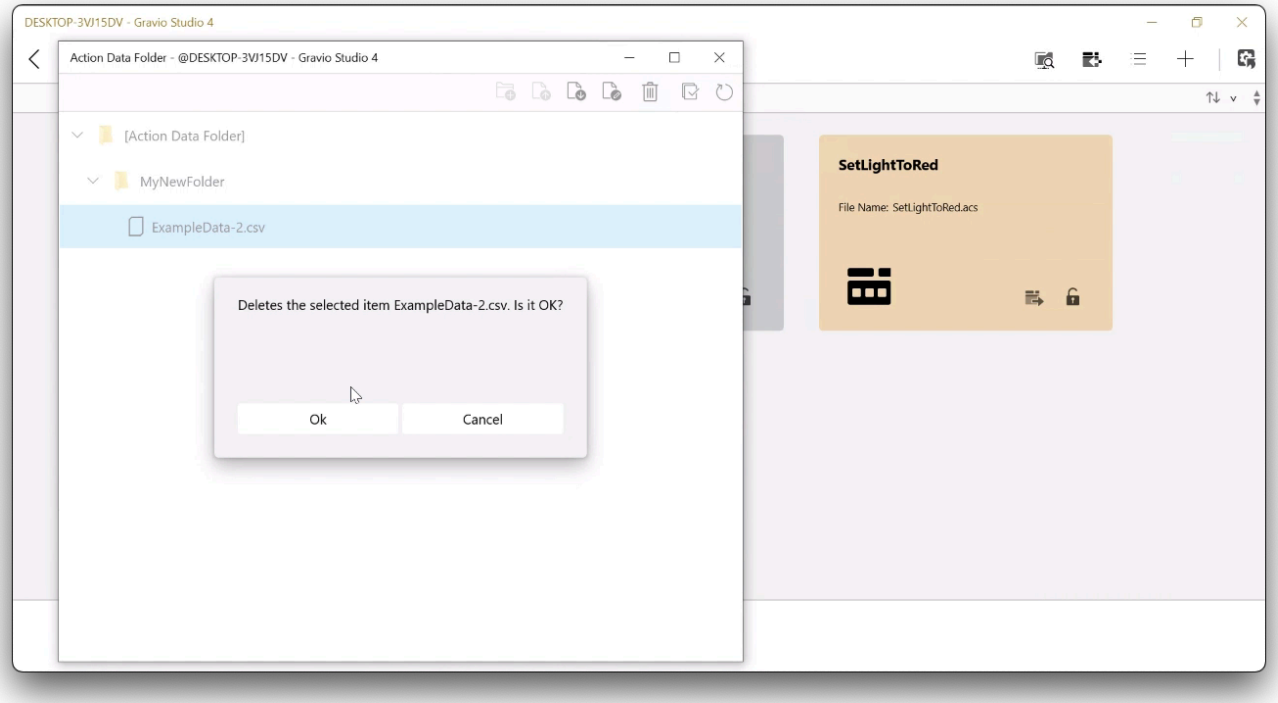
Pick the file or folder you like to delete



Click the trash can to delete it:



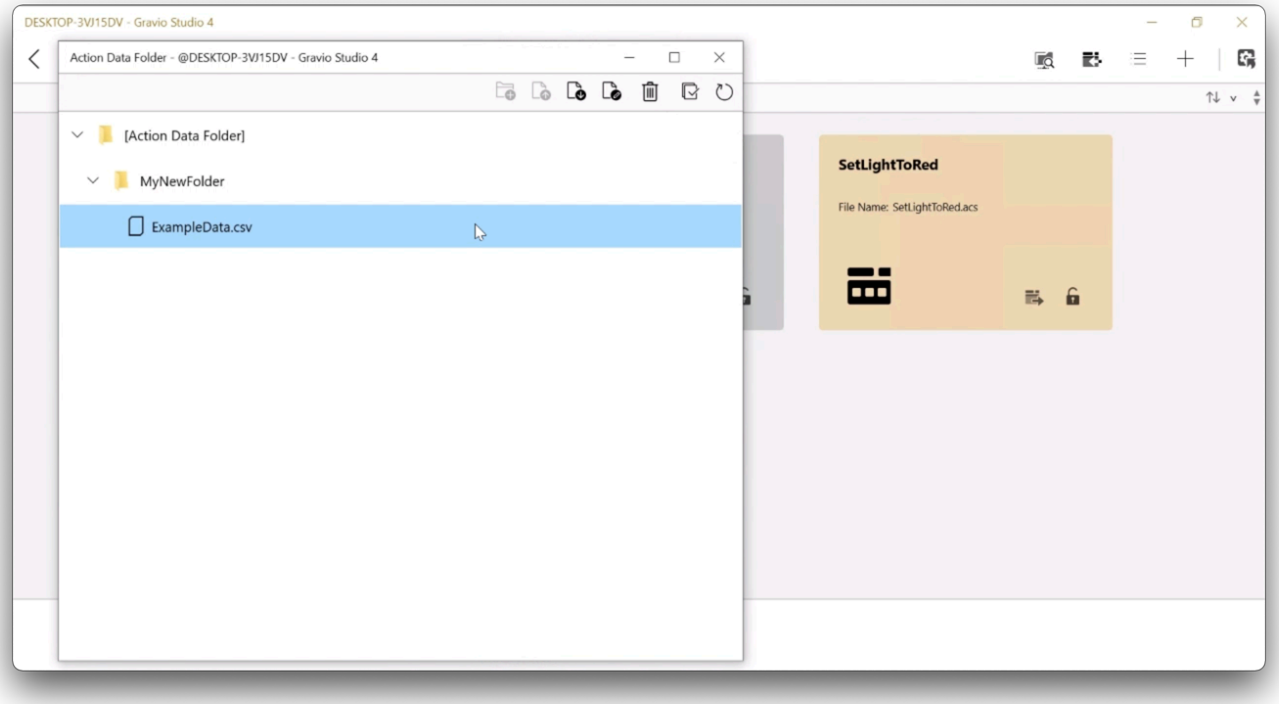
You will be prompted to confirm the deletion:



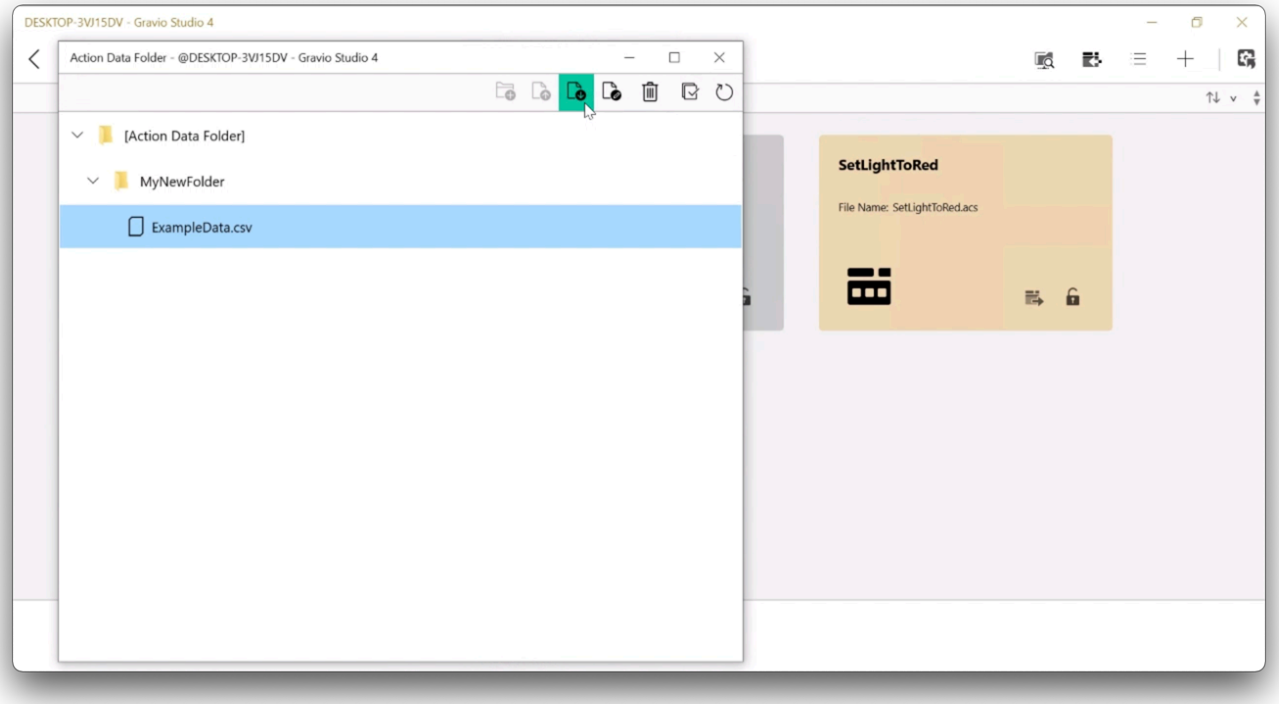
Downloading a file

To download files created by Gravio HubKit from the HubKit to your computer, follow the steps here:

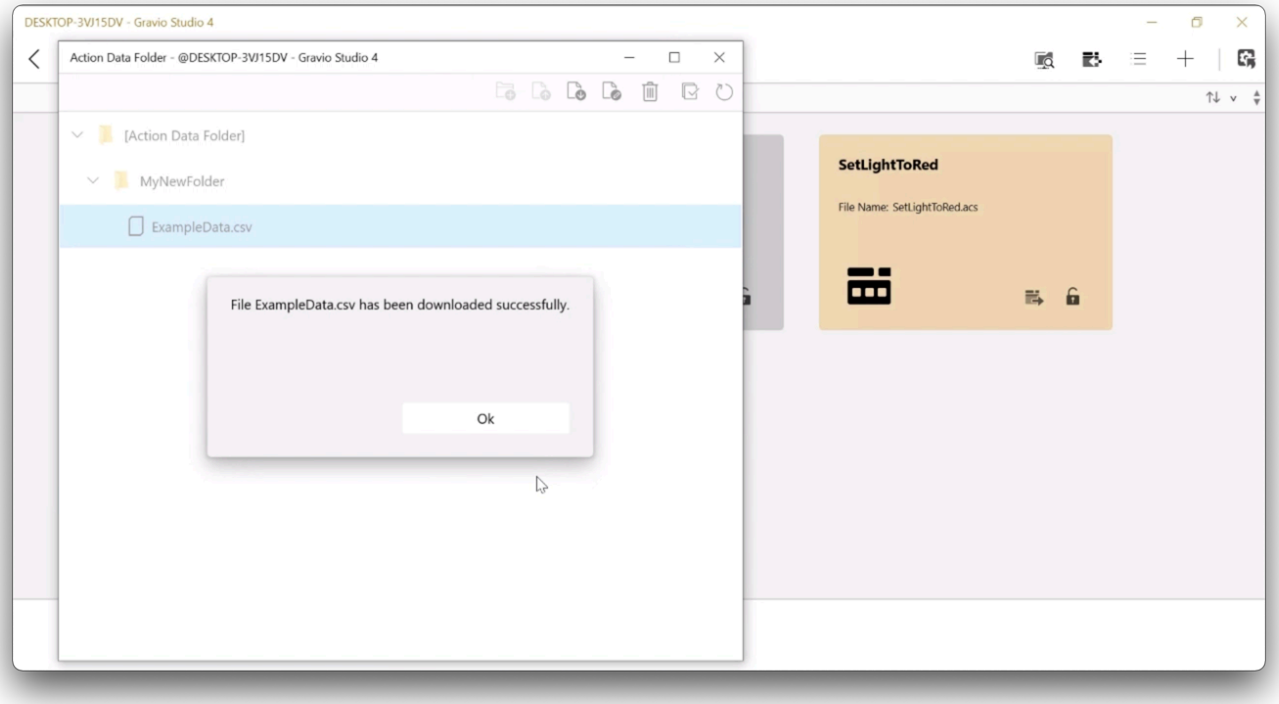
Select the file you like to download:



Click the “Download button”:

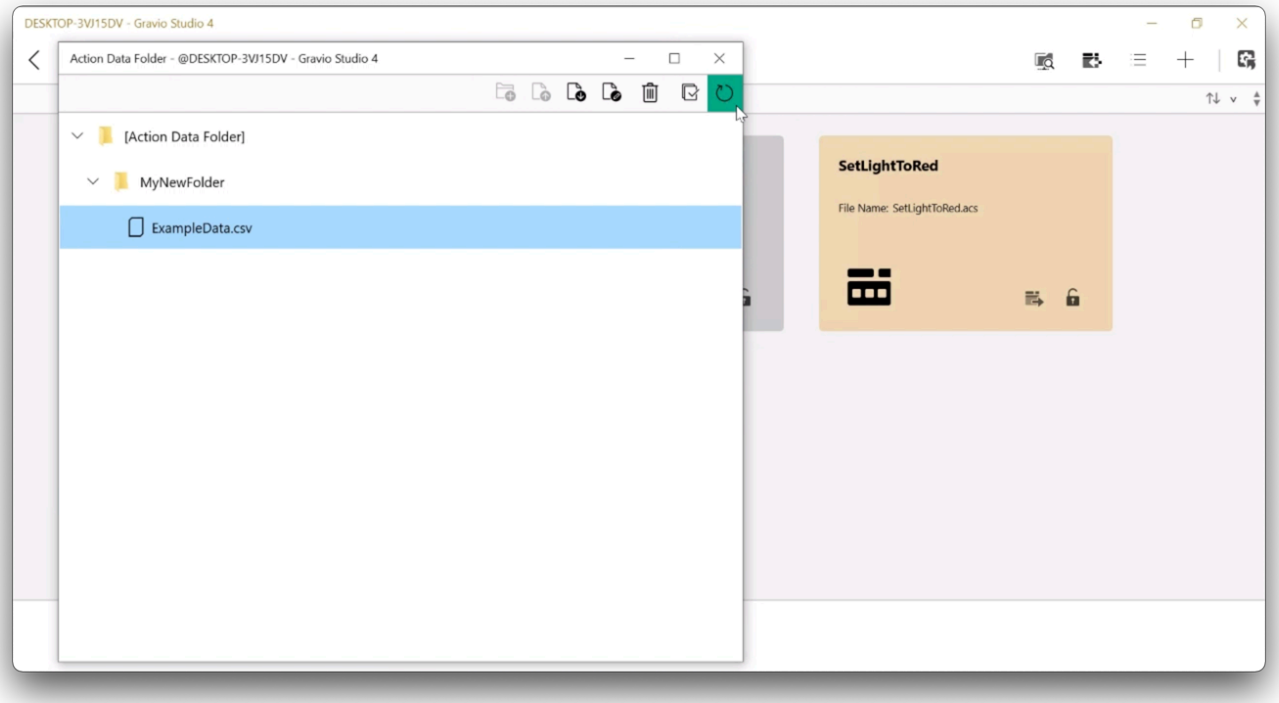


Pick a destination and name on your computer, and after confirmation you shluld see the following message:



Refreshing the view

As Gravio HubKit can create new files at any time, you may want to refresh the files view every now and then. You can press the button in the top right to do so:



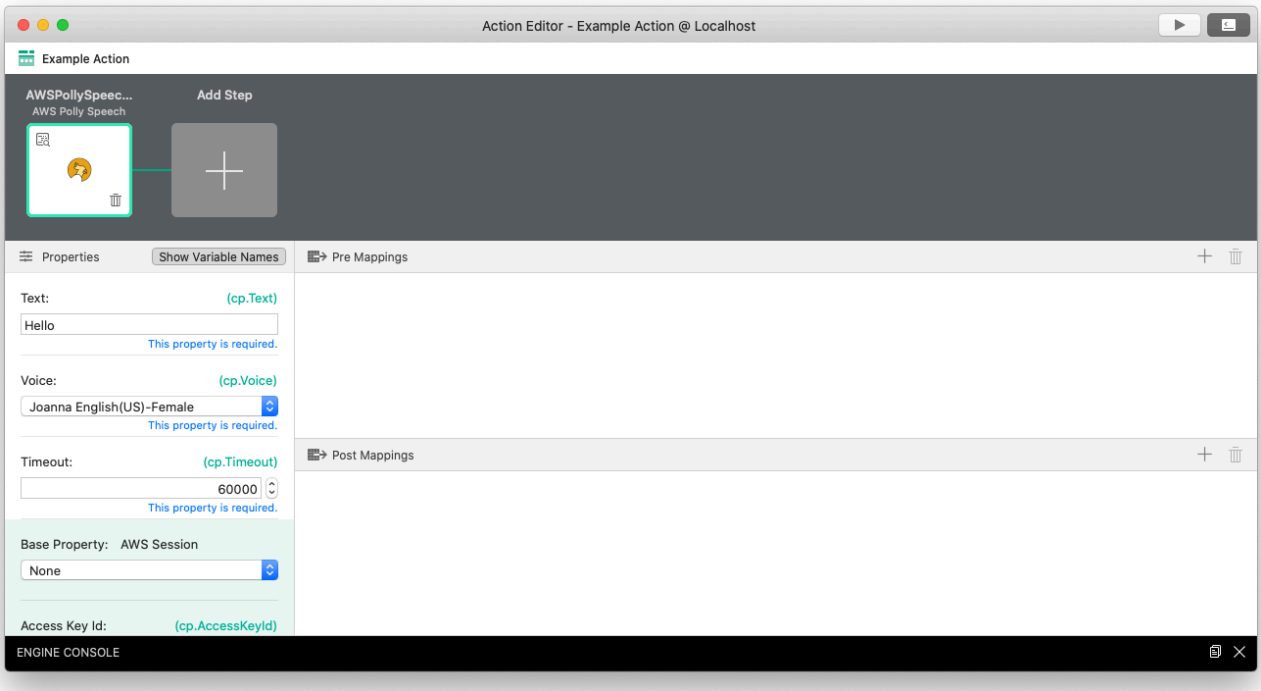
6.3.9. Action Components

An action consists of one or more steps. Each step can be populated with action components and define inputs and outputs.

This section describes the different components available and explains how to use them.

6.3.9.1. Amazon Polly Speech

The AWS Polly Speech Action Component is to read out text using AWS Polly text-to-speech. Amazon Polly converts input text into life-like speech. This functionality can also be useful for debugging purposes.



Note that this functionality may not be available for your operating system. Playing sounds is currently supported under Windows and Linux, but not yet on macOS.

Component Properties

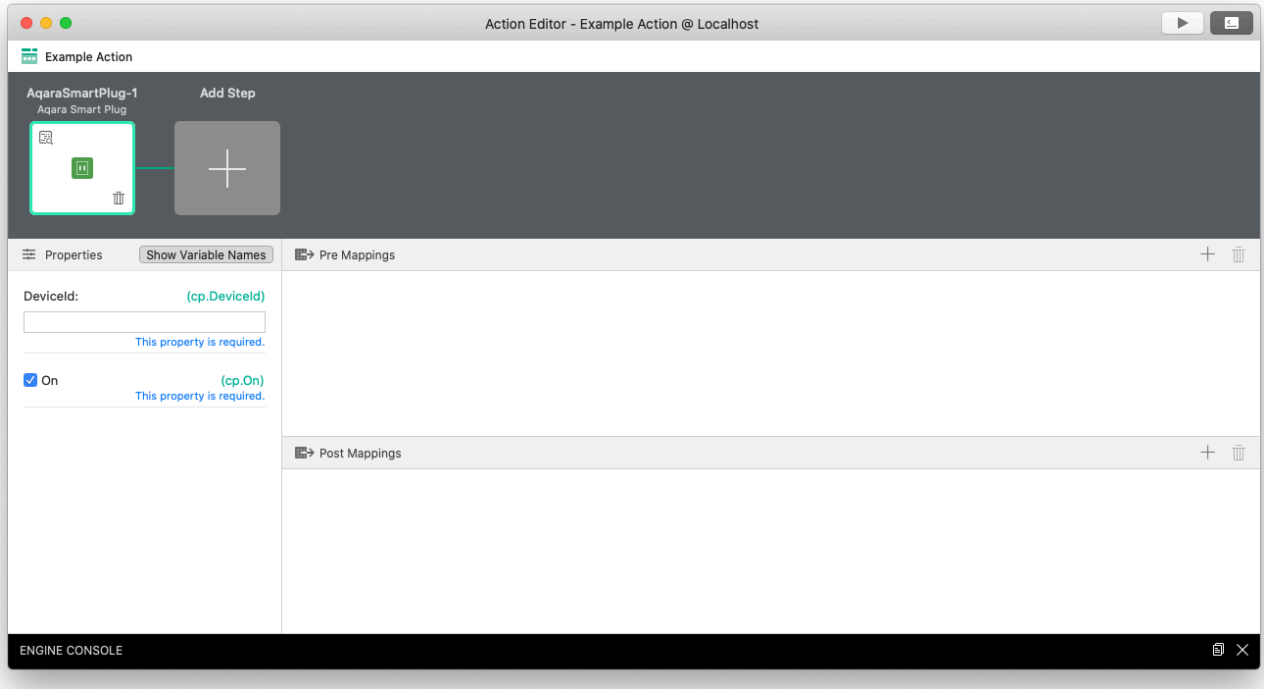
Name	Required	Description
Text	True	Specify the message text you want to speak. If you like to include sensor data, you can set this message using Pre-Mapping, for example, using `cp.Text = "The current temperature is"+cv.Payload+"°C"`
Voice	True	Specify the type of voice of the speaker. Select a voice type in the same language as the text language. If you want to set it by cp.Voice value, you can use “Zeina”, “Zhiyu”, “Naja”, “Mads”, “Lotte”, “Ruben”, “Nicole”, “Russell”, “Amy”, “Emma”, “Brian”, “Aditi”, “Raveena”, “Ivy”, “ Joanna”, “Kendra”, “Kimberly”, “Salli”, “Joey”, “Justin”, “Kevin”, “Matthew”, “Geraint”, “Celine”, “Mathieu”, “Chantal”, “Marlene”, “Vicki”, “ Hans”, “Aditi”, “Dora”, “Karl”, “Carla”, “Bianca”, “Giorgio”, “Mizuki”, “Takumi”, “Seoyeon”, “Liv”, “Ewa”, “Maja”, “Jacek”, “Jan”, “Camila”, “ Vitoria”, “Ricardo”, “Ines”, “Cristiano”, “Carmen”, “Tatyana”, “Maxim”, “Conchita”, “Lucia”, “Enrique”, “Mia”,

		“Lupe”, “Penelope”, “Miguel”, “ Astrid,” “Filiz,” or “Gwyneth.”
Method	True	`Play` Audio or `Save` Audio File
Filename	False	Specify the file name in case of file saving
Timeout	True	Specify the transmission timeout time (ms)
AWS Session	False	AWS Session from the Base Property Profile
Access Key Id	True	Access Key Id if not taken from the Base Property Profile
Secret Access Key	True	Secret Access Key if not taken from the Base Property Profile
Region	True	Region if not taken from the Base Property Profile

You can find you access keys on your Amazon console.

6.3.9.2. AqaraSmartPlug

The Aqara Smart Plug Action Component is to control an Aqara smart plug device. To use this device, you must have the Aqara Smart Plugs paired in advance.



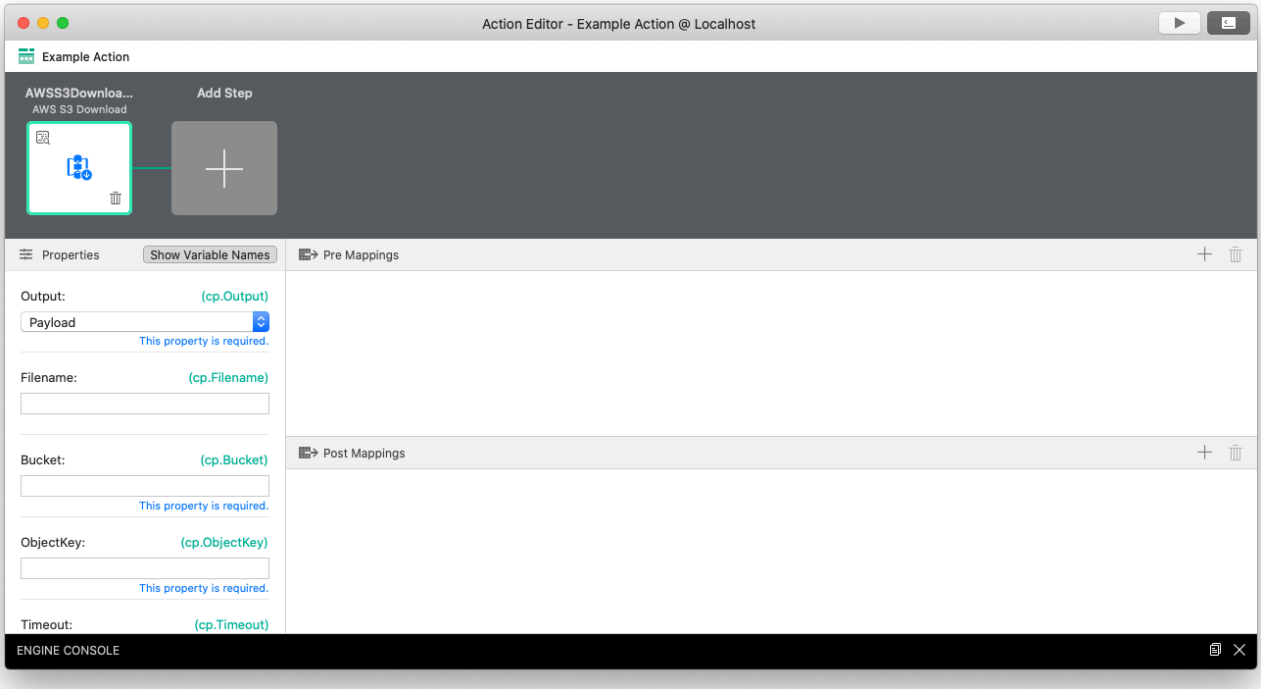
Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

Component Properties

Name	Required	Description
DeviceId	True	Device IDs to be turned on/off. If none are configured, all Aqara smart plugs are addressed. Multiple IDs can be separated by commas.
On	True	On. The value specified for cp.On is “true” or “false”.

6.3.9.3. AWS S3 Download

The AWS S3 Download Action Component is to download content from Amazon AWS S3.



Component Properties

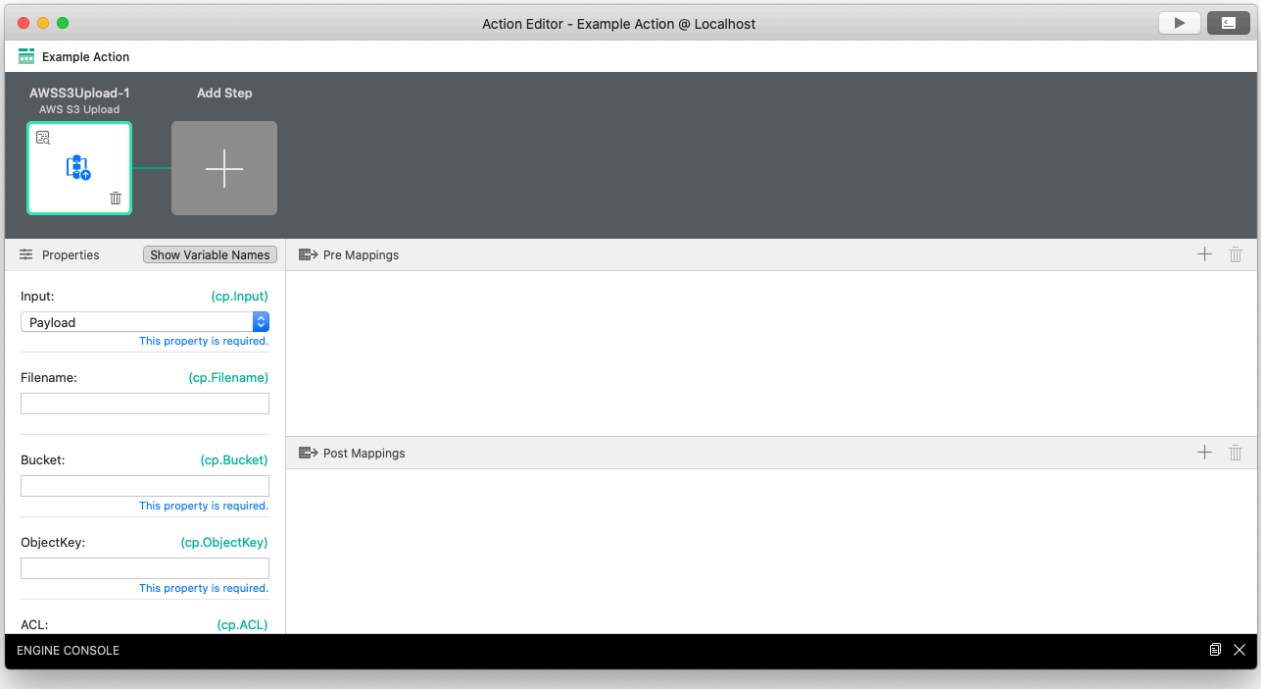
Name	Required	Description
Output	True	Specify whether to keep the content in the Payload variable of the Action or as a file. cp.Output is set to either “Payload” or “File”
Filename	False	File name when downloading as a file
Bucket	True	Bucket name in AWS S3
ObjectKey	True	Object key in AWS S3
Timeout	True	Specifying the transmission timeout time in ms
AWS Session	False	AWS Session
Access Key Id	True	Access key obtained from AWS
Secret Access Key	True	Secret access key obtained from AWS
Region	True	Specify the AWS region to use

Use the base Base Property Profile property to read the values from the settings rather than specifying the details within the component.

You will need your AWS Key and Secret, which you can obtain from your Amazon console.

6.3.9.4. AWS S3 Upload

The AWS S3 Upload Action Component is to upload content to amazon aws s3.



Component Properties

Name	Required	Description
Input	True	Upload a payload or a file. To set the value of cp.Input, use either “Payload” or “File”.
Filename	False	The input file name
Bucket	True	Bucket name to upload to in AWS S3
ObjectKey	True	Object key in AWS S3
ACL	True	ACL permissions settings on AWS S3. To set the value in the cp.ACL, use one of the following: “private”, “public-read”, “public-read-write”, “authenticated-read”, “aws-exec-read”, “bucket-owner-read”, “bucket-owner-full-control”. owner-full-control”.
Storage Class	True	Storage type in AWS S3. When set by the value of cp.StorageClass, “STANDARD”, “REDUCED_REDUNDANCY”, “GLACIER”, “STANDARD_IA”, “ONEZONE_IA”, “INTELLIGENT_TIERING”, “DEEP_ARCHIVE “one of”.
Timeout	True	Specifying the transmission timeout time in ms
AWS Session	False	AWS Session

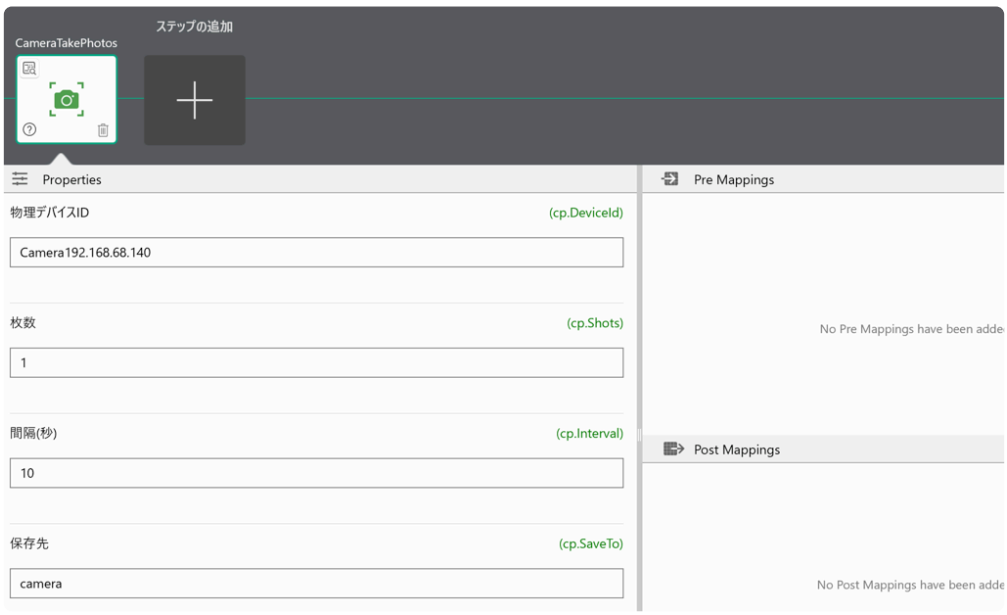
Access Key Id	True	Access key obtained from AWS
Secret Access Key	True	Secret access key obtained from AWS
Region	True	Specify the AWS region to use

Use the base Base Property Profile property to read the values from the settings rather than specifying the details within the component.

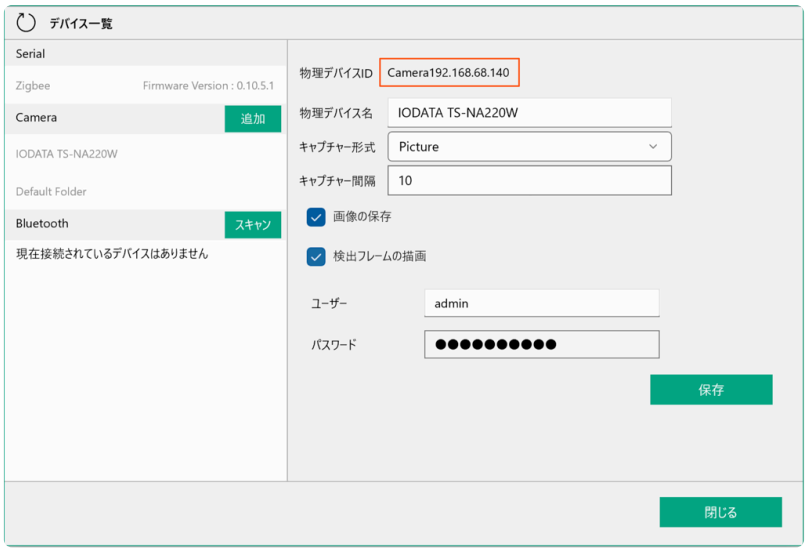
More details on Access Control List (ACL) Permissions and Storage Class.

6.3.9.5. Camera Take Photos

Save photos from the connected camera at the timing of the execution of the action.



Check the device list in advance to note down the physical device ID of the connected camera.



Output Payload	Description
`cv.Payload`	Outputs an array of file names of saved photos

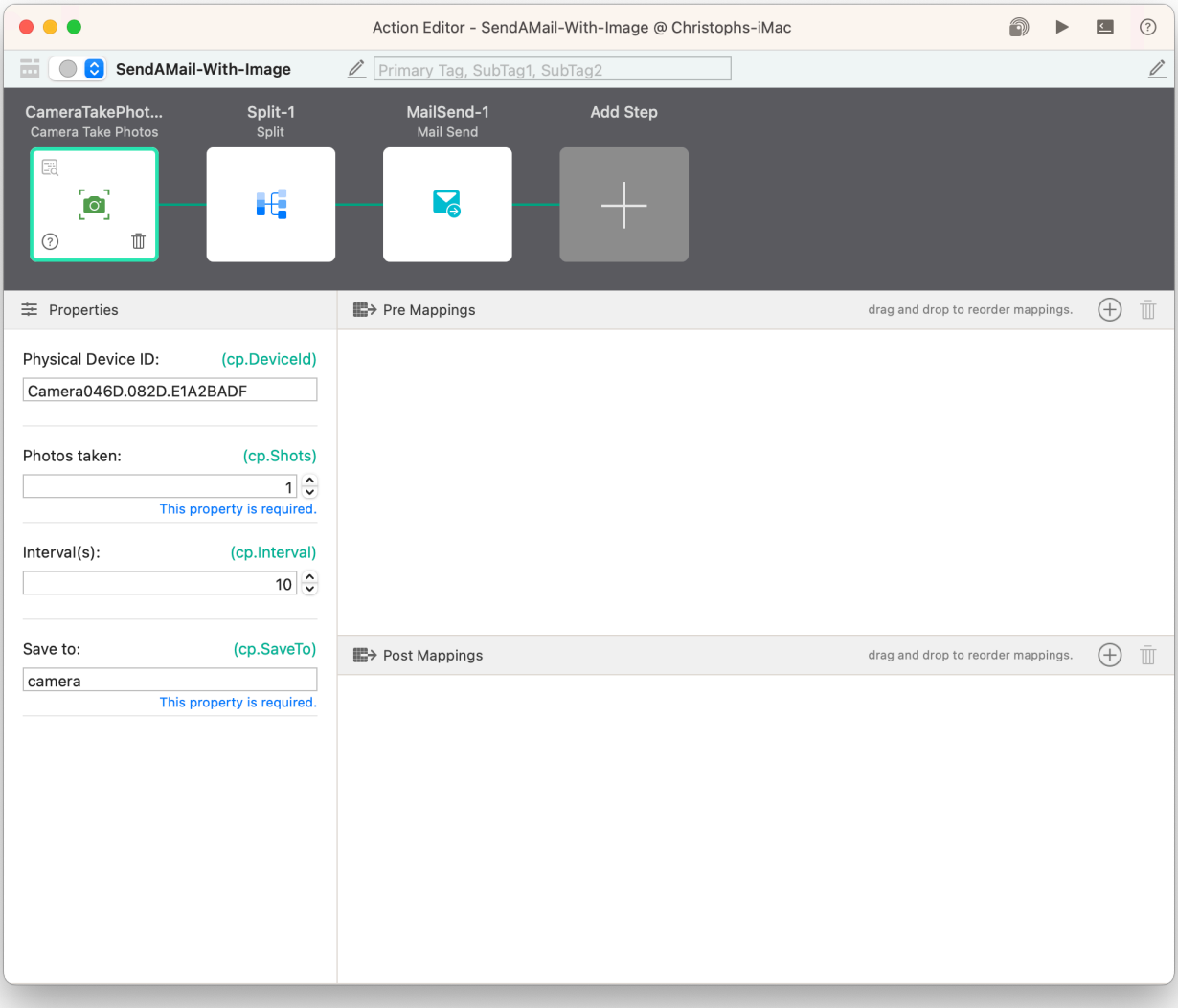
component properties

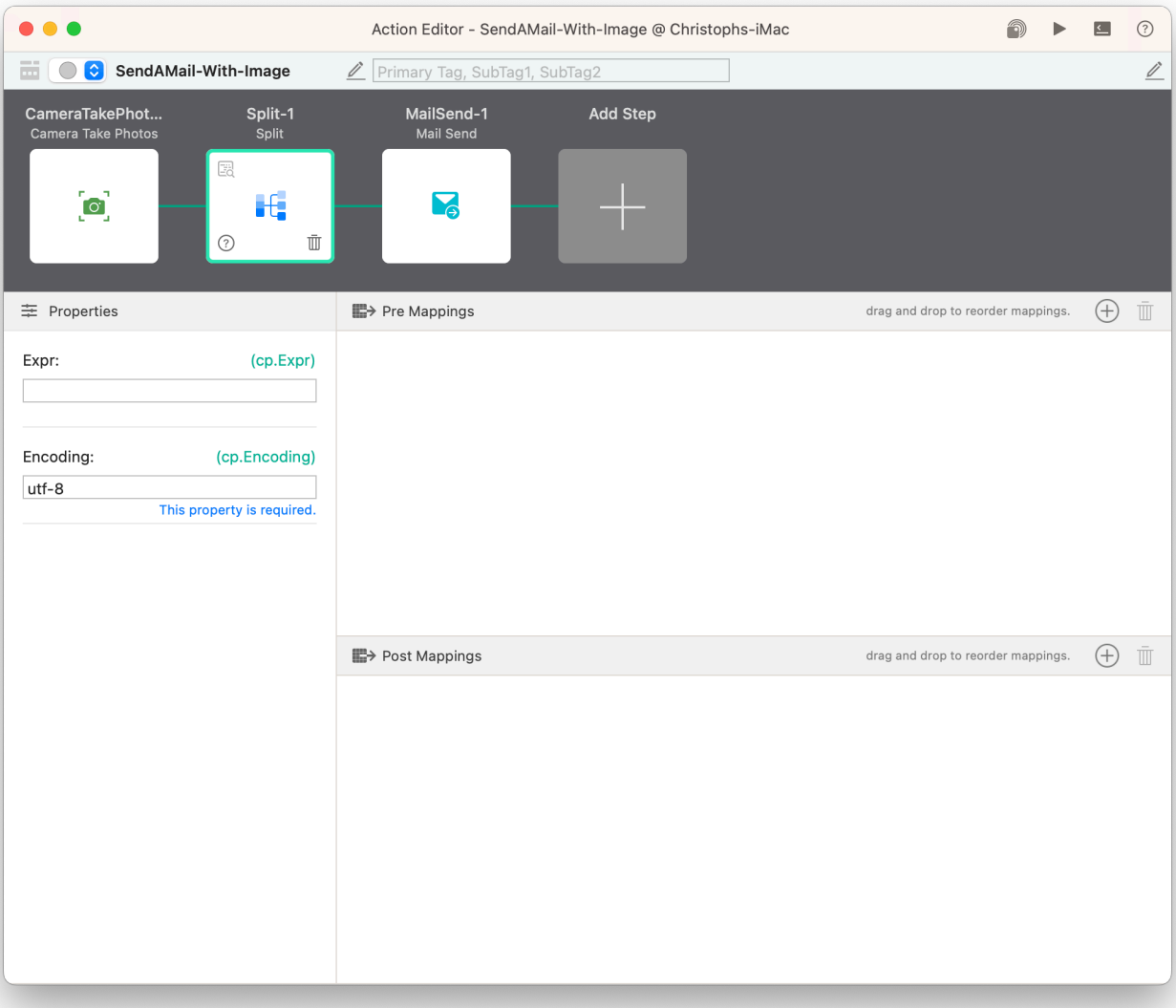
property name	Description
Physical device ID	Specifies the physical device ID of the camera connected to the device list.
Number of	Specify the number of pictures to be saved when the component is executed (up to 5)

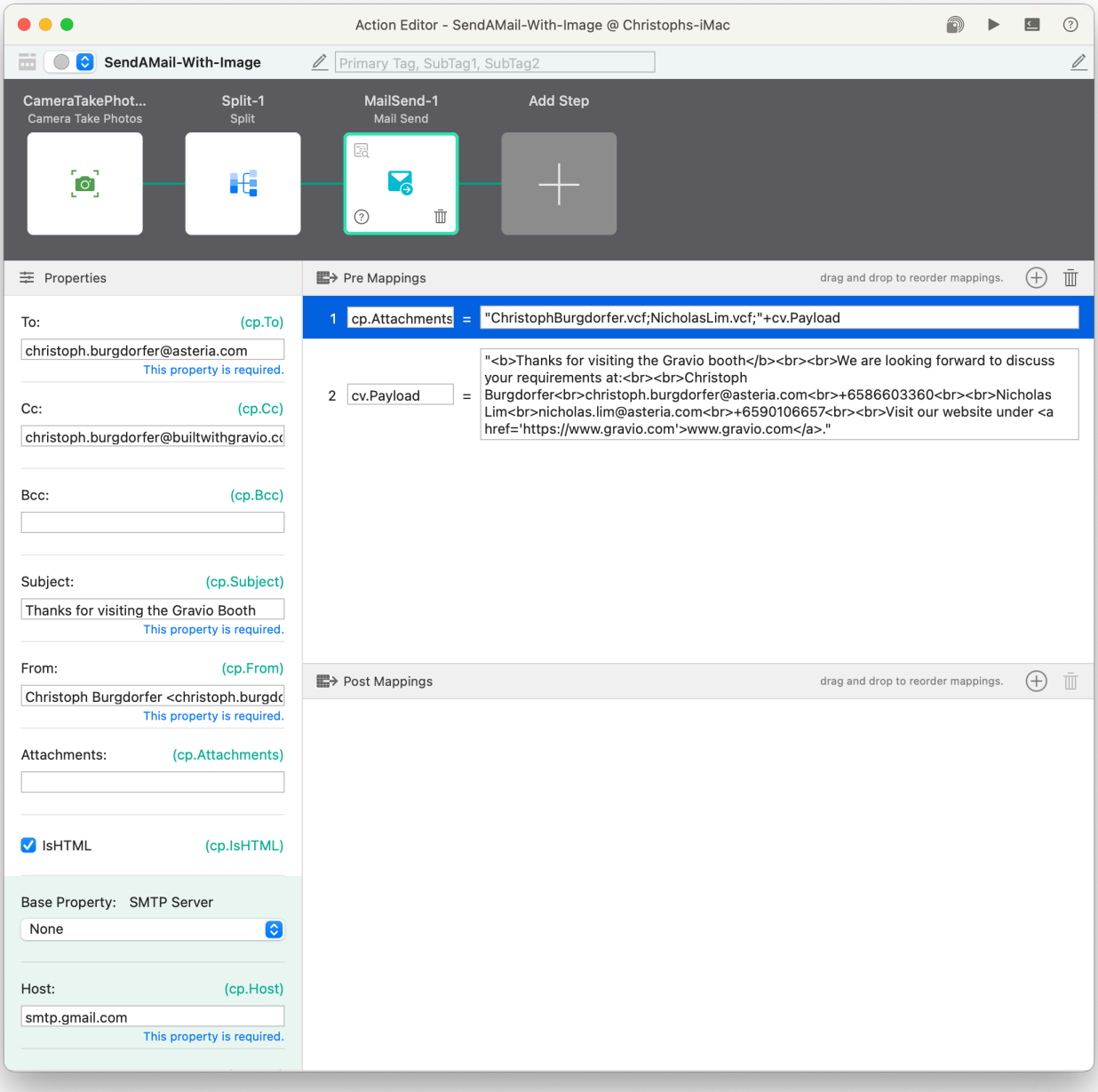
pictures	
Interval (seconds)	Specifies the interval to save photos
Destination	Specify a subfolder of actmgr/data as the file destination. The photos will be saved in this subfolder.

- The execution of the component requires a time specified by the number of pictures* interval, so please avoid settings that are too long. Example: If you set the number of photos to 5 and the interval to 6 seconds, the estimated execution time of the component will be 5*6=30 seconds.

Example of taking an image and sending it via e-mail




Action Editor - SendAMail-With-Image @ Christophs-iMac
SendAMail-With-Image
Primary Tag, SubTag1, SubTag2
CameraTakePhot...
Camera Take Photos
Split-1
Split
MailSend-1
Mail Send
Add Step
Properties
Expr:
(cp.Expr)
Encoding:
(cp.Encoding)
utf-8
This property is required.
Pre Mappings
drag and drop to reorder mappings.
Post Mappings
drag and drop to reorder mappings.


Action Editor - SendAMail-With-Image @ Christophs-iMac
SendAMail-With-Image
Primary Tag, SubTag1, SubTag2
CameraTakePhot...
Camera Take Photos
Split-1
Split
MailSend-1
Mail Send
Add Step
Properties
To: (cp.To)
christoph.burgdorfer@asteria.com
This property is required.
Cc: (cp.Cc)
christoph.burgdorfer@builtwithgravio.co
Bcc: (cp.Bcc)
Subject: (cp.Subject)
Thanks for visiting the Gravio Booth
This property is required.
From: (cp.From)
Christoph Burgdorfer <christoph.burgdo
This property is required.
Attachments: (cp.Attachments)
IsHTML (cp.IsHTML)
Base Property: SMTP Server
None
Host: (cp.Host)
smtp.gmail.com
This property is required.
Pre Mappings
drag and drop to reorder mappings.
1 cp.Attachments = "ChristophBurgdorfer.vcf;NicholasLim.vcf;"+cv.Payload
2 cv.Payload = "Thanks for visiting the Gravio booth

We are looking forward to discuss your requirements at:

Christoph Burgdorfer
christoph.burgdorfer@asteria.com
+6586603360

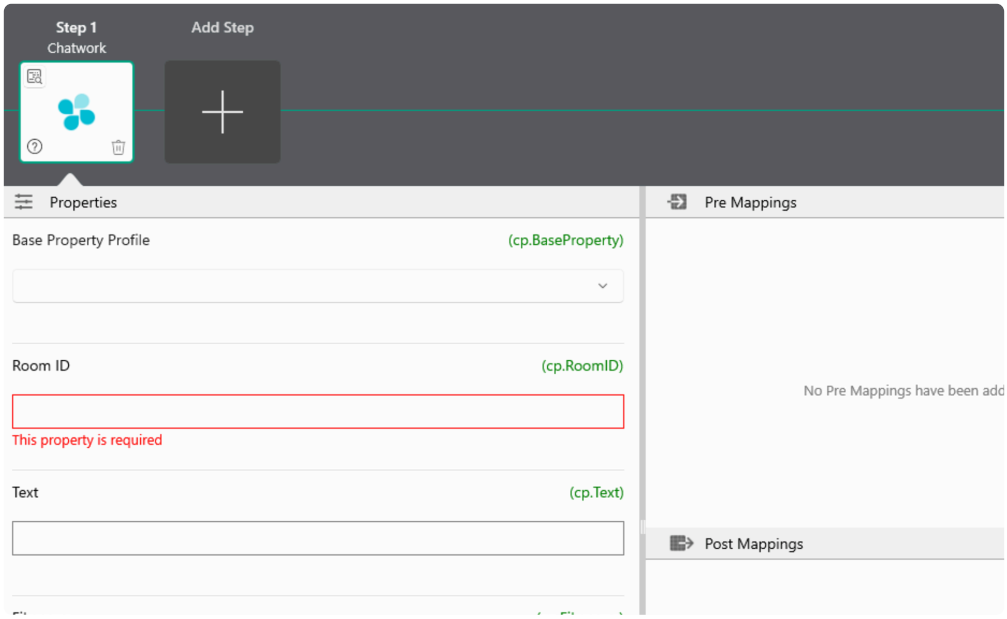
Nicholas Lim
nicholas.lim@asteria.com
+6590106657

Visit our website under www.gravio.com."
Post Mappings
drag and drop to reorder mappings.

6.3.9.6. Chatwork

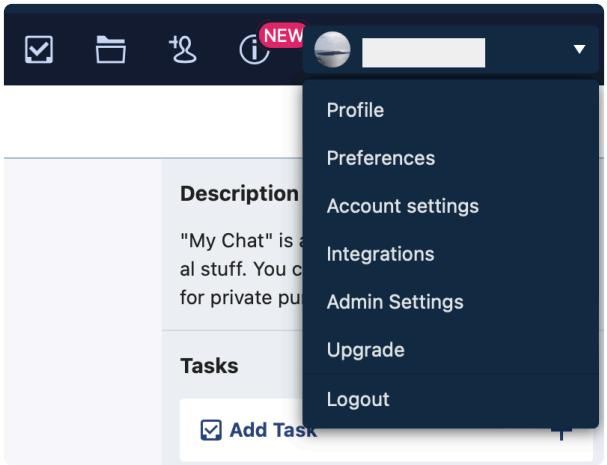
The Chatwork component can post messages specified in the component properties using Chatwork. This is handy for notifying individuals about information from sensors.

- Please note that in order to use the Chatwork component, the administrator must initially configure it in the service integration and obtain an API token.

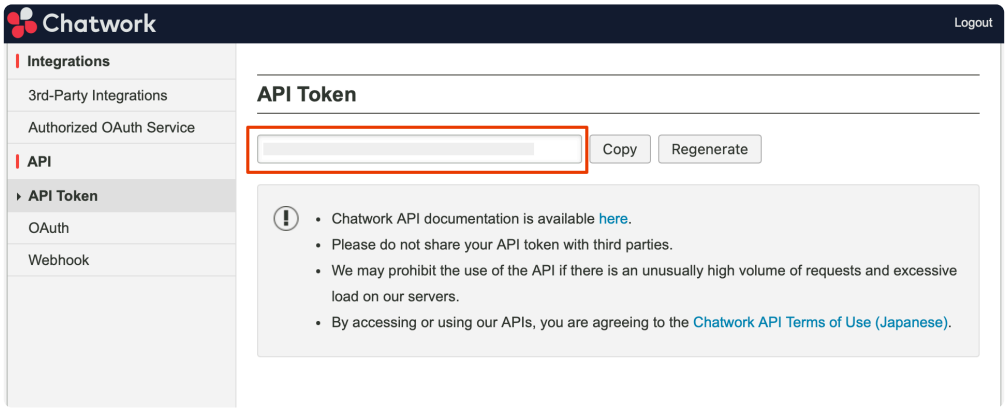


Initial Setup

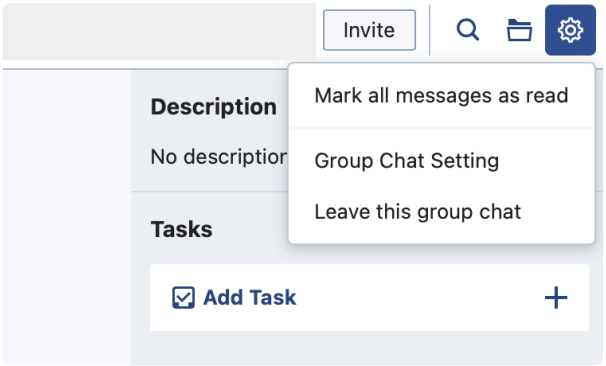
You can obtain the API token from the service integration screen of Chatwork.



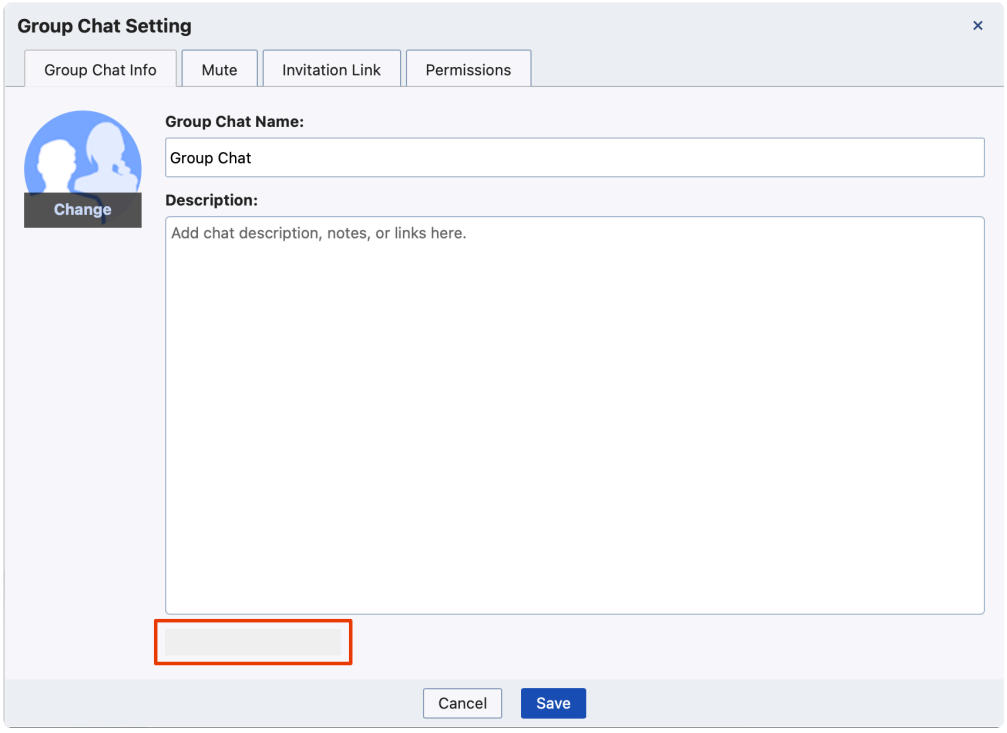
Select service integration from the account menu.



Obtain the value of the API token in the API menu. This will be used in the component property.



The Chatwork component posts in group chat, so open that group chat and choose group chat settings from the settings menu.



Obtain the room ID displayed on the group chat settings screen. This will be used in the component property.

With this, the initial setup is complete.

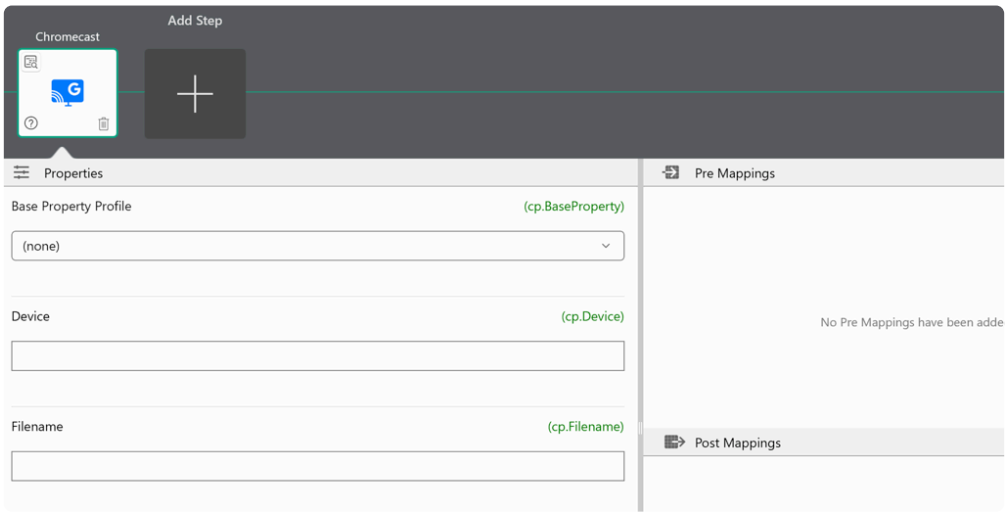
Component Properties

Property Name	Description	Example
Basic Properties	Specify the basic properties of the 'Chatwork Token' category	
Room ID	The Room ID of the group to post (obtained during initial setup)	
Text	The message string to post on Chatwork	Example) `"The current temperature is "+cv.Payload+" degrees"`
File Name	The file to send	
Timeout	API timeout	
Token	API Token (obtained during initial setup)	

- If both text and file are specified, both text and file will be sent.
- If the file path is not specified, the file reference will be actmgr/data, but for the notation of the file name specifying the file path, please refer to here.

6.3.9.7. Chromecast

The Chromecast component is a component that plays the video and audio specified by the Filename property on the Chromecast device.



Component Properties

Property Name	Description
Device	IP address of the Chromecast device
Filename	File name of the file to be read (supported file mp4/webm and mp3/ogg/wav formats)

※The reference for files that do not specify a file path is actmgr/data, see here for how to write a file name that specifies a file path.

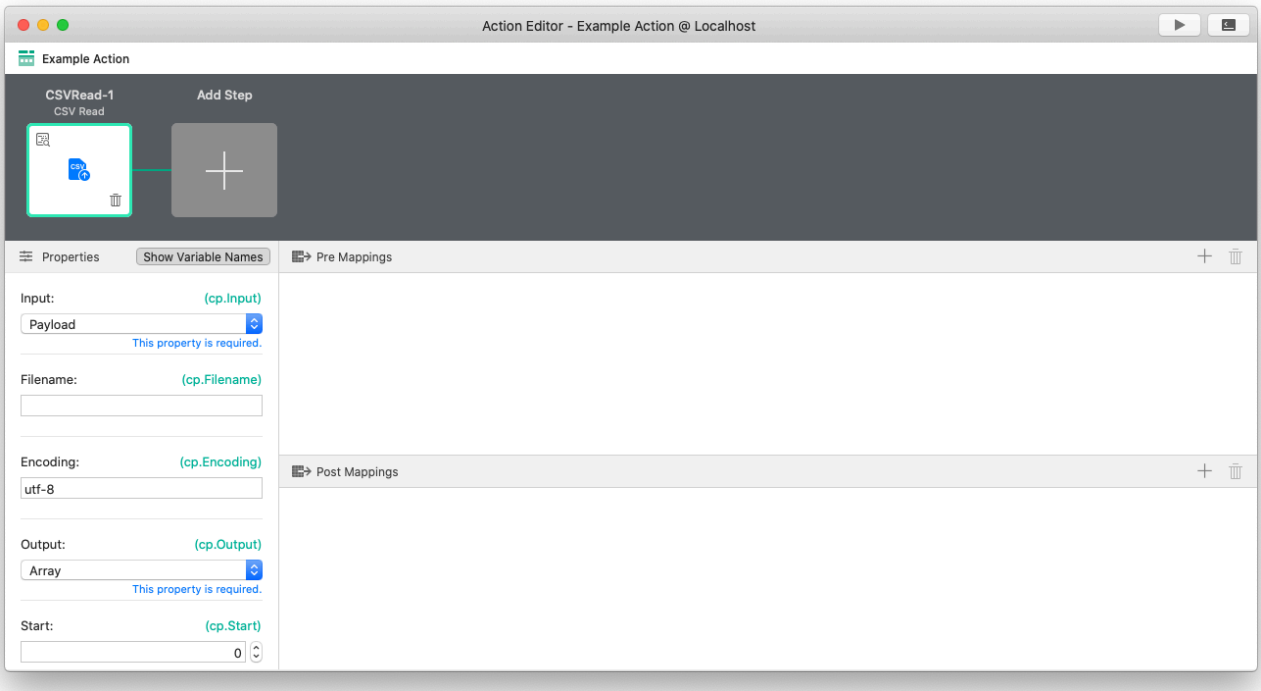
6.3.9.8. CSVRead

The CSV Read component takes CSV data as input, parses it and creates an output payload in the format specified by the Output Format component property (`cp.Output`).

CSVRead has an “Input” property (`cp.Input`), which allows you to select the data source. If you select [Payload], the input payload (`cv.Payload`) from the previous component is used as the source data of CSV, and if you select [File], the file specified by the [File name] property (`cv.Filename`) is used as the source data of CSV.

If the input payload (`cv.Payload`) is a byte sequence or a string, it becomes the source data of CSV as it is, but for other types, it is converted to a string and processed as the source data of CSV.

If “Array” is selected for the “Output Format” component property (`cp.Output`), an array of strings of CSV fields is put into the output payload (`cv.Payload`) for each line of the CSV and passed to the next component.
If you select “Object” in the “Output Format” component property (`cp.Output`), the first line of the CSV is considered as the key, and a JSON Object is generated, and the object is put into the output payload for each function and passed to the next component.



Component Properties

Name	Required	Description
Input	True	Specify either payload or file for input. To set the value of cp.Input, use either “Payload” or “File”.

Filename	False	Specify the filename for file input
Encoding	False	Specify the encoding of the file output (utf-8, shift_jis, euc-jp,iso-2022-jp)
Output	True	Specify either array or object as the output format. cp.Output value can set “Array” or “Object”.
Start	False	Specify the start line to read as CSV data
Length	False	Specify how many lines to read. If -1 is specified, read all lines.

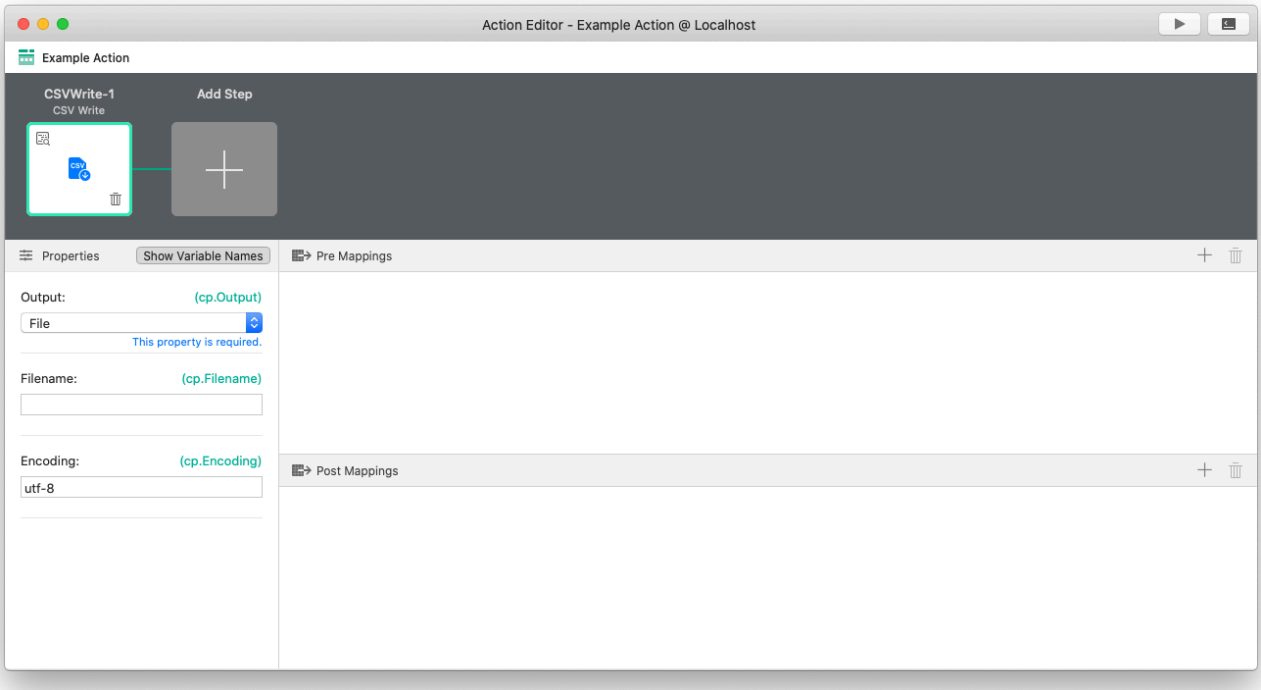
6.3.9.9. CSVWrite

CSV Write generates a comma-separated string with the input payload as input. It is not meant to create CSV files for further processing as writing the output to a file will not append, but overwrite it. If you want to populate a CSV file appending new lines of data after every incoming sensor data, you can use the File Write component and follow the example there.
The input payload can be an array of JSON arrays or an array of JSON objects. Otherwise, it will be converted to a string and output as a CSV with one line and one column.

Input Payload	Example
JSON array payload example	[[123, “abc”, “xxx”], [456, “def”, “yyy”], [789, “hij”, “zzz”]]
JSON Object payload example	[{field1, “field2”:“abc”, “field3”:“xxx”}, {field1, “field2”:“def”, “field3”:“yyy”}, {field1, “field2”:“hij”, “field3”:“zzz”}]

The “Output” property allows you to select the output destination as file or payload.

Output payload	Description
When you choose “File” for the “Output Destination” property	The generated CSV is written to a file and the output payload passes the input payload. (Pass-through.)
When you choose “Payload” for the “Output to” property	Put the generated CSV byte sequence into the output payload.



Component Properties

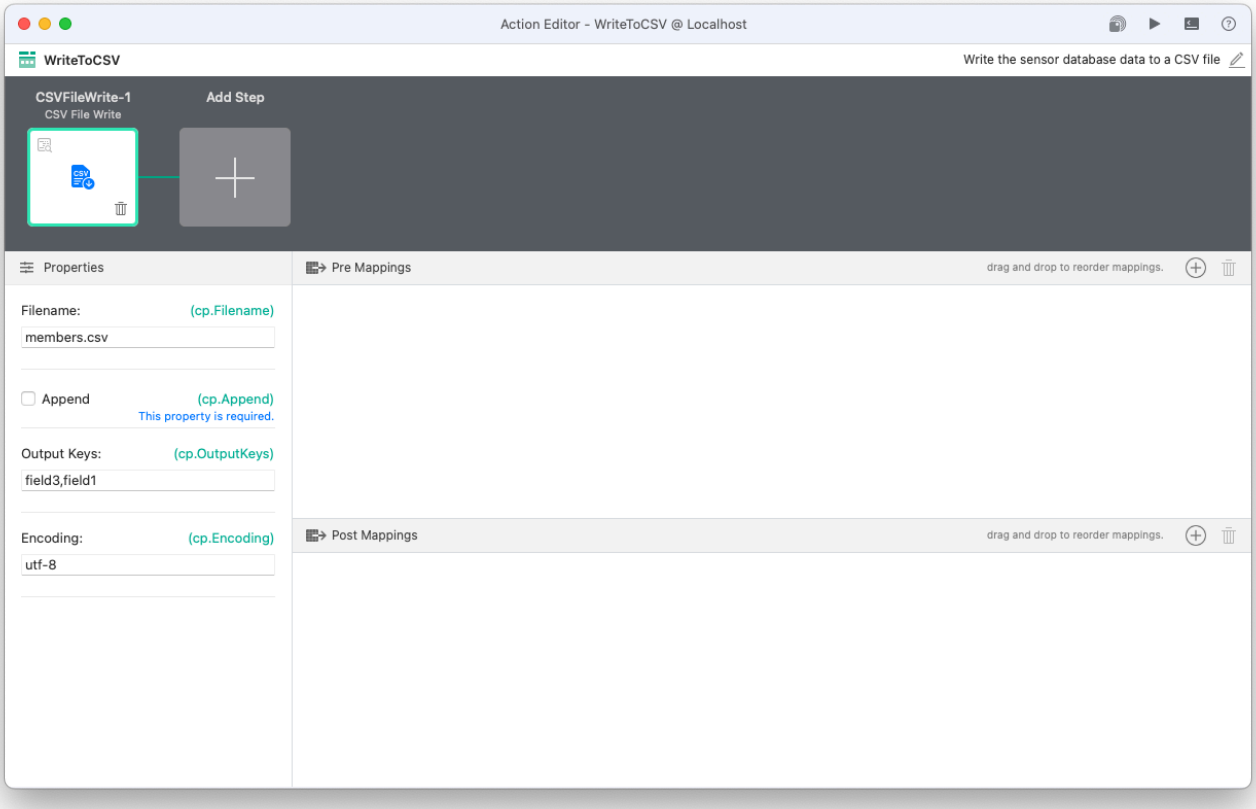
Name	Required	Description
Output	True	Specify either File or Payload as the output destination. cp.Output is set to either “Payload” or “File”.
Filename	False	Specify the file name for file output
Encoding	False	Specify the encoding of the file output (utf-8, shift_jis, euc-jp,iso-2022-jp)

6.3.9.10. CSV File Write

CSVFileWrite generates CSV with the input payload as input.

The input payload (`cv.Payload`) accepts input as an array of JSON arrays or an array of JSON Objects. Otherwise, it converts it to a string and outputs it as a one-line, one-column CSV file.

Input payload	Example
Example of JSON array payload	`[[123, "foo", "bar"], [456, "jane", "doe"], [789, "john", "doe"]]`
Example of JSON Object payload	`[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, " field2": "hij", "field3": "zzz"}]`



component properties

Property name	Description
File Name	Specify the file name in case of file output

Append	If true, appends data to the file if it already exists; if false, overwrites the file
Output Keys	If the input payload is a JSON Object, list the field names to be output in the order you want them to be output, separated by commas.
Encoding	Specify the encoding of the file output.

If the file path is not specified, the file is saved in `actmgr/data`. For how to refer to a file path, refer to this part.

How to use the Output Keys

The output keys specify the order, in which the data will be written to the CSV if you use keys.

For example, when the payload of JSON Object is as follows

```
[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3":" zzz"}]
```

and you specify `field3, field1` as the output field order, the CSV output will be:

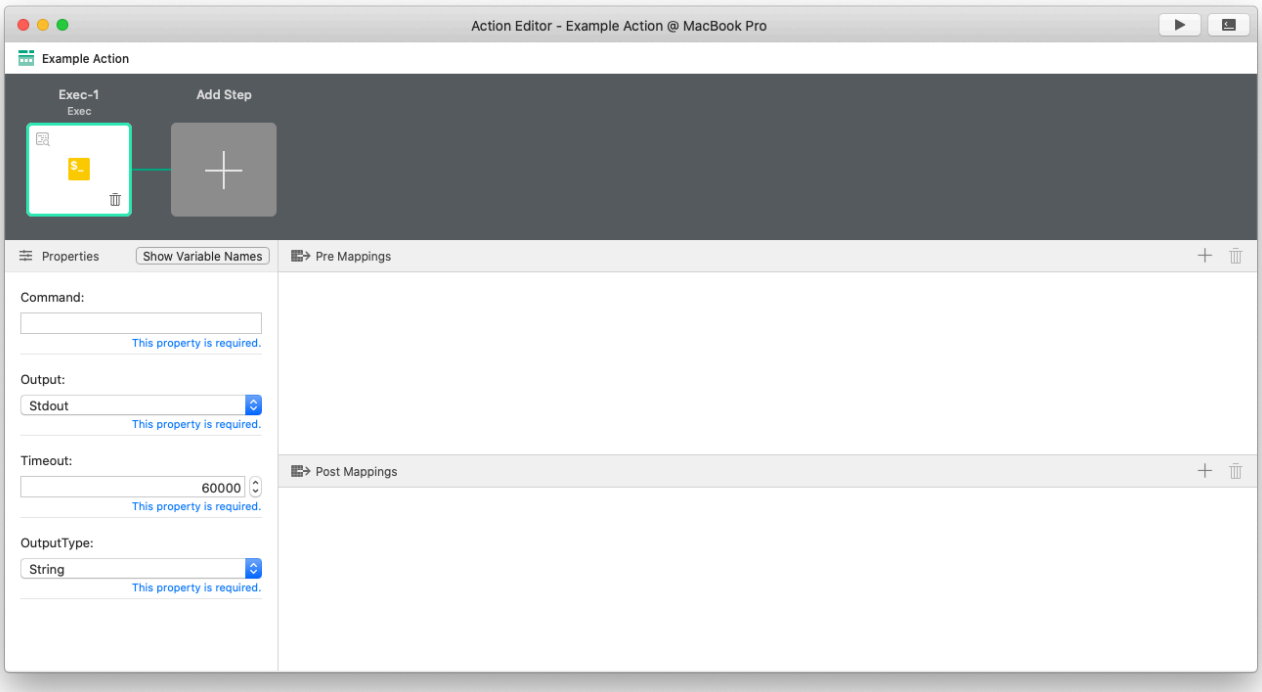
```
xxx,123
yyyy,456
zzz,789
```

6.3.9.11. Exec

The RunScript component allows you to run OS standard scripts.
If Gravio Server is running on Windows it can run batch files, Linux, Raspbian and macOS can run shell scripts.
If it is not Windows, you need to give execute rights to the uploaded file. The uploaded file is placed in `<Gravio data folder>/actmgr/data/`, so

For the Gravio Folder in your operating system, consult the documentation about Path Handling

You can also pass variable values to the scripts, for example, to pass sensor values.



Parameters

Command	Required	The command you like to execute
Output	Required	The output of the command. cp.Output is set to either “stdout” or “combine”.
Timeout	Required	Timeout
OutputType	Required	Set the output type. cp.OutputType can be either “string” or “binary”.

The output of the component is passed on to the `cv.Payload`

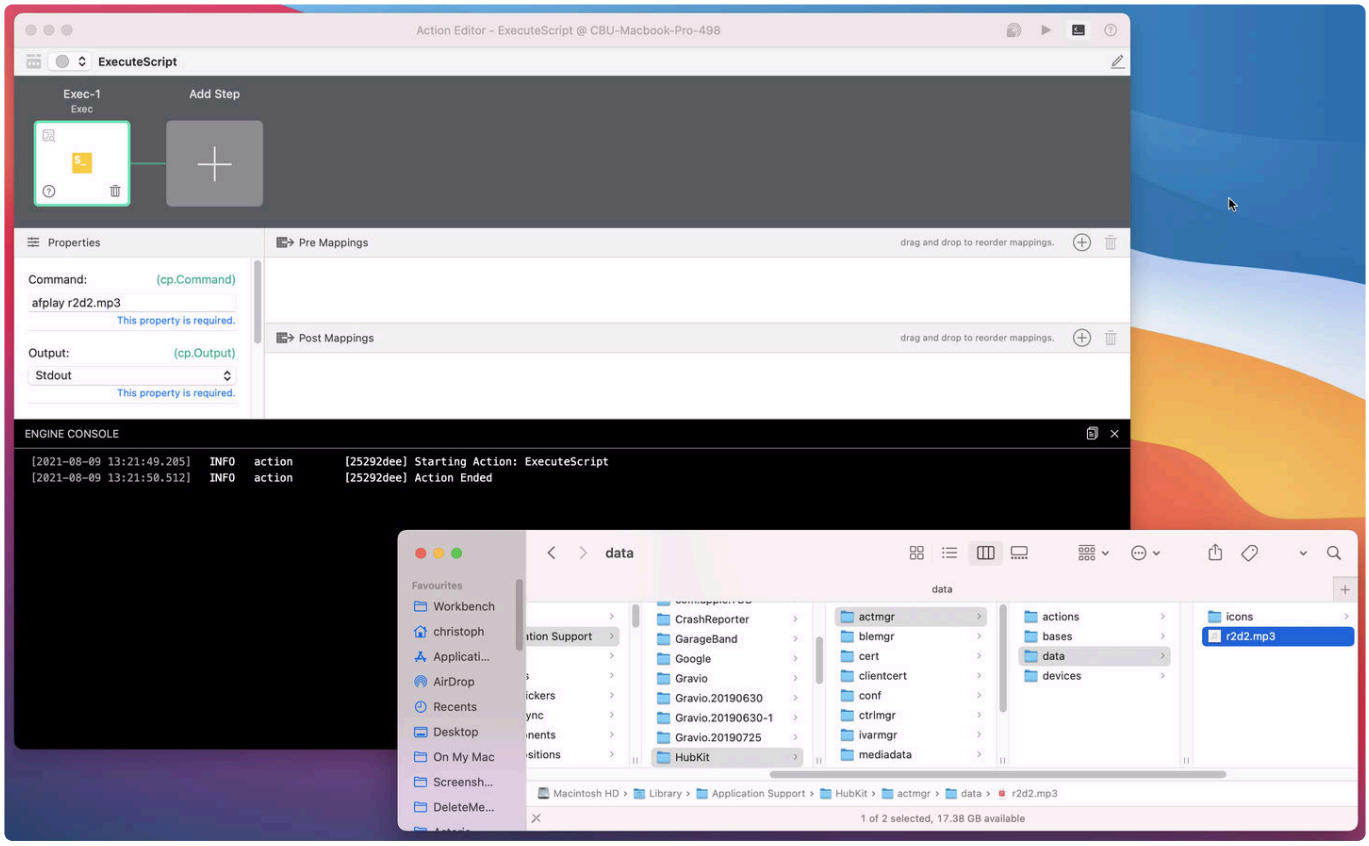
Macintosh Users

Macintosh users must give Gravio HubKit 4 full harddisk access in the security settings if they want to

access files outside the Gravio folder. Please find the appropriate settings in your system preferences under the Privacy and Security settings:

Example: Playing an MP3 File

Place an `mp3` file in your `/actmgr/data` folder and use the command `afplay` (on mac) followed by the filename to play it:



Linux

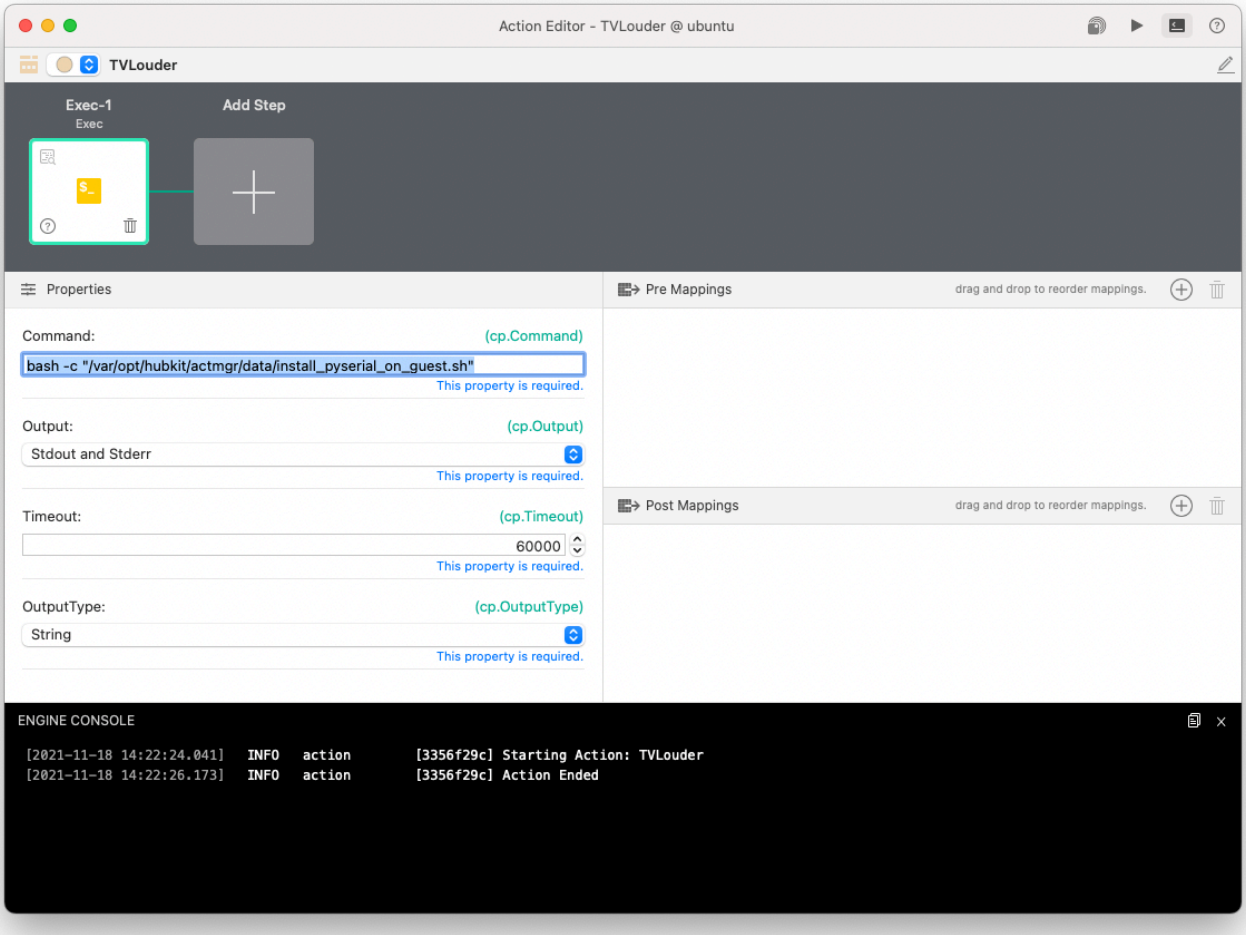
Note, in Linux Gravio is running in a Docker container. This means that you may not be able to access the same environment as the shell script. This also means you will need to use different paths. You can see the relevant paths here. To recap: the folder `/home/gravio/hubkitrepo4/data/actmgr/data/` on the host system is mounted on `/var/opt/hubkit/actmgr/data/` on the guest system (Gravio).

If you run a Python script that relies on libraries, you may have to install those libraries first. In our example, we want to trigger a Python script that talks to the serial port. This requires the `serial` package. In order to install that, you must run this shell script called `install_pyserial_on_guest.sh` within the Gravio environment first:

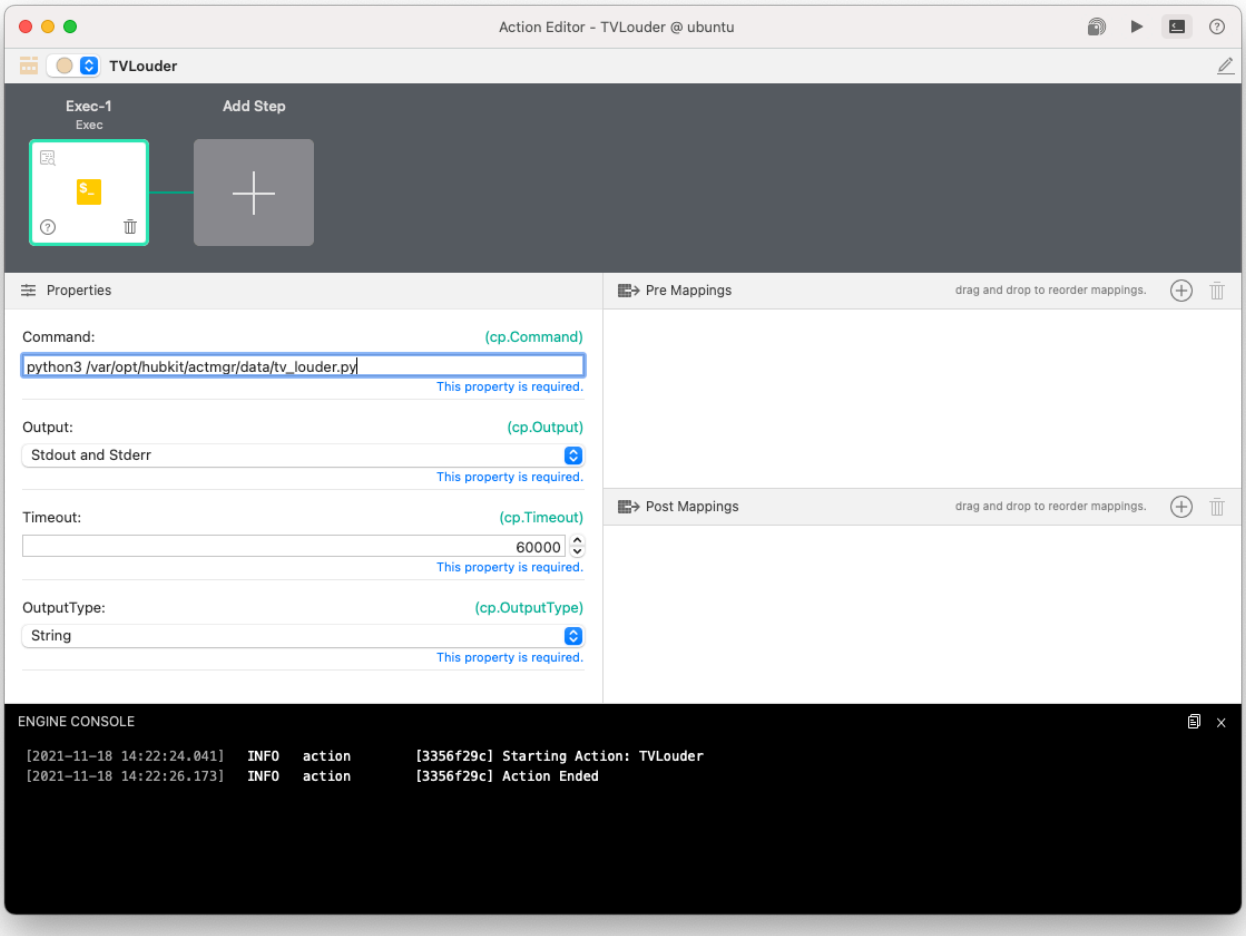
```
#/bin/bash

# install pip3
if ! command -v pip3 &>/dev/null; then
        apt update
        apt install python3-pip
else
        echo "pip3 found"
fi

# install pyserial
python3 -c "import serial"
```

```
if [ $? -ne 0 ]; then
        pip3 install pyserial
else
        echo "pyserial found"
fi
```

You need to run this within your Gravio Exec component prefixed by `bash -c`, i.e. `bash -c "/var/opt/hubkit/actmgr/data/install_pyserial_on_guest.sh"`



That may take a little while, but after that, you can trigger your python scripts that rely on those libraries:



Using Named Pipes

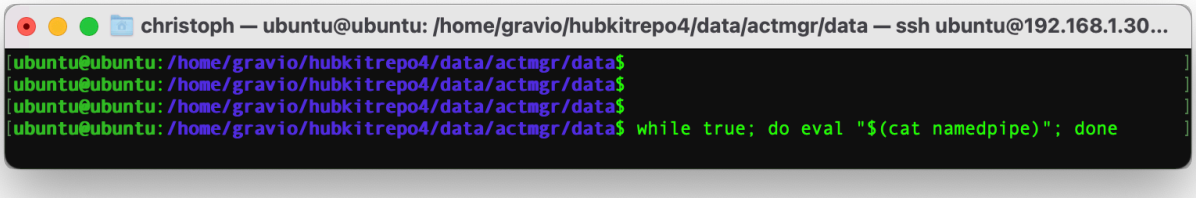
You can also use Named Pipes (FIFOs) that are mounted on both, the docker as well as the host system, to pass trigger commands on the host system from the docker system. You can then use `eval` on the host system to receive the commands coming from Gravio through the named pipe.

Step 1: Create a named pipe “file” (FIFO) by using the `mkfifo` command somewhere in your mounted folder. For example, on the host system, enter:

```
sudo mkfifo /home/gravio/hubkitrepo4/data/actmgr/data/namedpipe
```

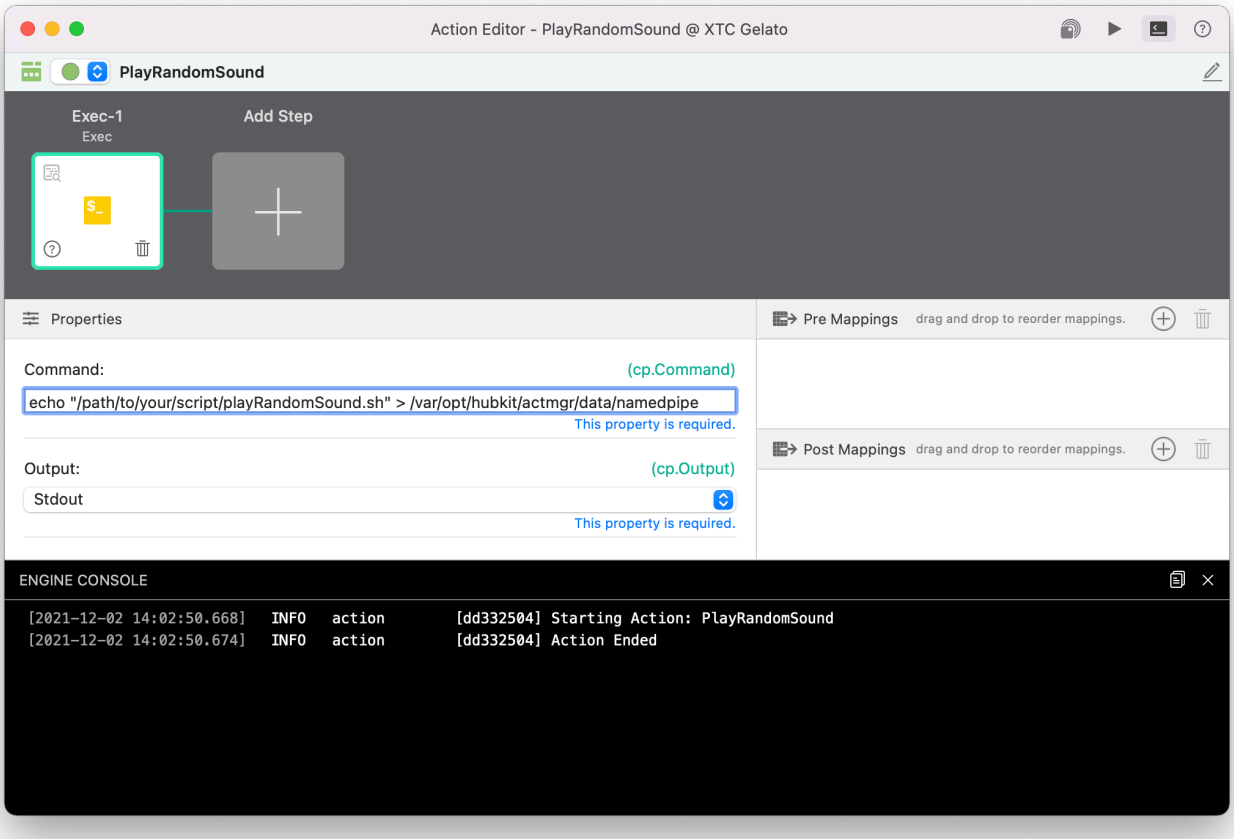
Step 2: Listen on the host system for the command to be executed. Move to the folder you just created the `namedpipe` file and write:

```
while true; do eval "$(cat namedpipe)"; done
```



This will wait for anything that comes through the named pipe and execute it. Please bear in mind the necessary security restrictions for that.

Step 3: Simply `echo` the command you want to execute on the host system to the Named Pipe from Gravio.



Of course, you would want the “listener” to be executed automatically every time the system boots. For this, you can create a script named `execpipe.sh` somewhere on your system:

```
#!/bin/bash
while true; do eval "$(cat /home/gravio/hubkitrepo4/data/actmgr/data/namedpipe)"; done
```

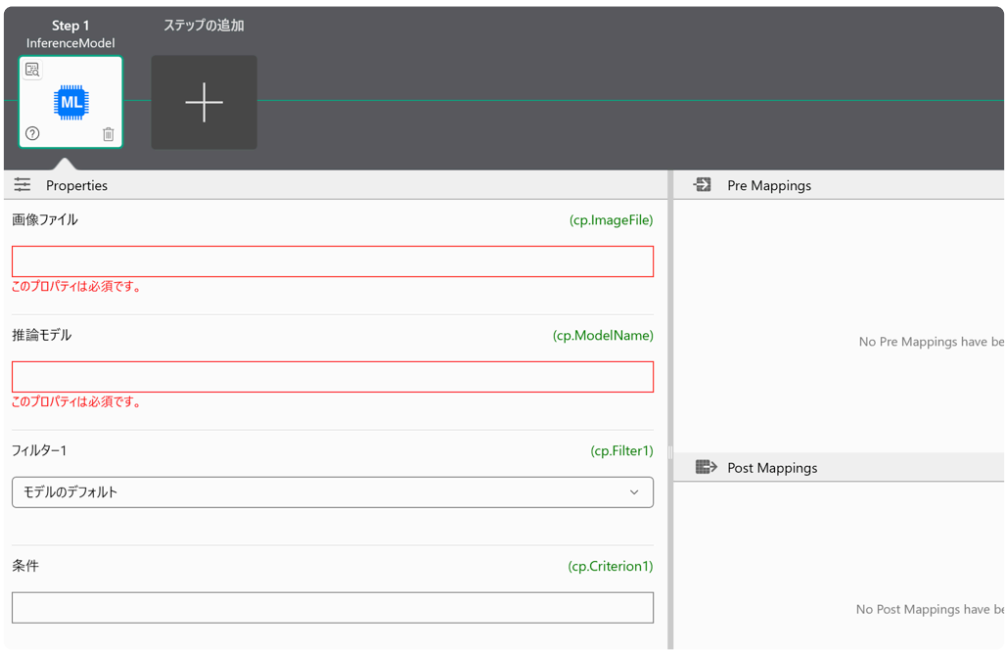
and execute it every time the system boots by adding the following command to your `crontab` by entering `crontab -e` and adding this line:

```
@reboot /path/to/execpipe.sh
```

Important: be aware of any security considerations. More information about this solution can be found on https://stackoverflow.com/a/63719458/354204

6.3.9.12. Execute Inference Model

This component allows you to specify an inference model to be performed on an image file.



Output payload	Description
`cv.Payload`	The result of the inference run is output

Component Property

property name	Description
image file	Specifies the file path of a picture saved as a file
Inference model	Specify the model name of the inference to be performed
Filter1	Select to filter inference results
filter1	select if you want to filter inference results
filter2	select if you want to filter inference results
Condition	Specify the condition for filter 2
Conversion	Specify a conversion
Draw	Specify if you want to save the inference result as an image
Copy Destination	Specify where to save the image if “Copy” is specified in “Draw”

Filter

This is used when you want to extract only the results that match the condition from the inference results.

h3. Filter	Description
None	Do not use filter
Default of the model	Use the filter defined in the model
highest n predictions	retrieve the n data specified by the condition from the higher of multiple predictions
the lowest n predictions	retrieve the n data specified by the condition from the lowest of the multiple predictions
predictions scored higher than	retrieve the n data specified by the condition from the lowest of the multiple predictions
predictions cored lower than	retrieve data from multiple forecasts whose score is less than x specified by the condition
prediction labeled with	retrieve data from multiple predictions where the label contains a string specified by the condition

Conversion

Use this to convert the result of inference.

Conversion	Description
None	Do not use transformation
Default of the model	Use the transformation defined in the model
number	get the number of inference results
first label	get the first label of the inference result
all labels	retrieve all labels in the inference result

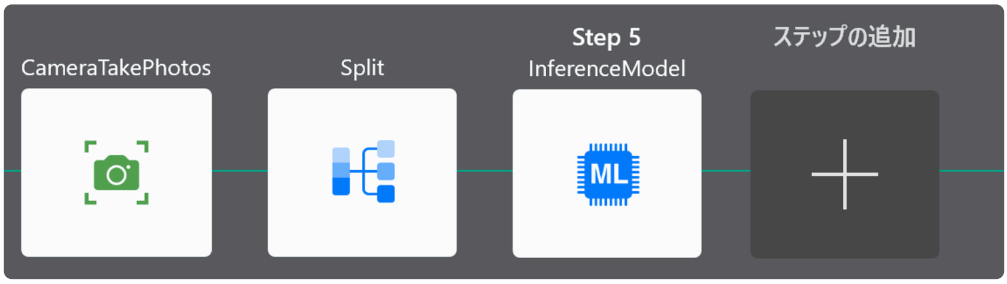
Draw

Use to save the inference result as an image.

Draw	Description
None	Do not save the result of inference as an image
Original	Save the result of inference as an original image
Copy	Create another file in the destination of the copy and save the result of the inference

Use in combination with CameraTakesPhotos component

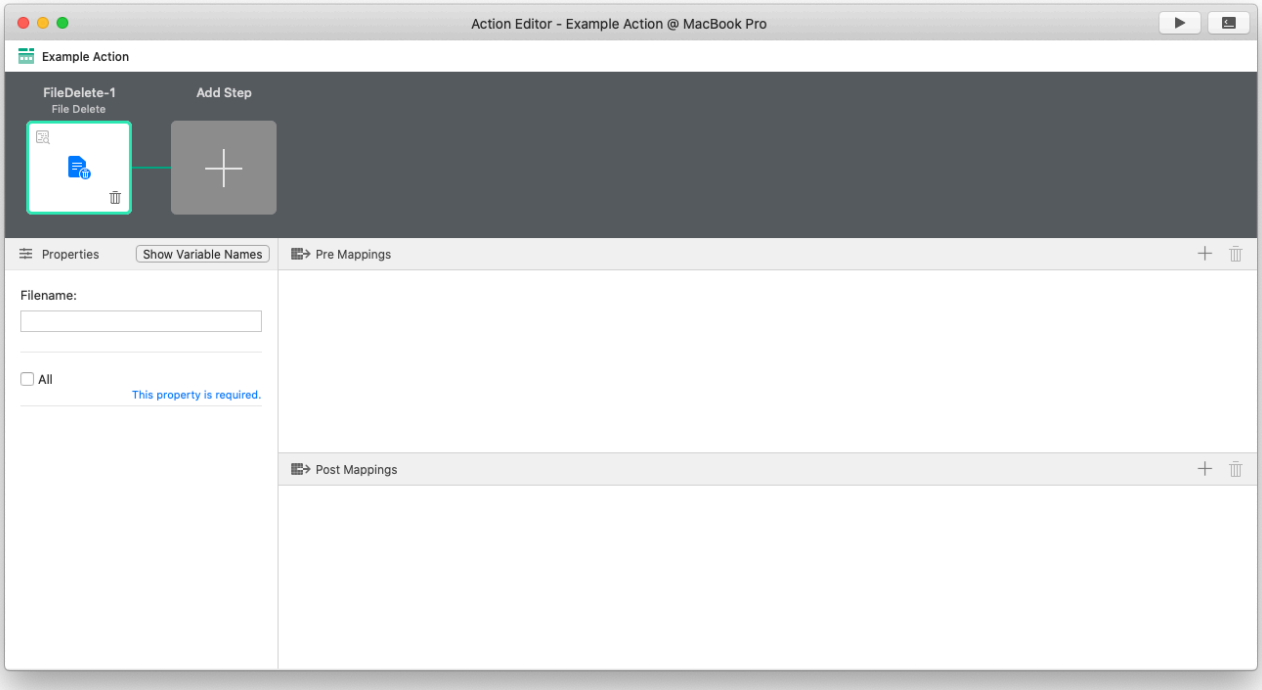
You can use the CameraTakesPhotos component when executing an action to perform inference on a photo saved as a file.



Place a Split component between the CameraTakesPhotos component and the ExecuteInferenceModel component as shown in this example.
Since the output of the CameraTakesPhotos component is an array of filenames of the saved photos, the Split component splits the array into
individual items to be processed each in the subsequent ExecuteInferenceModel component.

6.3.9.13. File Delete

The File Delete Action Component is to delete a file.



Component Properties

Name	Required	Description
Filename	False	Filename
All	True	All. The value of cp.All can be either “true” or “false

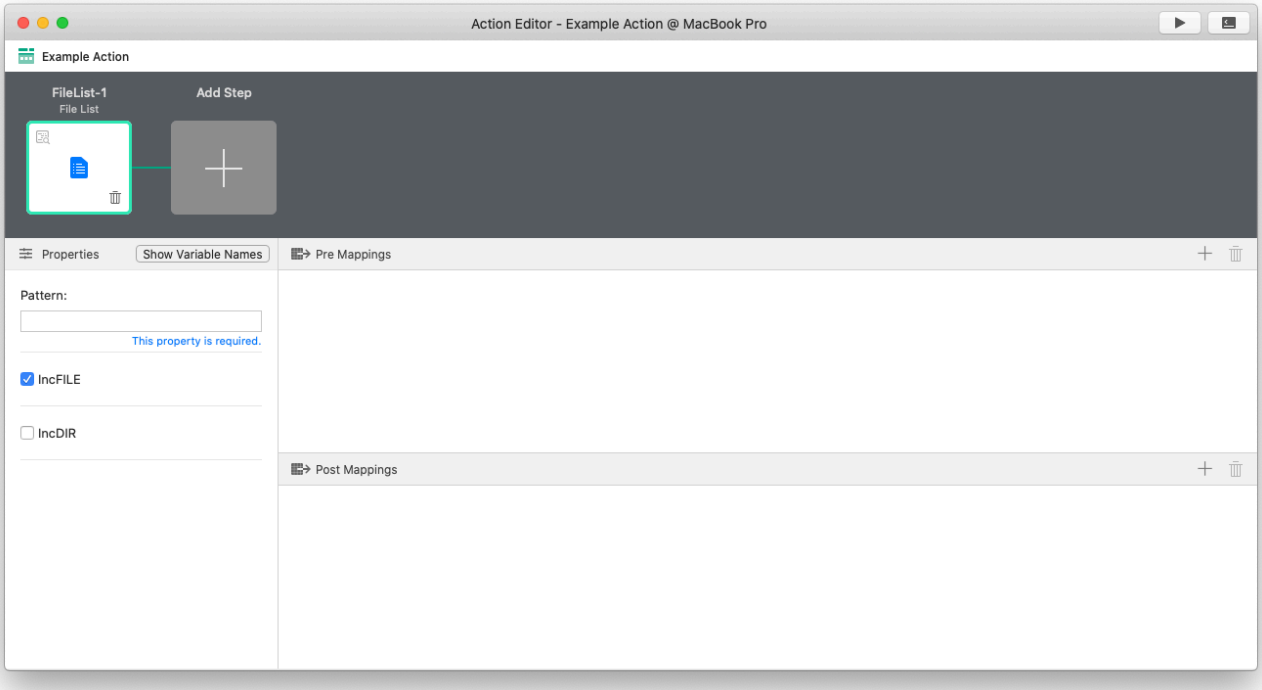
The file has to be inside these folders:

On Windows: C:\ProgramData\HubKit\action\actmgr\data\
On Mac OS: /Library/Application Support/HubKit/action/scripts/actmgr/data/
On Linux: /var/opt/hubkit/action/scripts/actmgr\data\

any path information will be relative from these folders.

6.3.9.14. File List

The File List Action Component is to get a list of files that match the pattern.



Output payload	Description
`cv.Payload`	Outputs the contents of the read directory as individual JSON objects.

The contents of the directory will have the following data:

For example the pattern `"icons/*"` might return something like this for each file:

```
{"IsDir":false,"ModTime":"2021-08-10T13:18:29.2577374+09:00","Name":"arrow_down.bmp","Path":"icons/arrow_down.bmp","" Size ":824}
```

Data	Description
IsDir	`true` for directories, `false` for files
ModTime	Modification Date and Time
Name	File name
Path	File path
Size	File size

Component Properties

Name	Description
Pattern	The conditional expression to list files, e.g. `icons/*`. Note these folders are relative to `actmgr/data`
IncFILE	Specify whether to include files or not
IncDIR	Specify whether to include directories or not

The pattern without a file path refers to the `actmgr/data` folder, so all paths need to be relative to that folder. You can find more details about the paths on different operating systems here.

Note: The payload will contain the metadata for one file returned by the list each. If you follow the File List component with the Join component, you'll get a payload with every file metadata joined together. In this case, you will get:

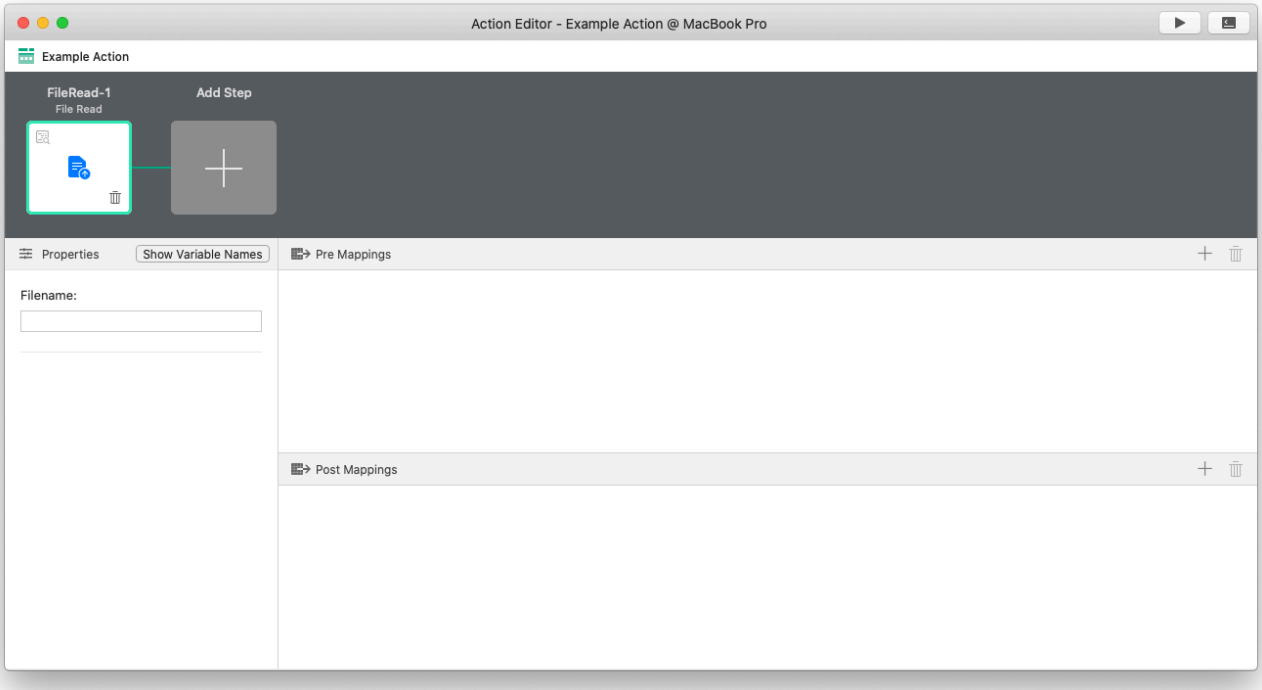
```
[
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"arrow_down.bmp",
      "Path":"icons/arrow_down.bmp",
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"arrow_left.bmp",
      "Path":"icons/arrow_left.bmp",
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"arrow_right.bmp",
      "Path":"icons/arrow_right.bmp",
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"arrow_up.bmp",
      "Path":"icons/arrow_up.bmp",
      "Size":824
   },
```

```
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"circle.bmp",
       "Path":"icons/circle.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"correct.bmp",
       "Path":"icons/correct.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"downloading_f1.bmp",
       "Path":"icons/downloading_f1.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"downloading_f2.bmp",
       "Path":"icons/downloading_f2.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"downloading_f3.bmp",
       "Path":"icons/downloading_f3.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"error.bmp",
       "Path":"icons/error.bmp",
       "Size":824
    },
    {
       "IsDir":false,
       "ModTime":"2017-12-14T17:23:11+08:00",
       "Name":"sad.bmp",
       "Path":"icons/sad.bmp",
```

```
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"smile.bmp",
      "Path":"icons/smile.bmp",
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"zigbee_off.bmp",
      "Path":"icons/zigbee_off.bmp",
      "Size":824
   },
   {
      "IsDir":false,
      "ModTime":"2017-12-14T17:23:11+08:00",
      "Name":"zigbee_on.bmp",
      "Path":"icons/zigbee_on.bmp",
      "Size":824
   }
]
```

6.3.9.15. File Read

The File Read Action Component is to read a file.



Component Properties

Name	Required	Description
Filename	False	Filename of the file to be read.

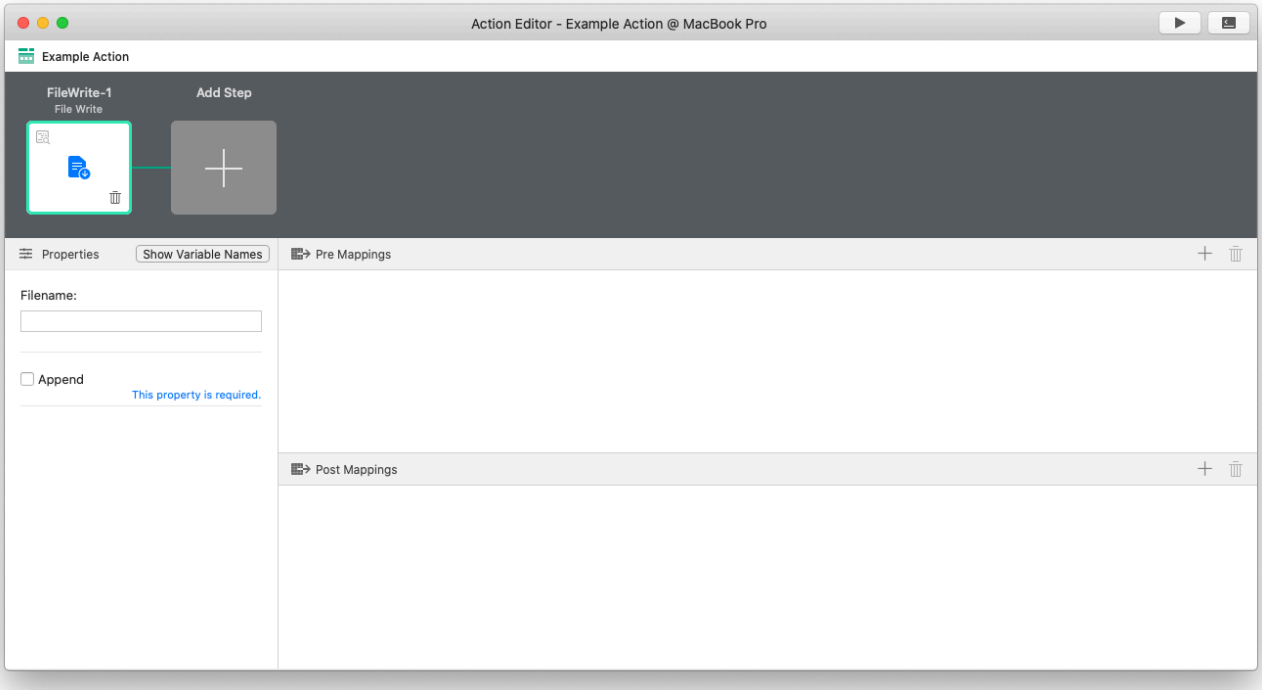
The file has to be inside these folders:

On Windows: C:\ProgramData\HubKit\action\actmgr\data\
On Mac OS: /Library/Application Support/HubKit/action/scripts/actmgr/data/
On Linux: /var/opt/hubkit/action/scripts/actmgr\data\

any path information will be relative from these folders.

6.3.9.16. File Write

The File Write Action Component is to write data to a file.



Component Properties

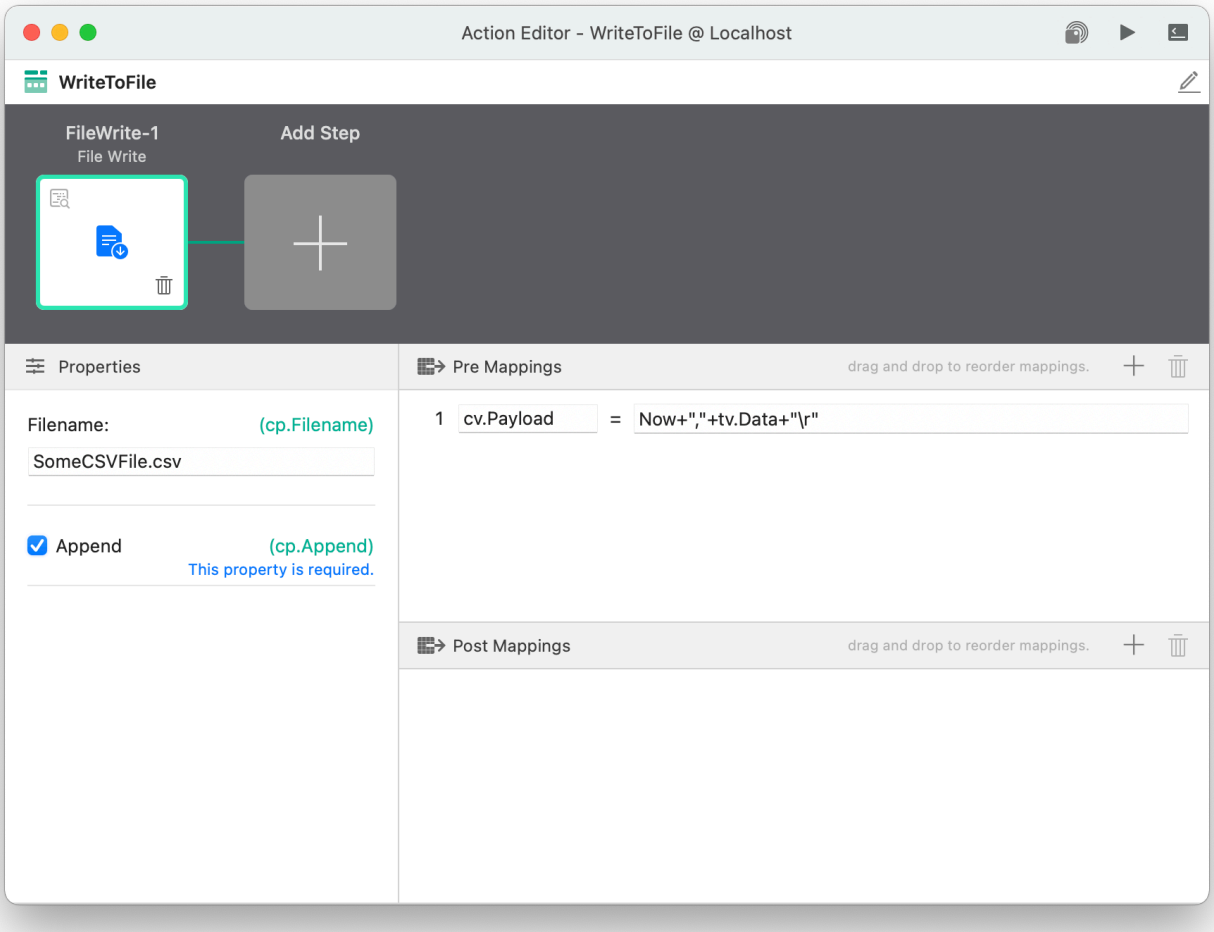
Name	Required	Description
Filename	False	Filename to be written.
Append	True	Append new data instead of replacing the file.

The file has to be inside these folders:

On Windows: C:\ProgramData\HubKit\action\actmgr\data\
On Mac OS: /Library/Application Support/HubKit/action/scripts/actmgr/data/
On Linux: /var/opt/hubkit/action/scripts/actmgr\data\

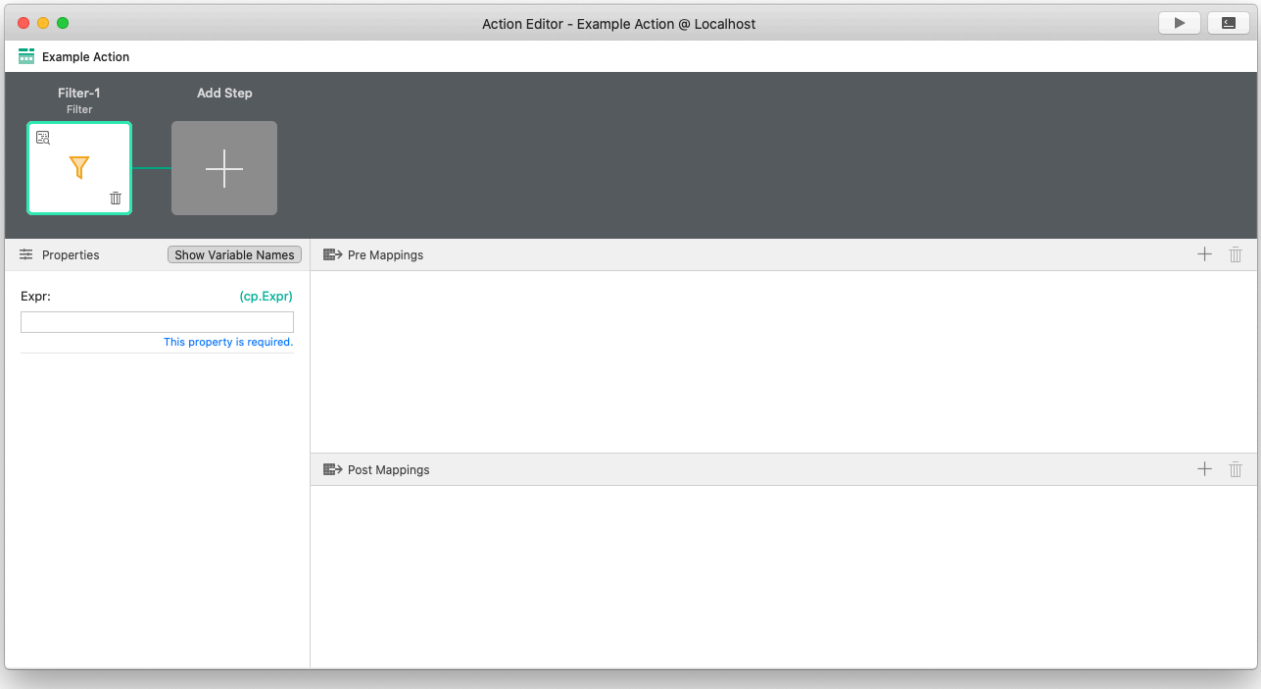
any path information will be relative from these folders.

You can, for example, use the File Write component to create a CSV file that contains sensor data alongside with timestamps. In this case, your configuration looks something like this:


Action Editor - WriteToFile @ Localhost
WriteToFile
FileWrite-1
File Write
Add Step
Properties
Filename:
(cp.Filename)
SomeCSVFile.csv
Append
(cp.Append)
This property is required.
Pre Mappings
drag and drop to reorder mappings.
1
cv.Payload
=
Now+","+tv.Data+"\r"
Post Mappings
drag and drop to reorder mappings.

6.3.9.17. Filter

The Filter component is a component that filters the data that passes the contents of the input payload to the next component and outputs it to the output payload.



You can enter a conditional expression that allows you to pass a test if it's true, or not.
The conditional expression is written with variables with `tp.` / `tv.` / `ap.` / `av.` / `cp.` / `cv.` prefixes as comparison targets.

Condition Example	Formula
Payloads are targeted and Payloads are numerical in nature	`cv.Payload > 10`
String	`cv.Payload =~ "^abc"`
JSON Array	`cv.Payload[1] > 10`
JSON Object	`cv.Payload.key1 == "abc"`

The expression field supports basic operators such as ^ to indicate the start of a string.

Component Properties

Name	Description
Condition	Write a conditional expression

Available Modifiers

Modifiers	`+ - / * & | ^ ** % >> <<`
Comparators	`> >= < <= == != =~ !~`
Logical Ops	`|| &&`

Examples

- If you like the action to stop unless the JSON value `book_id` of the first book in the `books` array in `tv.Data` payload is set to `12345`, you can use a Comparator, for example `tv.Data.books[0].book_id == 12345`

6.3.9.18. GCS Upload

The GCS Upload component can send data to Google Cloud Storage as files.

To use this component, a Google Cloud subscription and pre-configuration are required.

Roles for accessing Google Cloud Storage must be added “Storage Legacy Bucket Owner” and “Storage Legacy Object Owner”.

Please refer to here for pre-configuration.

component properties

property name	Description
Input	Choose whether to upload the contents of the payload or specify a file name and upload it
File name	When a file name is selected, specify the file to be uploaded as a path relative to the Data directory
Bucket	Specify the name of the bucket to upload to.
object	Specify the name of the object to be uploaded. When specifying a folder, separate it with a slash. If the property is empty and the upload source is a file, only the filename part without the directory name is used for the file name.
ACL	Specify an access control list *1
Storage Class	Specify the storage class *2
ContentType	Specify the ContentType of the object. If empty, it is automatically appended.
Timeout	API timeout
Private key JSON	Specify the private key JSON file required to authenticate the service account created by Google Cloud.

*1 ACLs can be achieved by giving multiple users in a list format, such as which user (IAM or other user) is given Owner/Reader rights.

This format cannot be configured in Gravio Studio, so use predefined ACLs.

If you select (default) for the ACL, no ACL is specified and the bucket default is used.
Also, buckets are not allowed to specify ACLs by default because they are not recommended.
If you want to use ACLs in a bucket, you must select “fine-grained management” on the screen that allows you to choose between “uniform” and “fine-grained management” when creating the bucket.
If an ACL other than (default) is selected for a bucket for which “uniform” is selected, an error will occur when the component is executed.

*2 If you select (default), the bucket defaults will be used.

6.3.9.19. GCS Download

The GCS Download component can receive data as files from Google Cloud Storage.
This component requires a Google Cloud subscription and pre-configuration.

Roles for accessing Google Cloud Storage must be added “Storage Legacy Bucket Read” and “Storage Legacy Object Read”.
If you also use the GCS Upload component, please add “Storage Legacy Bucket Owner” and “Storage Legacy Object Owner”.

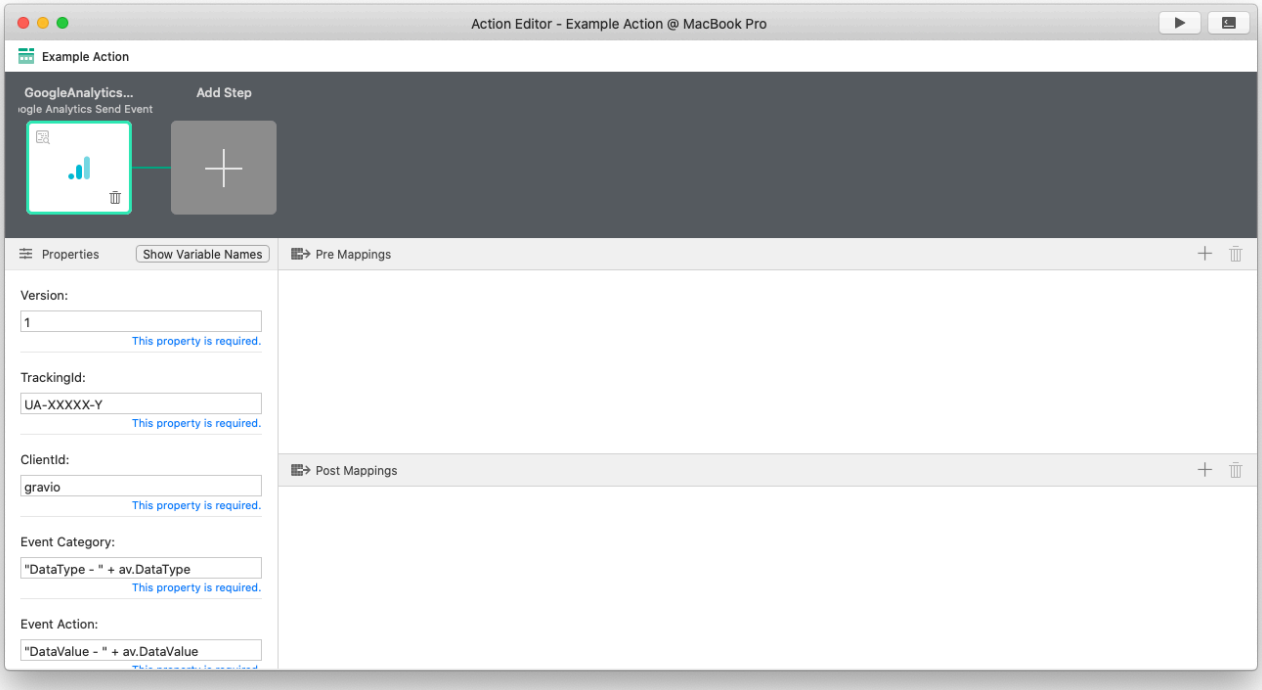
Please refer to here for pre-configuration.

component properties

property name	Description
Output	Specifies whether the downloaded object should be output to a payload or saved to a file
File Name	Specifies the file to download as a path relative to the Data directory when a file name is selected
bucket	Specify the name of the bucket to download
Object	Specify the object name to download
timeout	API timeout
Private key JSON	Specify the private key JSON file required to authenticate a service account created with Google Cloud

6.3.9.20. Google Analytics Send Event

The Google AnalyticsSendEvent component sends events using the Google Analytics Measurement Protocol.



For details of Measurement Protocol and parameters, please refer to this link.

Checking “Debug Mode” puts you in debug mode. In debug mode, if “Tracking ID” is incomplete, an error will be displayed, but it will not be counted as an event. Accordingly, if you are not in debug mode, you will not get an error even if the parameters are incomplete, so be careful.

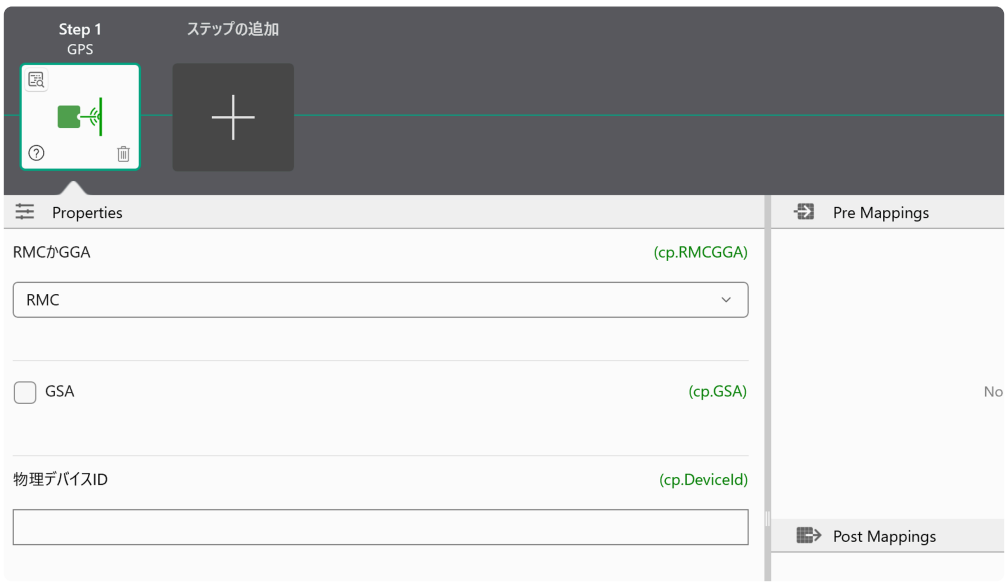
Component Properties

Name	Required	Description
Version	True	Google API Version
TrackingId	True	TrackingId
ClientId	True	ClientId
Event Category	True	Google Analytics Event Category
Event Action	True	Google Analytics Event Action
Event Label	True	Google Analytics Event Label
Event Value	True	Version
DebugMode	True	DebugMode

Timeout	True	Timeout (ms)

6.3.9.21. GPS

Receive and output GPS data from GPS devices.



Output Payload	Description
`cv.Payload`	The following data will be output

Data	Description
Course	Azimuth of the device's movement (number from 0 degrees north to 359.9 degrees clockwise) if the value is -1, the azimuth is invalid
SpeedKm	Device's movement speed (km/h)
SpeedKn	device's movement speed in knots (km/h)
LatitudeDeg	latitude (DEG format)
LongitudeDeg	Longitude (DEG format)
LatitudeDeg	Latitude (DEG format)
LongitudeDir	Longitude (DEGM format)
LatitudeDir	Latitude direction (N or S)
LongitudeDir	Azimuth in longitude (E or W)

Component Properties

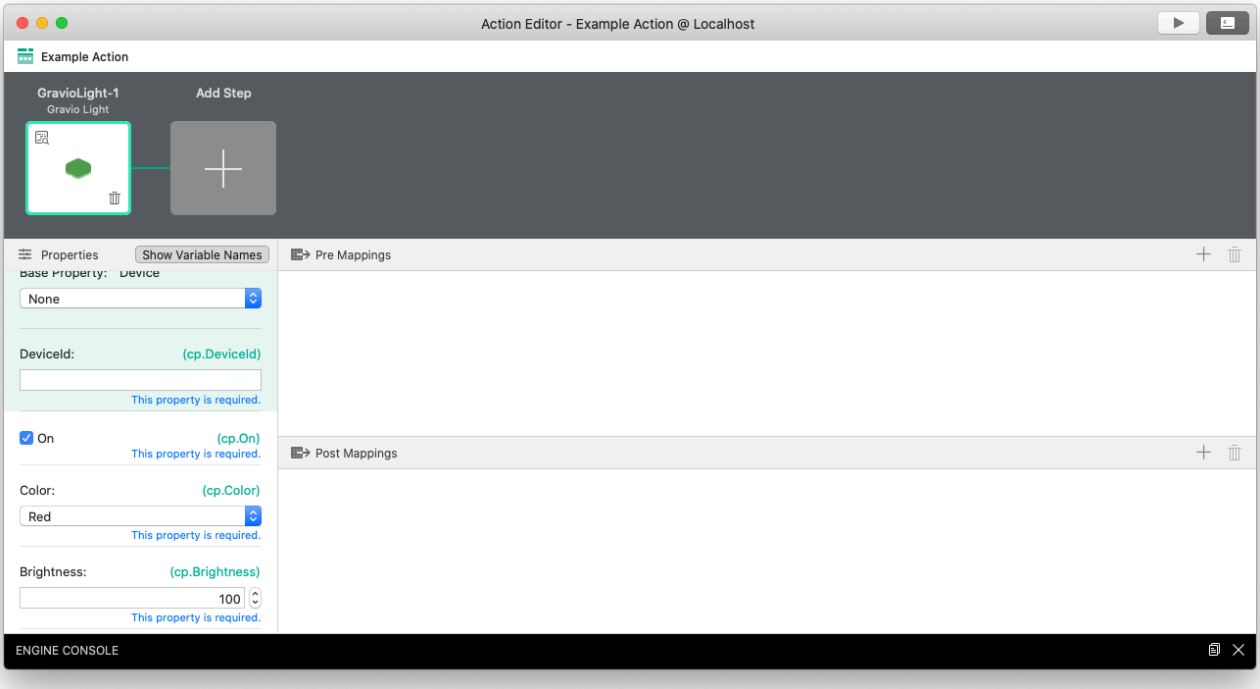
Property Name	Description
RMC or GGA	Selects whether to output data in RMC or GGA format
GSA	Specify whether to include GSA in the output data

Device ID	GPS device ID, If nothing is set, data output from all GPS devices is executed.

*When RMC is specified, use GPS devices that output RMC sentences.

6.3.9.22. GravioLight

The Gravio Light Action Component is to control a Gravio Lights. For this component to work, you must have the Gravio Lights paired in advance.



Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

Component Properties

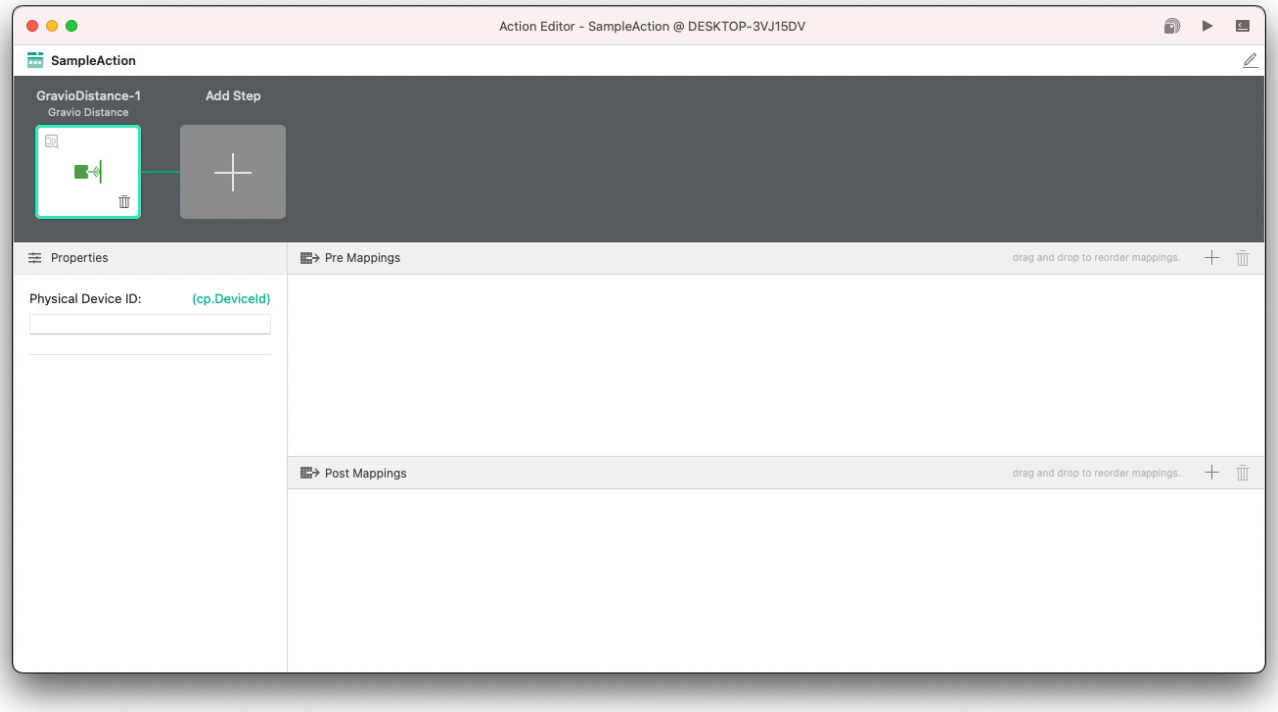
Name	Required	Description
Device	False	Devices as defined in the Base Property Profile
DeviceId	True	If not taken from the Base Property Profile, define here the device IDs to be turned on/off. If none are configured, all paired Gravio Lights will be affected. Multiple device IDs can be separated by commas. For example `32-40-00-FE-FF-57-B4-14,32-40-00-FE-FF-57-B4-15`
On	True	Specify if the lights should be on or off. The value specified for cp.On is “true” or “false”.
Color	True	Predefined Colors. cp.Color property’s value can be “red”, “yellow”, “green”, or “blue “ string.
Brightness	True	Brightness in %age

Note: If the light is not reacting initially, try setting it off and then on again using an action.

6.3.9.23. GravioDistance

The Gravio Distance component allows you to retrieve distance measurements from the Gravio Laser Distance sensors.
To use this component you must have paired the Gravio Laser Distance sensor beforehand.

If the Device ID property is empty, the command will be sent to all connected Gravio Distance.



Output payload	Description
`cv.Payload`	Outputs an array of measured distances

Component properties

Property name	Description
Device ID	Device ID for which the distance is to be measured, if none is set then the distance is measured for all Gravio Distances, if more than one Gravio Distance is used then the device ID can be comma-separated

Execution errors

Error code	Error	Description	Countermeasure

ERR—10	low power	insufficient current	check that the USB cable between the power supply and the sensor is correctly connected
ERR—14	calculation error	execution error	please run again
ERR—15	weak signal	Weak signal	Please do not move the device and remove any obstructions.
ERR—16	unstable laser transition	laser is unstable	hold device and target object still
ERR—18	ambient light too strong	ambient light is too strong	please reduce ambient light
ERR—26	out of display range	please try again	

About the error that occurs when bound to a layer

Note: When a Distance sensor that is bound to a device layer and enabled is specified and used with the Gravio Distance component, the following error will occur:

> ERROR 400 action: “GravioLaser”: failed to execute: component: GravioDistance: device is being used

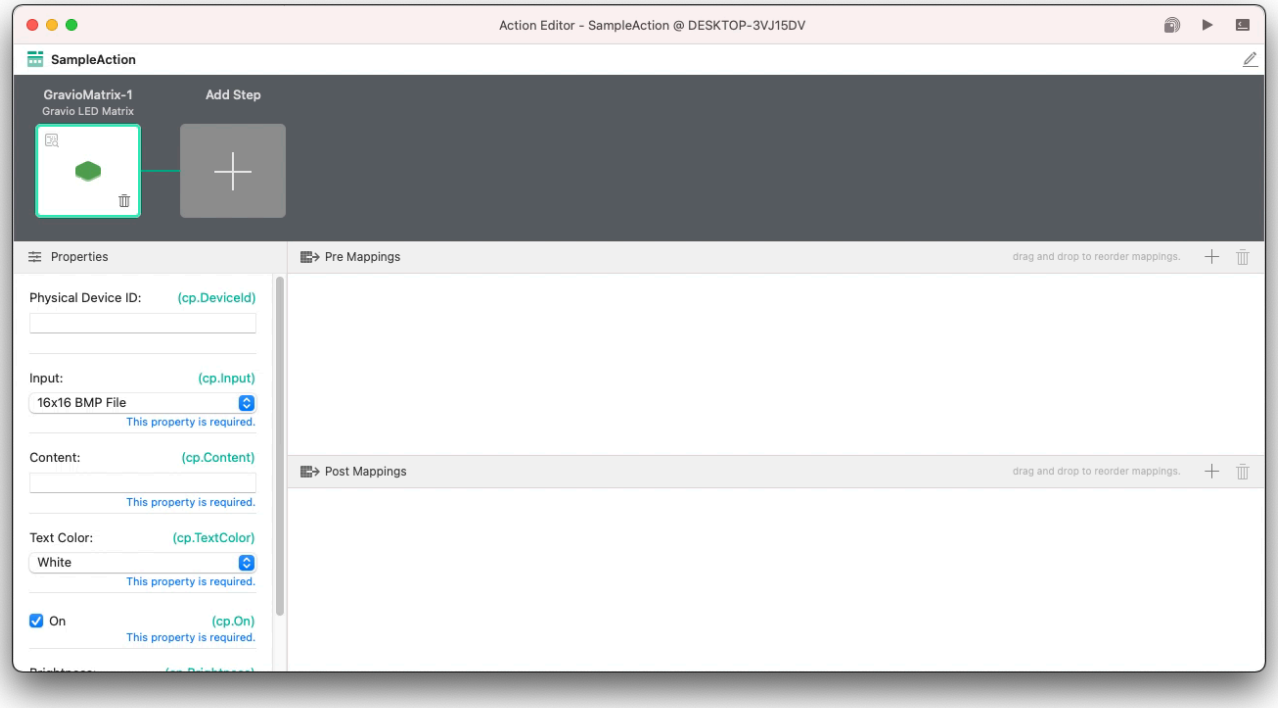
This is because the Distance sensor that is bound and enabled in the layer is set to send distance data every 5 seconds, and if the Gravio Distance component is executed when triggered by a trigger, the processes are colliding.

Distance sensors that are bound to a layer and enabled cannot be used in a Gravio Distance component, therefore they should only be used in one of the processes at a time.

6.3.9.24. GravioMatrix

The Gravio Matrix component allows you to set text or an image to a Gravio Matrix.
To use this component, the Gravio Matrix must be paired beforehand.

If the Device ID property is empty, the command will be sent to all connected Gravio Matrices.



Output payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (pass-through)

component properties

Property name	Description
Device ID	Device ID to which text or image is to be set or turned on/off, if none is set then this will be done for all Gravio Matrices If multiple Gravio Matrices are used then the device ID can be comma-separated.
Input	Set to “16×16 BMP file” or “ASCII”
Content	When the input is “16×16 BMP file”, save the BMP file in `actmgr/data/icons` and specify “ `icons/BMP filename` “ for the “Contents” property. For “ASCII”, specify one or two letters of the alphabet.
Font Color	Set the `cp.Color` to either of the values `white`, `black`, `red`, `green`, `blue`, or `auto`. “Auto”

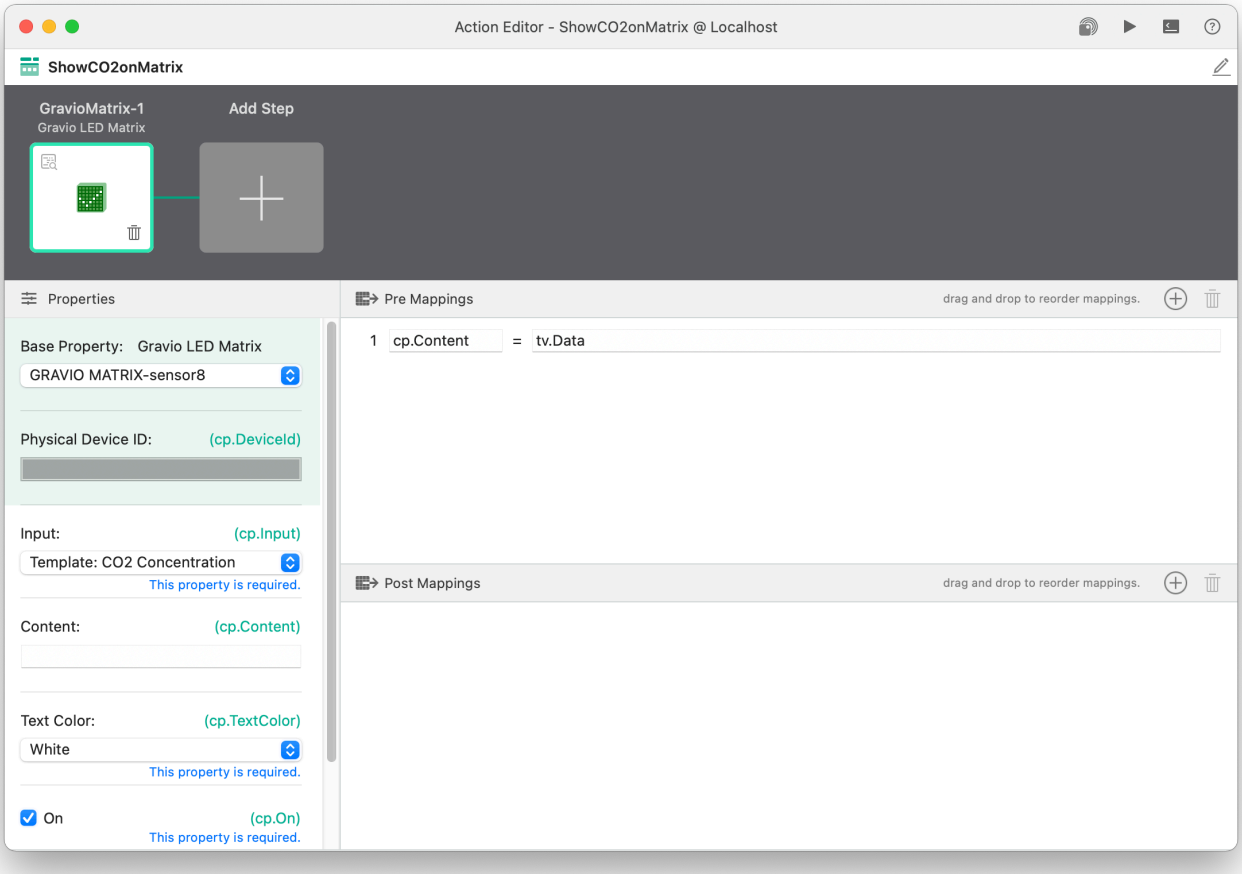
	means, in a temperature template the color will change according to the temperature: blue for below 10 degrees, green to 30 and red above 30.
On	On/Off – The value for cp.On can be “true” or “false”.
Brightness	Select brightness (0-100)

Using Templates

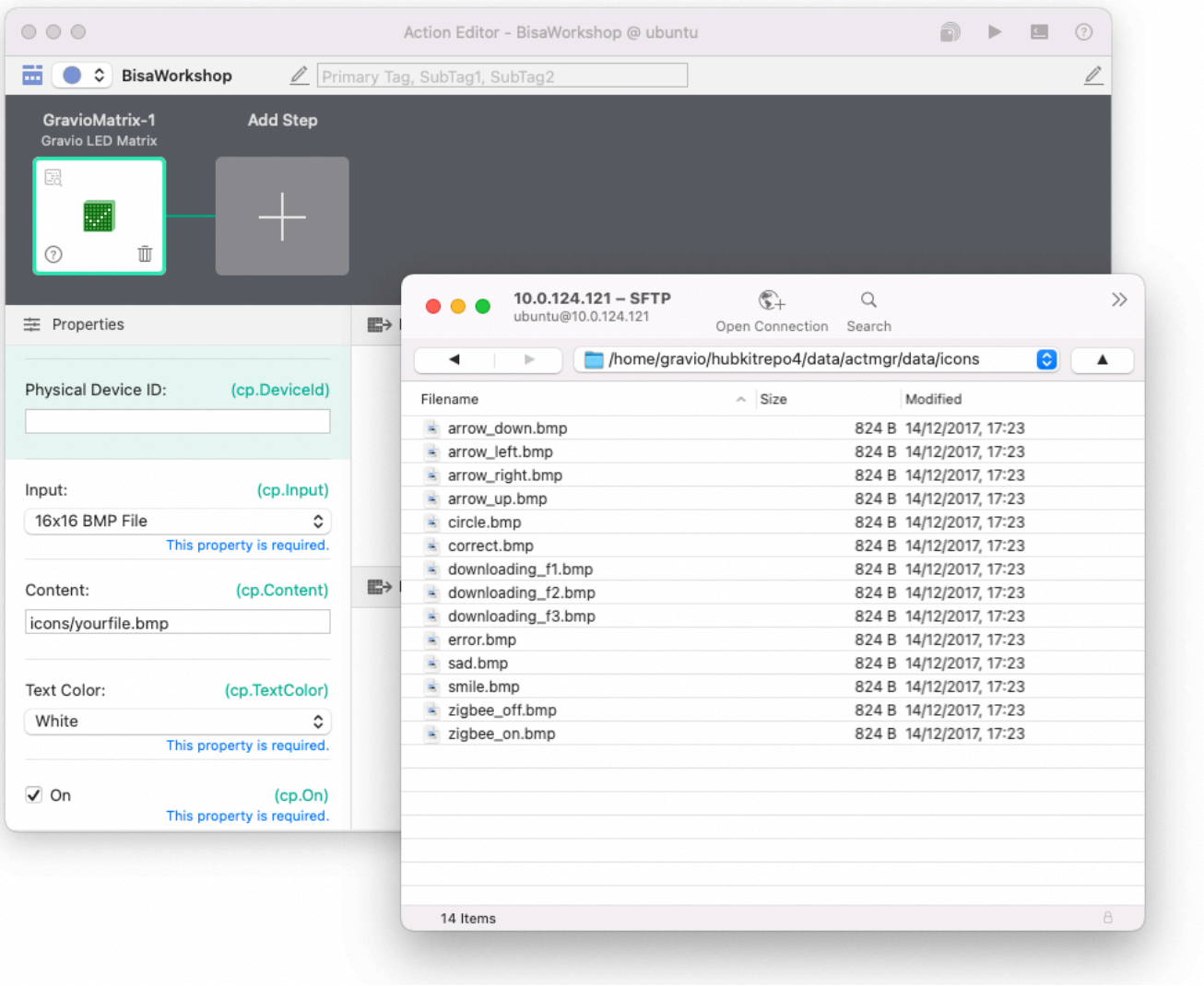
Using a template in the component properties (the Input drop-down), you can easily display the data alongside an icon. The template can be selected from the following, and if you specify a value for the content, the image and value that match the template will be displayed in the matrix.

Template	Description
Temperature ℃	Displays the value of the temperature (Celsius) sensor. Enter a number for the content
Temperature °F	Displays the value of the temperature (Fahrenheit) sensor. Enter a number for the content
Humidity%	Displays the value of the humidity sensor. Enter a number for the content
Atmospheric pressure hPa	Displays the value of the barometric pressure sensor. Enter a number for the content
CO2 concentration	Displays the value of the CO2 sensor. Enter a number for the content
Open / Close	Shows the opening and closing of the door. Enter either 0. Close or 1. Open for the content
date	Displays the current date (month and day). Enter `cp.Content = now()` in PreMapping for the content
time	Displays the current time (hours and minutes). Enter `cp.Content = now()` in PreMapping for the content

An Example setup using the CO2 template:



An Example setup using a bespoke image



You can use the xiconeditor.com to create the image online and the cloudconvert.com/ico-to-bmp to turn them into your own `bmp`

6.3.9.25. HTTP Request

Using the HTTP Request component, you can send an HTTP request to a specified URL.

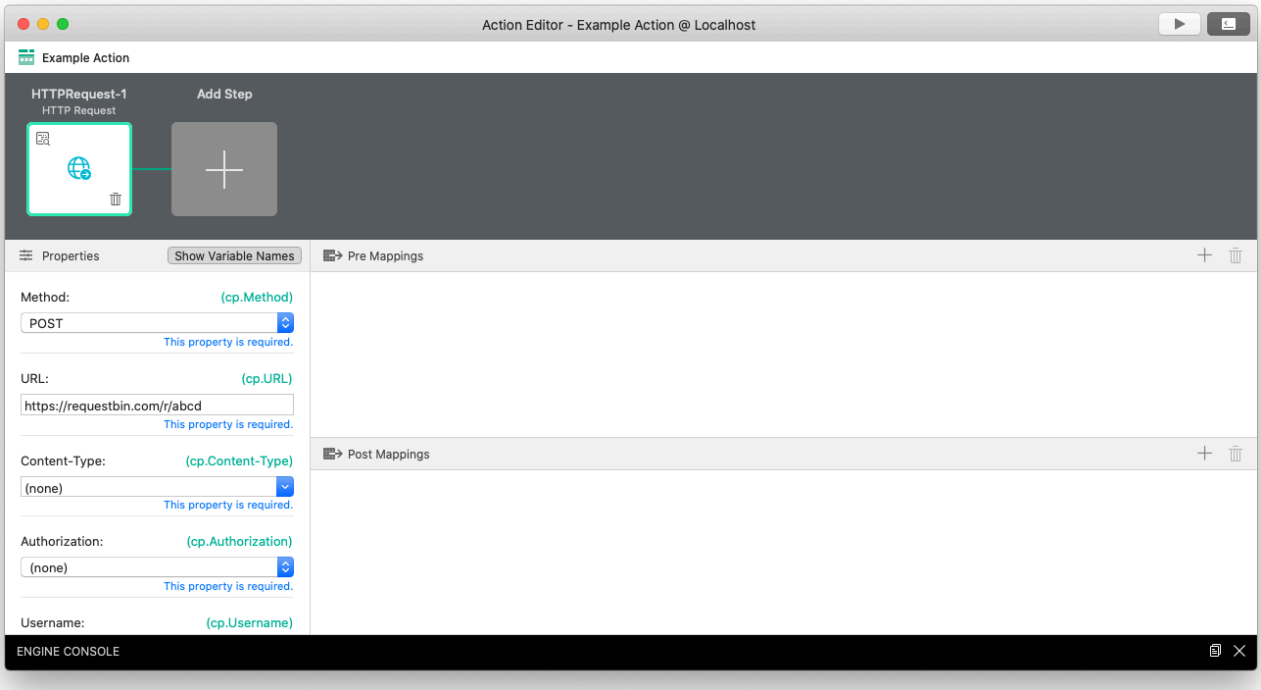
The HTTP Request component executes the HTTP request with the input payload as the Body content of the HTTP request, depending on its properties, and writes the HTTP response Body to the output payload (`cv.Payload`).

Body content in HTTP requests with Content-Type property

Content-Type Property	Content of the Body at the time of the HTTP request
None	The input payload executes the HTTP request without being set to the Body.
Default	Automatically selects the Content-Type based on the type of the input payload.
Character string type	Output characters with “text/plain; charset=utf-8”.
JSON array, JSON object	The HTTP request is executed as “application/json”, marshalling the Payload as a byte string.
Byte sequence	The HTTP request is executed with the string set to “application/octet-stream” as the Body.
Other	The HTTP request is executed with the string set to “text/plain” as the body.
application/x-www-form-urlencoded	The input payload is a JSON object type, for example in the Pre-Mapping: `cv.Payload = {"key": "value", "otherkey": "othervalue"}` (In this case, if data other than the JSON Object type is received as input, an error occurs.)
multipart/form-data	The input payload is JSON Object type, and the key and the value are separated by multipart, and the Content-Type multipart/form-data is automatically attached to the boundary before the HTTP request is executed. (In this case, if data other than the JSON Object type is received as input, an error occurs.)
Any other arbitrarily entered Content-Type	The Content-Type is left as it is and the output is a string of bytes of the input payload as the Body.

Output Payload	Description
`cv.Payload`	The output payload (`cv.Payload`) stores the body of the HTTP response as a string of

	bytes.



Properties

Name	Required	Description
Method	True	The HTTP Request Method
URL	True	The endpoint URL
Content-Type	True	The HTTP Header Content-Type
Authorization	True	The authorization type. When specified by the value of cp.Authorization, it can be one of “(none)”, “prebasic”, “basicdigest”, or “bearer”.
Username	False	Basic authentication username
Password	False	Basic authentication password
Token	False	The Bearer Token
Timeout	True	HTTP Request Timeout (ms)
HTTP Proxy	False	HTTP Proxy
Proxy URL	False	URL when using a proxy
Proxy Username	False	Username when using a proxy
Proxy	False	Password when using a proxy

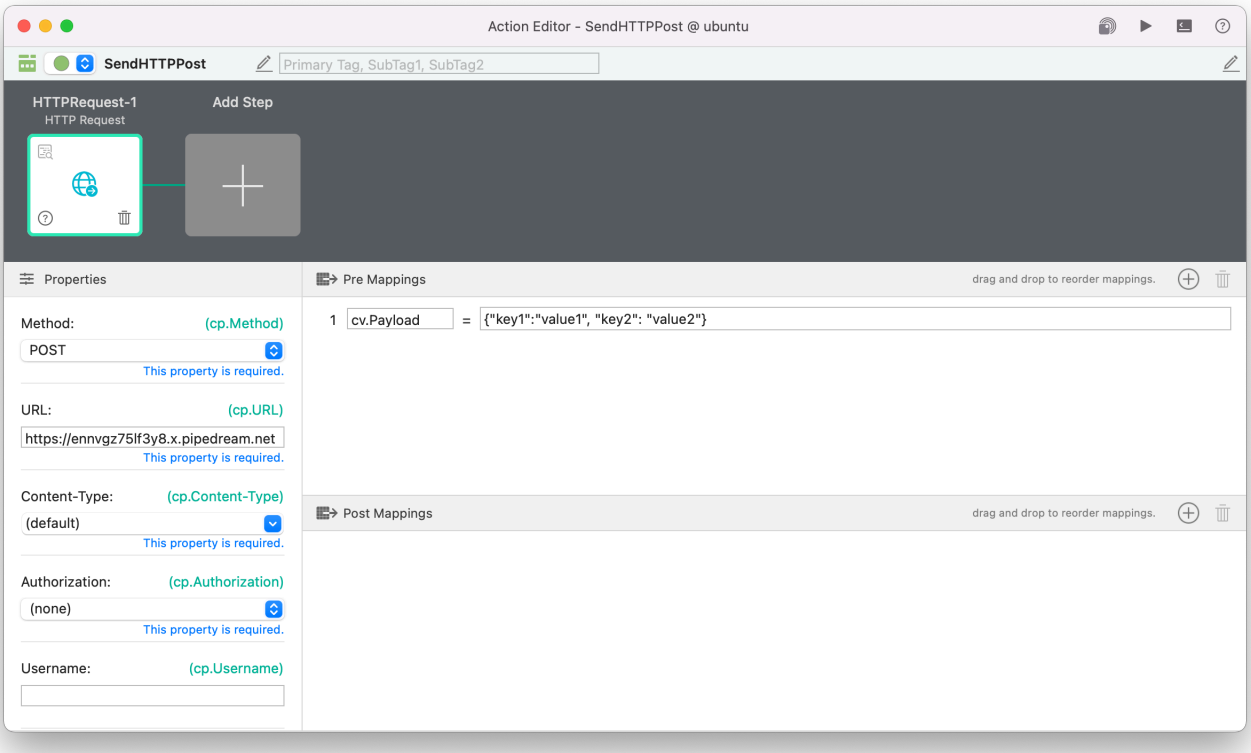
Password		
Dump	True	Dump the request/response to the output log (`actmgr...` logfile in the Gravio HubKit log directory of your OS.)

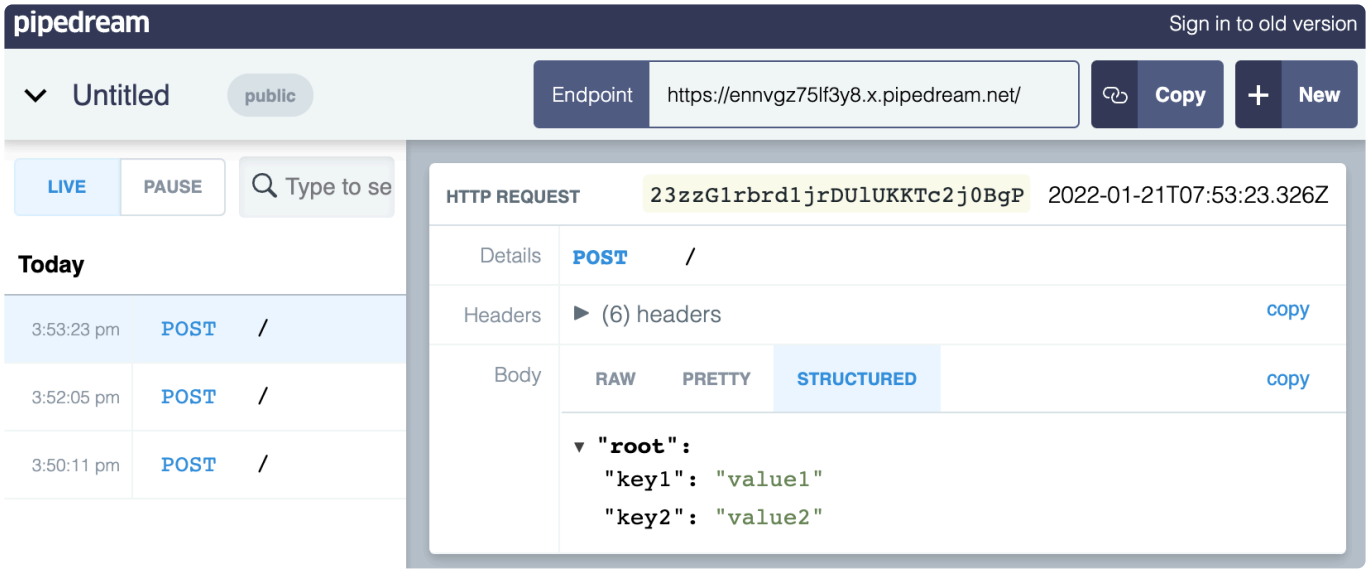
Component Variables

`cv.Payload`	On input: request body, on output: response body.
`cv.StatusCode`	Request result
`cv.Headers`	On output: response header will be stored in this component variable. To send custom request headers, please use component properties `cp.Headers` (see example below)
`cv.ContentLength`	Response content length

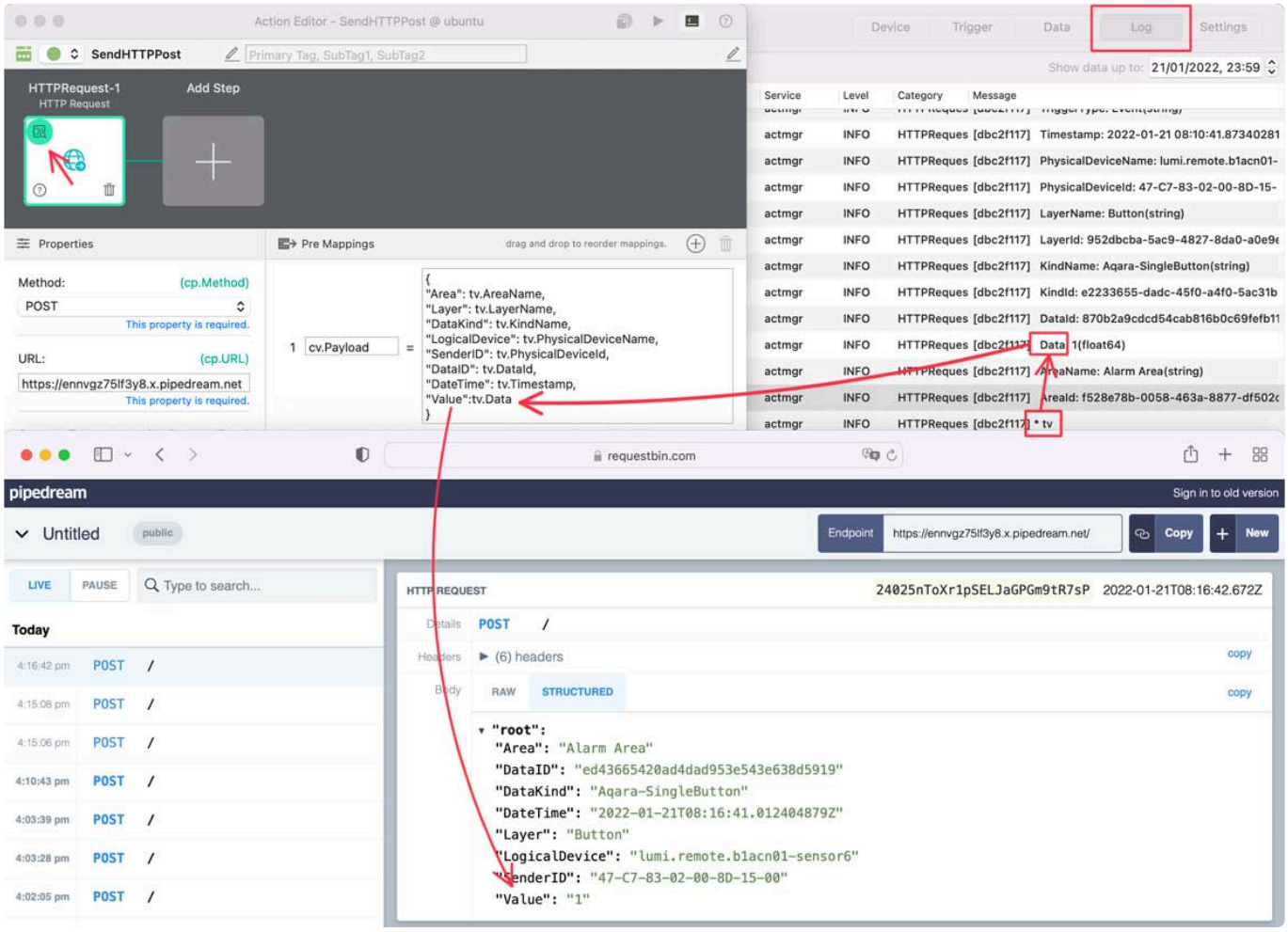
Testing the HTTP Requests

We recommend a free service called requestbin.com to create a testing endpoint and to send your HTTP requests to. In their live dashboard you can see the incoming requests. (Note, the above link creates a public requestbin. For private requestbins, you will need an account).





If you enable the “debug” flag on the top left of the component box, you will find the variables in the log. There you can also see what the variable scopes (here `tv.`) and names (for example `Data`) are. These you can then refer to inside your JSON string in the `cv.Payload`:



Here is a JSON example including an MD5 checksum you can use as template:

```
{
```

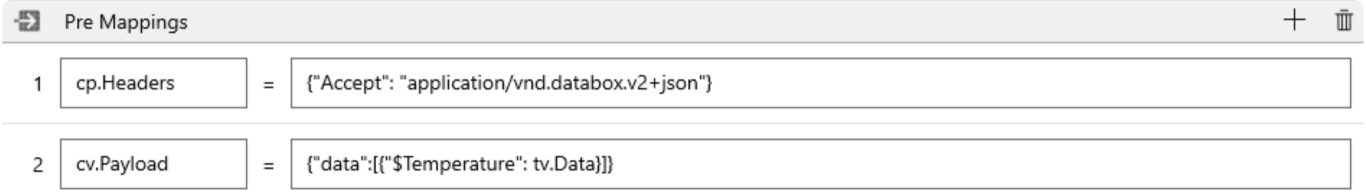
```
"AreaName": tv.AreaName,
"LayerName": tv.LayerName,
"KindName": tv.KindName,
"PhysicalDeviceName": tv.PhysicalDeviceName,
"PhysicalDeviceId": tv.PhysicalDeviceId,
"DataId": tv.DataId,
"Timestamp": ToInt(tv.Timestamp),
"Data":tv.Data,
"Checksum":MD5(tv.AreaName+tv.Data+tv.KindName+tv.PhysicalDeviceName+tv.Physic
alDeviceId+tv.DataId+ToInt(tv.Timestamp)+tv.LayerName+"SECRETSALT")
}
```

Overriding or Extending Header Information

You can add or override existing `Header` information by creating a `PreMapping` using this this `JSON` syntax:

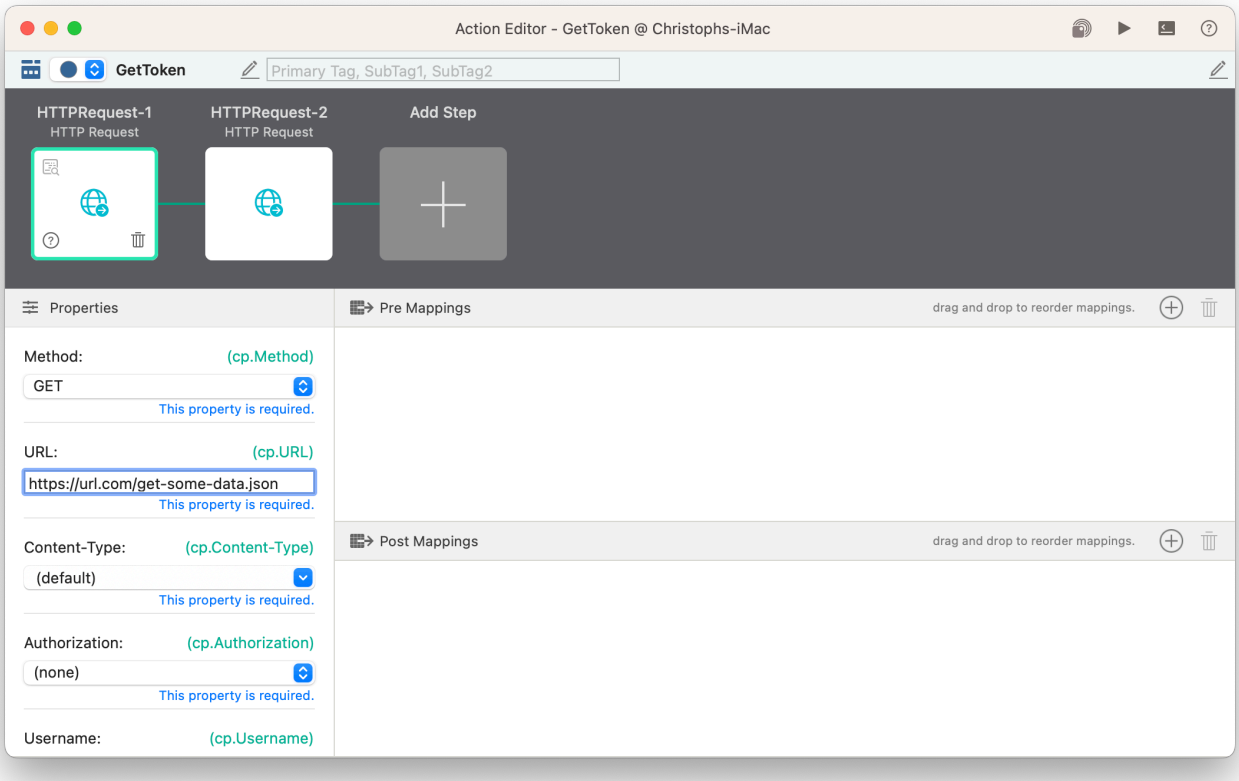
```
cp.Headers = {"X-Header-1": "abc; def", "X-Header-2": ap.Timeout/1000, "X-Heade
r-3": true, "X-Header-4": cp.URL, "Accept", "application/json", "User-Agent",
"fugahoge browser"}
```

For example, Databox, a visualisation software requires certain headers. They can be set as follows:

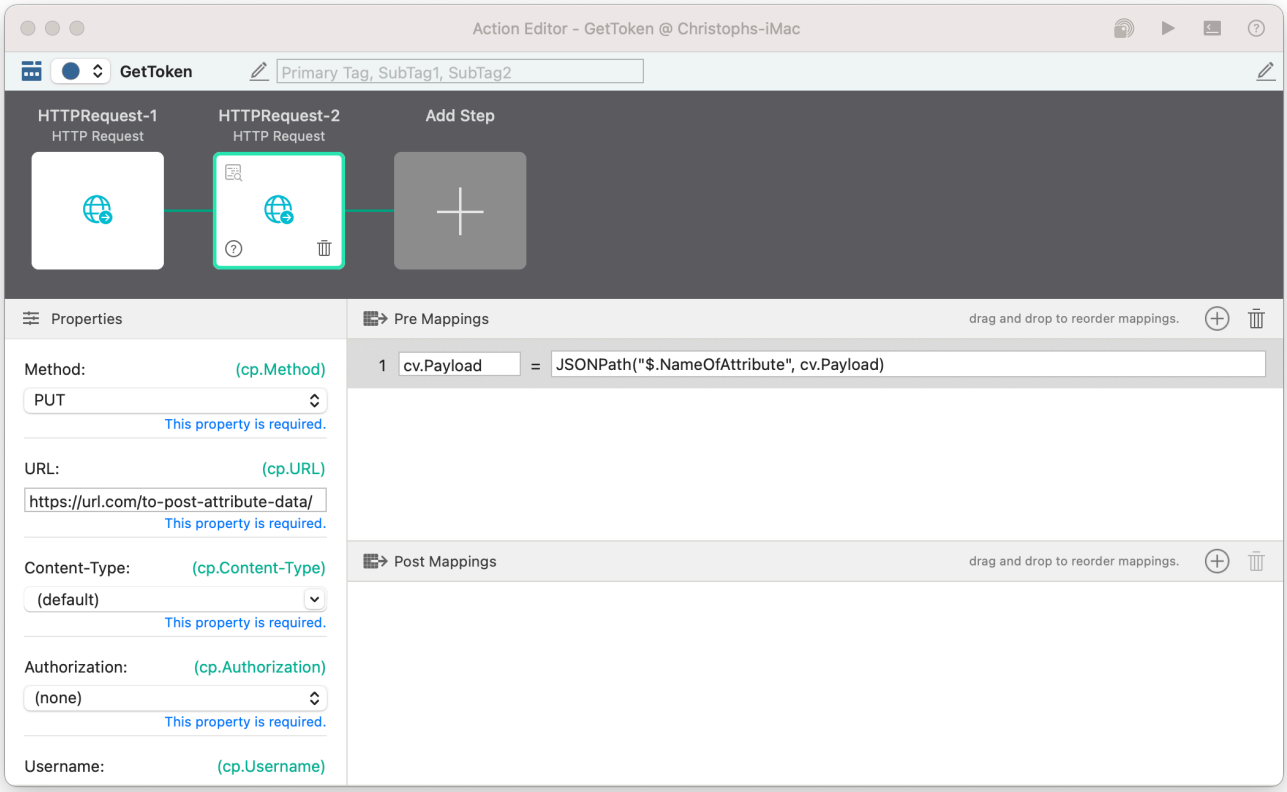


Using a JSON HTTP Response Body in a Subsequent HTTP Request

You can take the JSON response from a HTTP request and use JSONPath() to extract a specific attribute of the reply.

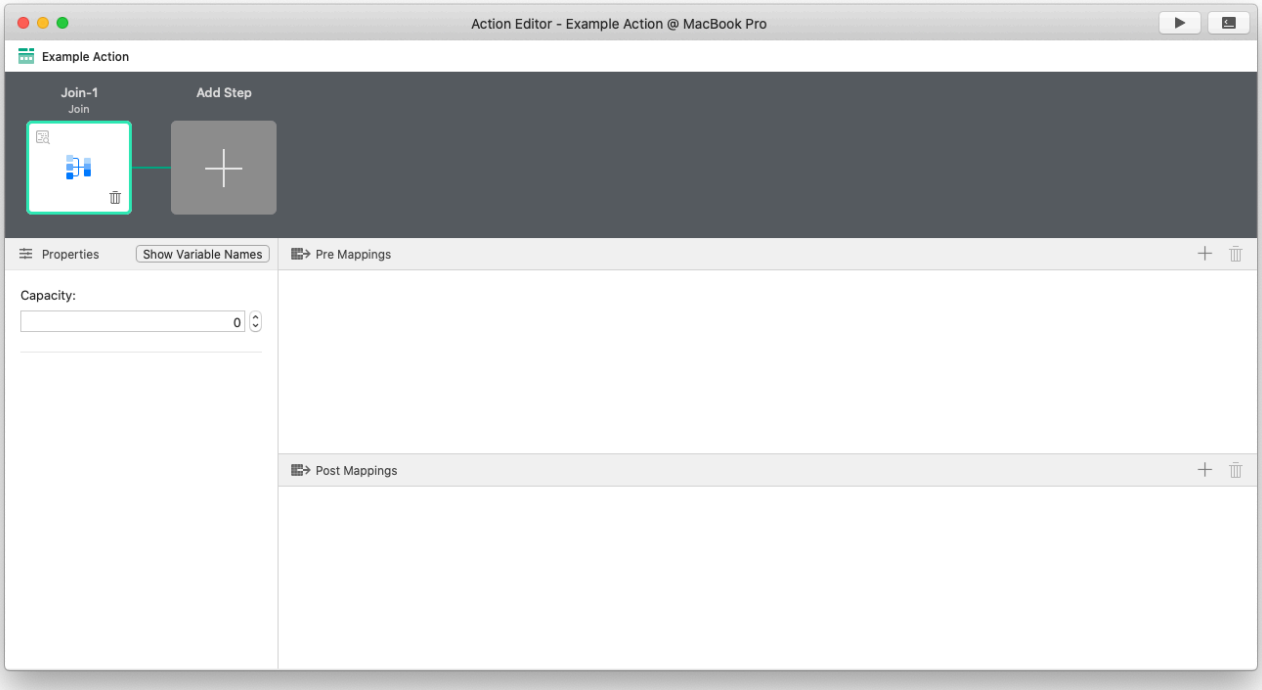


If you want to use the data in a subsequent component, e.g. a HTTP request post data, it could look like this:



6.3.9.26. Join

The Join Action Component is to join multiple inputs into an array.



Output Payload	Description
`cv.Payload`	Combined array

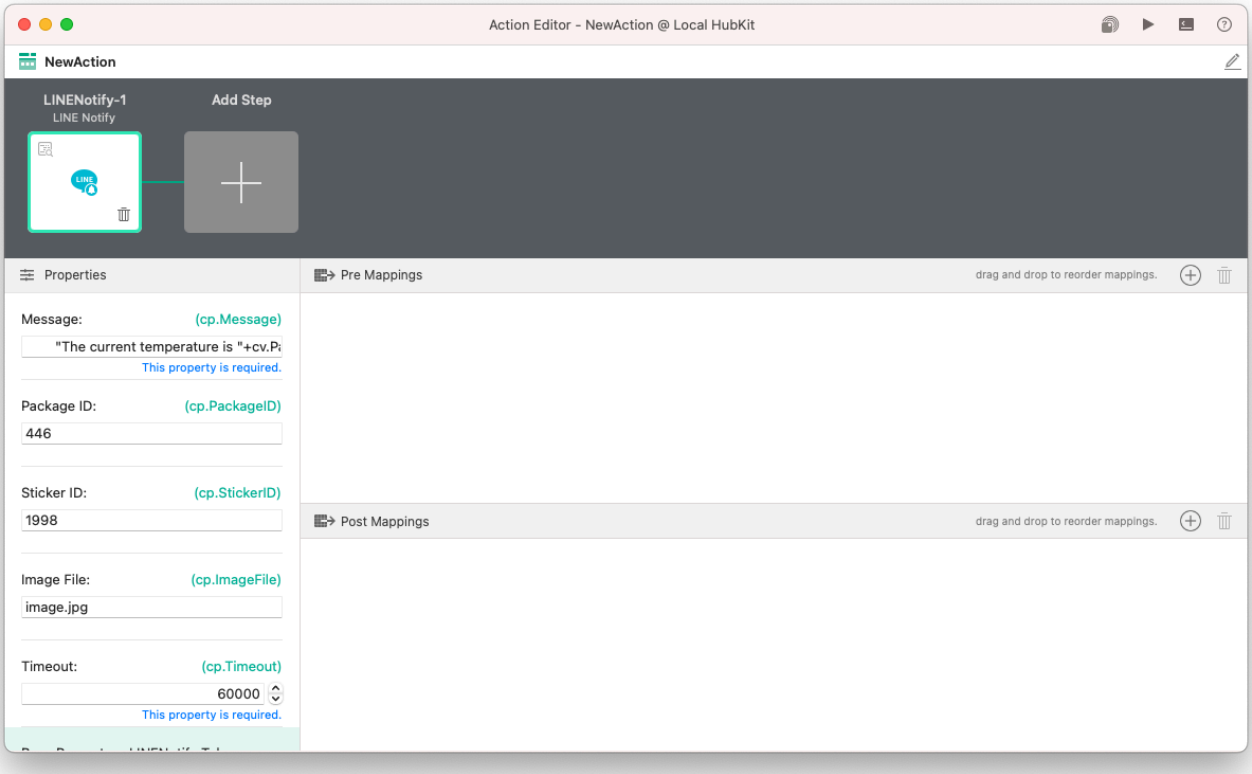
Component Properties

Name	Required	Description
Capacity	False	Specifies how many incoming values are to be output together.

If you set the value to `0` you will capture all amount of incoming data.

6.3.9.27. LINE Notify

The LINE Notify Action Component is to send a notification via the LINENotify service (https://notify-bot.line.me/). It's useful to notify people of information from sensors.



Component Properties

Property Name	Description	Example
Message	Message string to be posted to LINE	`"The current temperature is "+cv.Payload+"degrees"`
Package ID	Send a stamp with the message. You will need the Package ID alongside the Stamp ID (see comment below)	`446`
Stamp ID	Sends a stamp with the message, requires the corresponding Package ID (above)	`1988`
Image File	Send an image with the message, specify the file path of a JPG or PNG file	`image.jpg`
Timeout	Timeout (ms)	10000
Token	Authentication Token, issued by LINE Notify, if not taken from the Base Property Profile	

Notes:

For details on how to specify the Package ID and Stamp ID for LINE stamps, please refer to the LINE documentation

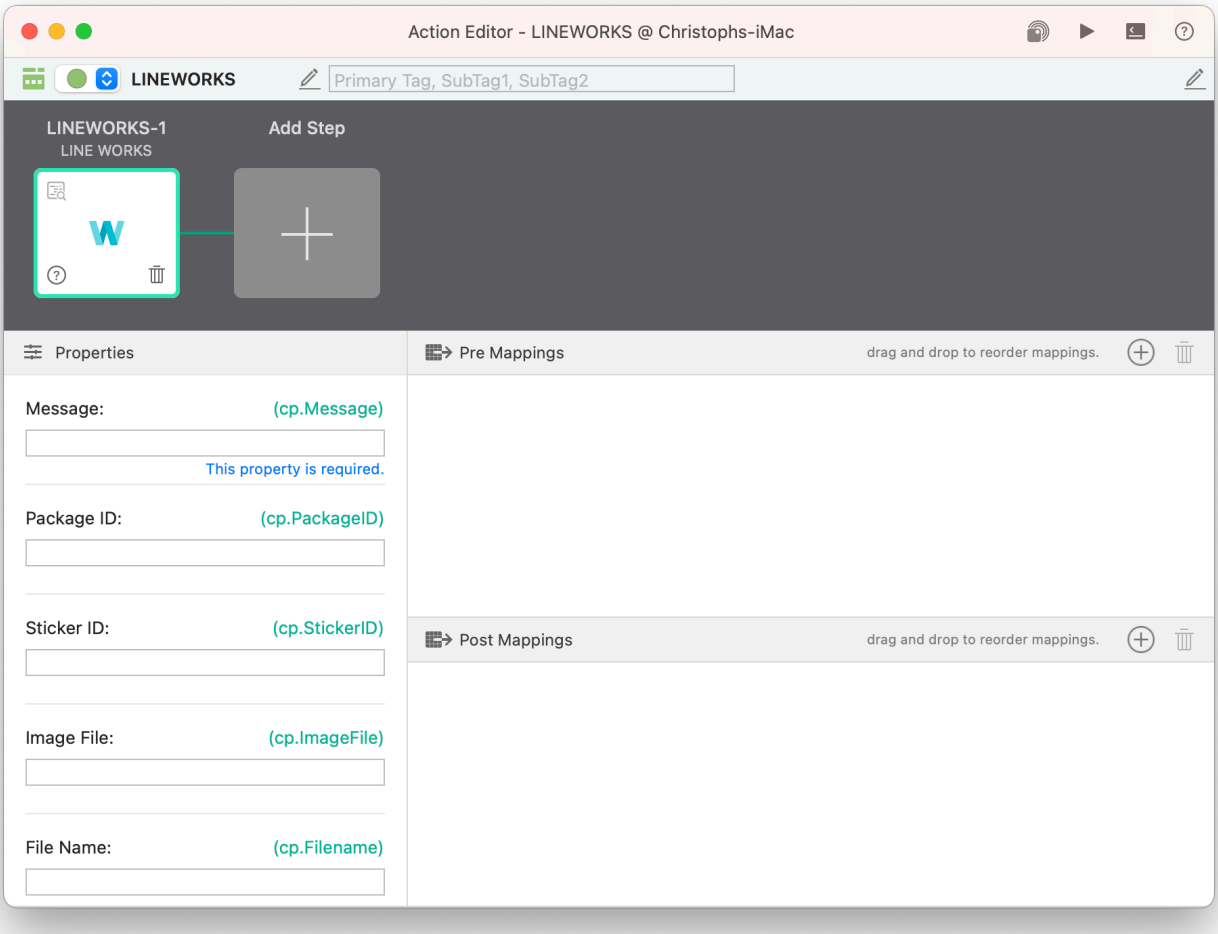
For more information on how to specify the file path, please refer to “

6.3.9.28. LINE WORKS

The LINE WORKS component can post messages specified in the component's properties using LINE WORKS, the business solution of LINE Messenger. This is useful for notifying people of information from sensor data.

In order to use the LINE WORKS component, the administrator must initially configure it in the LINE WORKS Developer Console and set the bot as public. For details on how to configure it in the LINE WORKS Developer Console, see here TOPIC-LINK+line-works-bot} for details on how to set up the bot in the LINE WORKS Developer Console.

When using the LINE Notify service (https://notify-bot.line.me/), please use the LINE Notify action component.



Component Properties

Property Name	Description	Example
Basic	Specify the basic properties of the "LINE WORKS Bot	

Properties	Connection" category	
Message	Message string to be posted to LINE WORKS	Example: `"The current temperature is "+cv.Payload+"degrees"`
Package ID	Send a stamp with the message. Specify with stamp ID *1	Example: `446`
Stamp ID	Send stamp with message. Specify with package ID *1	Example: `1988`
image file	Send an image with the message, specifying the file path of a JPEG or PNG file.	Example: `image.jpg`
Send the file with the message. file	Sends a file with the message. Files with extensions `bat` , `cmd` , `com` , `cpl` , `exe` , `scr` , `js` , `vbs` , `wsf` , `hta` cannot be sent	
Destination type	Select the member account or talkroom ID to send the message to	
Destination ID	Specify the member account or talkroom ID you selected in the Destination type	Example: aaaa@bbbb for member accounts, `12345` for talkroom IDs
Timeout timeout	API timeout	
API ID	Specify the API ID from the LINE WORKS Developer Console *3	
Consumer Key	Specify the Consumer Key from the LINE WORKS Developer Console *3 Consumer Key	Specify the Consumer Key from the LINE WORKS Developer Console *3
Consumer Key Server Token	Specify the Token for the Server List (fixed IP type) from the LINE WORKS Developer Console *2 *3	
Server ID	Specify the Server ID for the Server List (ID registration type) from the LINE WORKS Developer Console *2 *3	
Server ID Authentication Key	Specify the authentication key for the Server List (ID registration type) from the LINE WORKS Developer Console *2 *3	
Bot No	Specify the Bot No from the LINE WORKS Developer Console *3	

*1 For information on how to specify the LINE Stamp Package ID and Stamp ID, please refer to here. pdf. Please refer to the pdf file.
*2 The Server Token, Server ID, and authentication key will use the one set in the Server List (fixed IP

type) or Server List (ID registration type).
*3 Refer to here for items to be specified from the Developer Console.

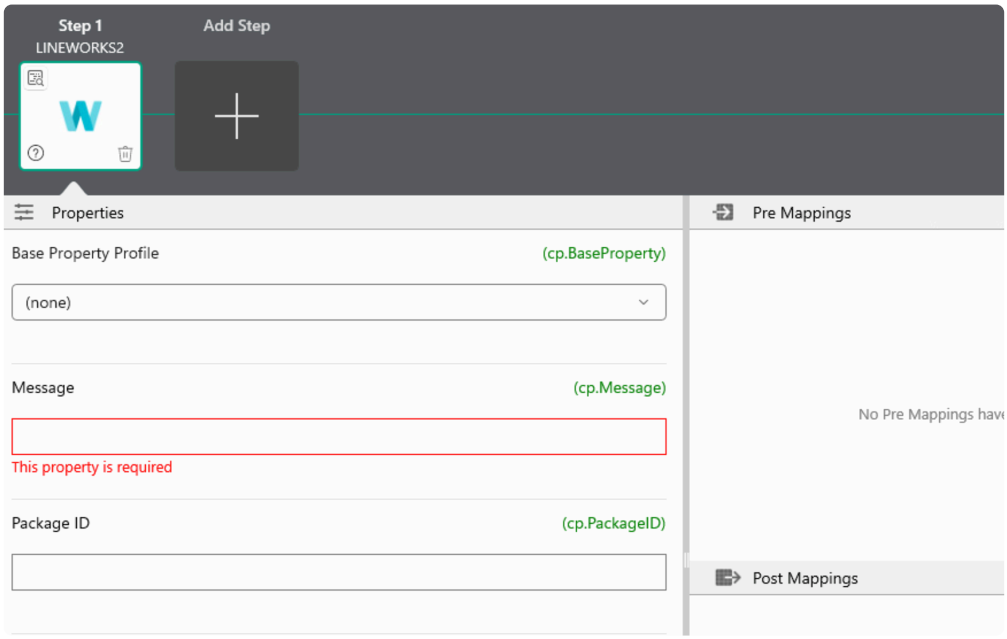
The image file/file reference without specifying the file path will be `actmgr/data`.

6.3.9.29. LINE WORKS

The LINE WORKS component can post messages specified in the component properties using LINE WORKS. This is useful for notifying people about information from sensors.

※ This LINE WORKS component is a new component that uses the LINE WORKS API 2.0. The conventional LINE WORKS component used API 1.0, which is not compatible. New settings are required on the LINE WORKS Developer Console. Please refer to here for how to set up in LINE WORKS Developer Console.

※ To use the LINE WORKS component, the administrator must initially set up in the LINE WORKS Developer Console and publish the Bot. Please refer to here for how to set up in LINE WORKS Developer Console.
※ Please use the LINE Notify Action Component if you are using the LINE Notify service (https://notify-bot.line.me/).



Component Properties

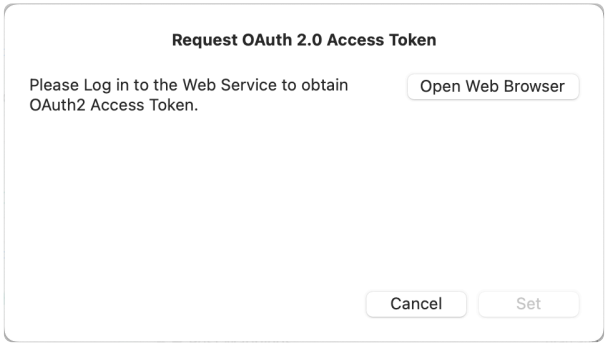
Property Name	Description	Example
Basic Properties	Specify the basic properties of the “LINE WORKS Bot Connection” category	
Message	Message string to post to LINE WORKS	e.g. `"The current temperature is"+cv.Payload+"degrees"`
Package ID	Sends a stamp along with the message. Specify it together with the Stamp ID ※1	e.g. `446`
Stamp ID	Sends a stamp along with the message. Specify it	e.g. `1988`

	together with the Package ID ※1	
Image File	Sends an image along with the message. Specify the file path of the JPEG or PNG file.	e.g. `image.jpg`
File	Sends a file along with the message. Files with extensions `bat`, `cmd`, `com`, `cpl`, `exe`, `scr`, `js`, `vbs`, `wsf`, `hta` cannot be sent	
Destination Type	Choose the destination to which the message is sent, either a member account or a chat room ID	
Destination ID	Specify the member account or chat room ID selected by Destination Type	e.g. In case of member account `aaa`bbbb@, in case of chat room ID `12345`
Timeout	API Timeout	
OAuth2 Client ID	Specify the Client ID for API 2.0 from the LINE WORKS Developer Console ※2	
OAuth2 Client Secret	Specify the Client Secret from the LINE WORKS Developer Console ※2	
OAuth2 Redirect URL	Specify the Redirect URL from the LINE WORKS Developer Console ※2	
OAuth2 Access Token	Press the Get Token button to get the token. You can also press the Refresh Token button to update it ※2 ※3	
Bot No	Specify the Bot No from the LINE WORKS Developer Console ※2	

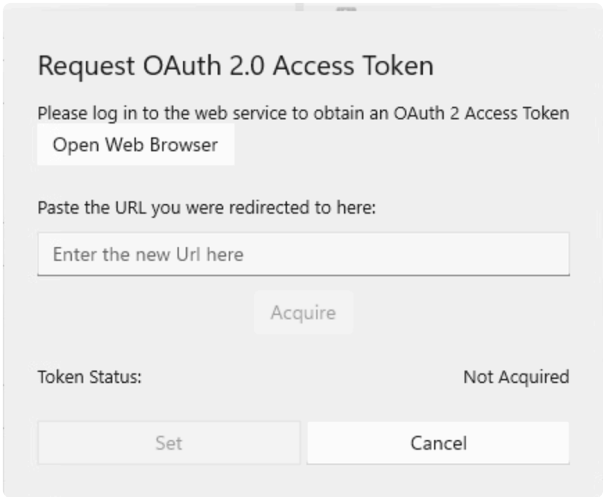
※1 Please refer to here for how to specify the LINE Stamp Package ID and Stamp ID.
※2 Please refer to here for the items to specify from the Developer Console.

How to Obtain the OAuth2 Access Token

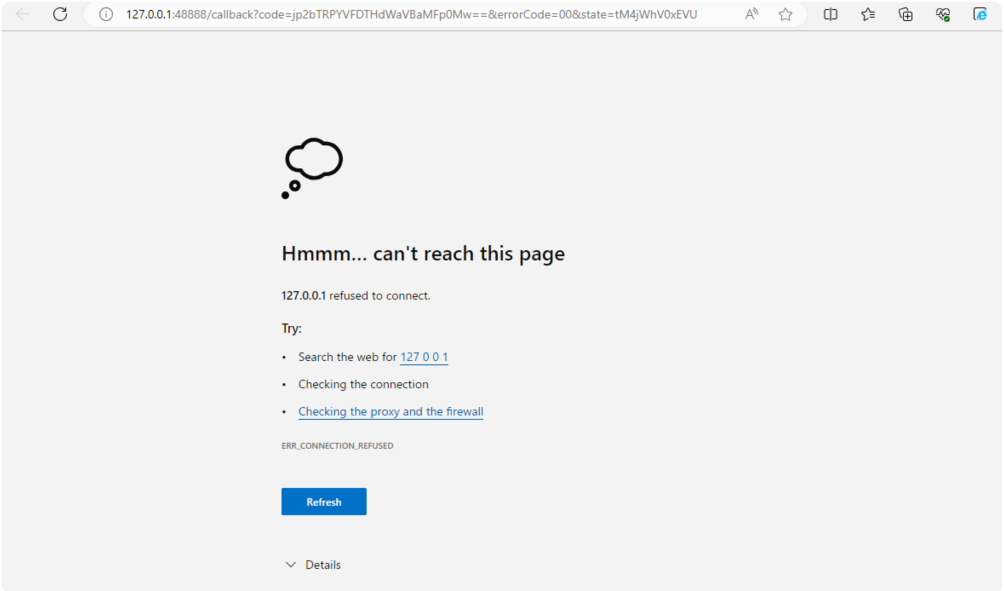
※3 When you press the Get OAuth2 Access Token button, the OAuth 2.0 Access Token Acquisition panel is displayed. Press the Open Browser button to obtain the token. When you get the token, press the Set button and it will be automatically set to the OAuth2 Access Token property.



In the case of Gravio Studio for Windows, press the Open Browser button to get the token, but an error screen will appear. Copy the URL from the address bar in this state and paste the URL into the panel and press the Get button. Once the token is obtained, press the Set button and it will be automatically set to the OAuth2 Access Token property.

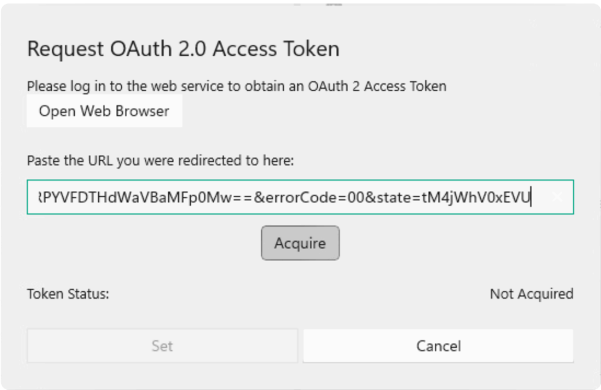


When you press the Open Browser button, authentication will automatically be performed in the browser. If necessary, follow the instructions on the screen to log in.



When authentication is complete, an error saying “This page cannot be reached” will be displayed, so

copy the URL from the address bar at this time.



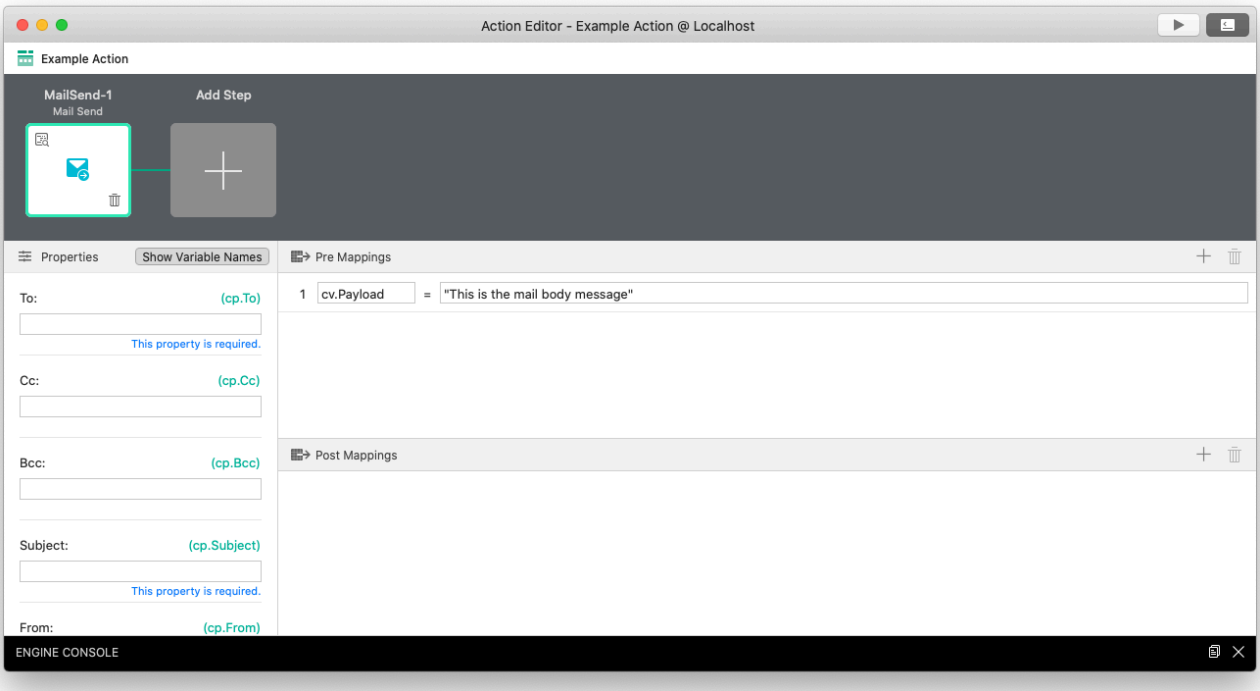
Paste the copied URL into the panel and press the Get button, and the OAuth2 Access Token will become “obtained”. Press the Set button to close the panel.

6.3.9.30. Mail Send

The Mail Send Action Component is to send email via SMTP.

The Mail Send component sends an input payload (`cv.Payload`) as the body of a mail.
It converts the data received by the input payload into a string and sends it as the body of an email.

After sending the mail, the input payload is stored as it is as the output payload (pass through).



Properties

Name	Required	Description
To	True	To whom the mail will be sent
Cc	False	Any carbon copies
Bcc	False	Any blind carbon copies
Subject	True	The subject of the mail
From	True	The from-address
Attachments	False	File(s) to be attached, details below
SMTP Server	False	SMTP Server you are going to use
Host	True	The smtp host
Port	True	Port

Username	False	Username of your smtp server
Password	False	Password of your smtp server

Use the base Base Property Profile property to read the values from the settings rather than specifying the details within the component.

Component Variables

Name	Description
`cv.Payload`	This variable will be used as the mail body. You can set it in the Pre Mappings, if no Payload is being passed into this step. (See screenshot) If you are using a HTML mail body, avoid linebreaks and escape double quotes with a backslash.

Using SMTPs

You can use any available SMTP server to send out e-mails, however, note that depending on what server you are using, they may have different restrictions in order to avoid spamming. To send mails via Apple iCloud or Google Gmail SMTP, we recommend creating app-specific passwords.

Attachments

Multiple files can be specified as attachments.

- Path of the file to be attached

Windows environment

The files are automatically stored in the path actmgr\data. The path can also be specified.
*If a folder name or file name contains spaces, it must be enclosed in “” (double quotes).

Example)

```
To attach image.jpg from actmgr\data, specify image.jpg as the attachment file.
To attach image.jpg of actmgr\data\subfolder, specify subfolder\image.jpg as the attachment.
To attach image.jpg of actmgr\data\subfolder 2021, specify "subfolder 2021\image.jpg" as the attachment.
```

Absolute paths can also be specified.

Example)

bc… To attach image.jpg in c:\, specify c:\image.jpg as the attachment.

Files can also be specified as multiple files and wildcards.
Use “;” (semicolon) for multiple file separators and “*” (asterisk) for wildcards.

Example)

```
To attach image.jpg and image2.jpg from actmgr\data, specify image.jpg;image
2.jpg as attachments.
To attach *.jpg in actmgr\data\subfolder, specify subfolder\*.jpg as the attac
hment.
To attach image.jpg and image2.jpg from actmgr\data and *.jpg from actmgr\dat
a\subfolder, specify image.jpg;image2.jpg;subfolder\*.jpg as attachments.
```

Mac, Linux, RaspberryPI and Gravio Hub environments.

Files are automatically stored in the path actmgr/data. The path can also be specified as.
*If a folder name or file name contains spaces, it must be enclosed in “” (double quotes).

Example)

```
To attach image.jpg from actmgr/data, specify image.jpg as the attachment.
To attach image.jpg of actmgr/data/subfolder, specify subfolder/image.jpg as a
ttachment.
To attach image.jpg of actmgr/data/subfolder 2021, specify "subfolder 2021/ima
ge.jpg" as attachment.
```

Absolute paths can also be specified.

Example)

```
To attach image.jpg of /home/username, specify /home/username/image.jpg as th
e attachment.
```

Files can also be specified as multiple files and wildcards.
Use “;” (semicolon) for multiple file separators and “*” (asterisk) for wildcards.

Example)

```
To attach image.jpg and image2.jpg from actmgr/data, specify image.jpg;image
2.jpg as attachments.
To attach *.jpg in actmgr/data/subfolder, specify subfolder/*.jpg as attachmen
t.
To attach image.jpg and image2.jpg from actmgr/data and *.jpg from actmgr/dat
a/subfolder, specify image.jpg;image2.jpg;subfolder/*.jpg as attachments.
```

6.3.9.31. MQTT Publish

The MQTT Publish Action Component allows you to publish messages to topics within an MQTT PubSub system.

Output Payload	Explanation
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass through)

Component Properties

Name	Required	Description
Topic	True	MQTT Topic
ClientId	True	MQTT ClientID
QoS	True	QoS value
Retain	True	Retain flag
Timeout	True	Timeout (ms)
MQTT Connection	False	MQTT Connection Settings from the Base Property Profile
URL	True	URL, note, leave the protocol out for unencrypted connections (e.g. `test.mosquitto.org`) but add `ssl://` for encrypted connections
Port	True	Port
Username	False	Username
Password	False	Password
Certificate	False	Certificate
Private Key	False	Private Key
Private Key Passphrase	False	Private Key Passphrase
Root CA	False	Root CA

test.mosquitto.org Example

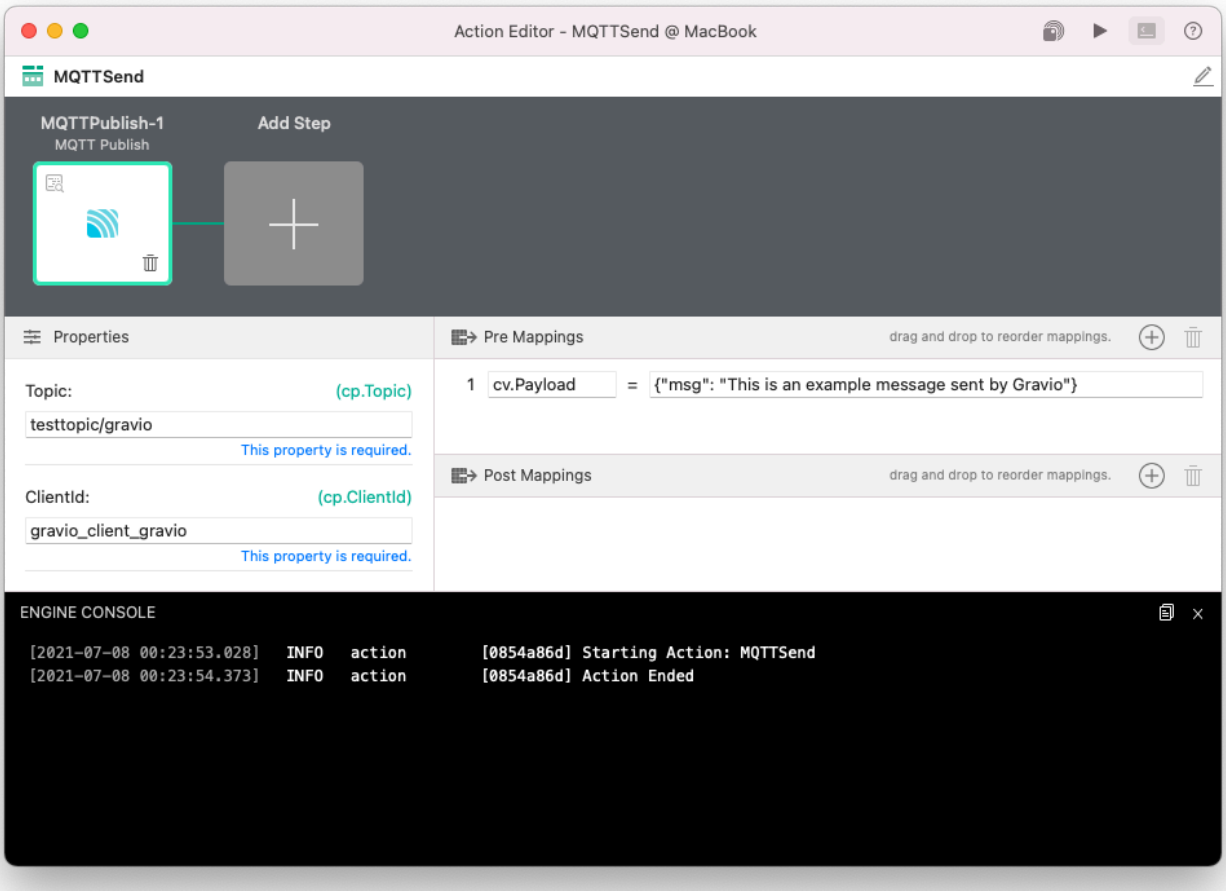
You can easily test the MQTT function by using the free public test broker provided by mosquito.org. A sample setup could be that you use the free MQTTX client alongside with Gravio using the following details:

Topic	`testtopic/gravio`

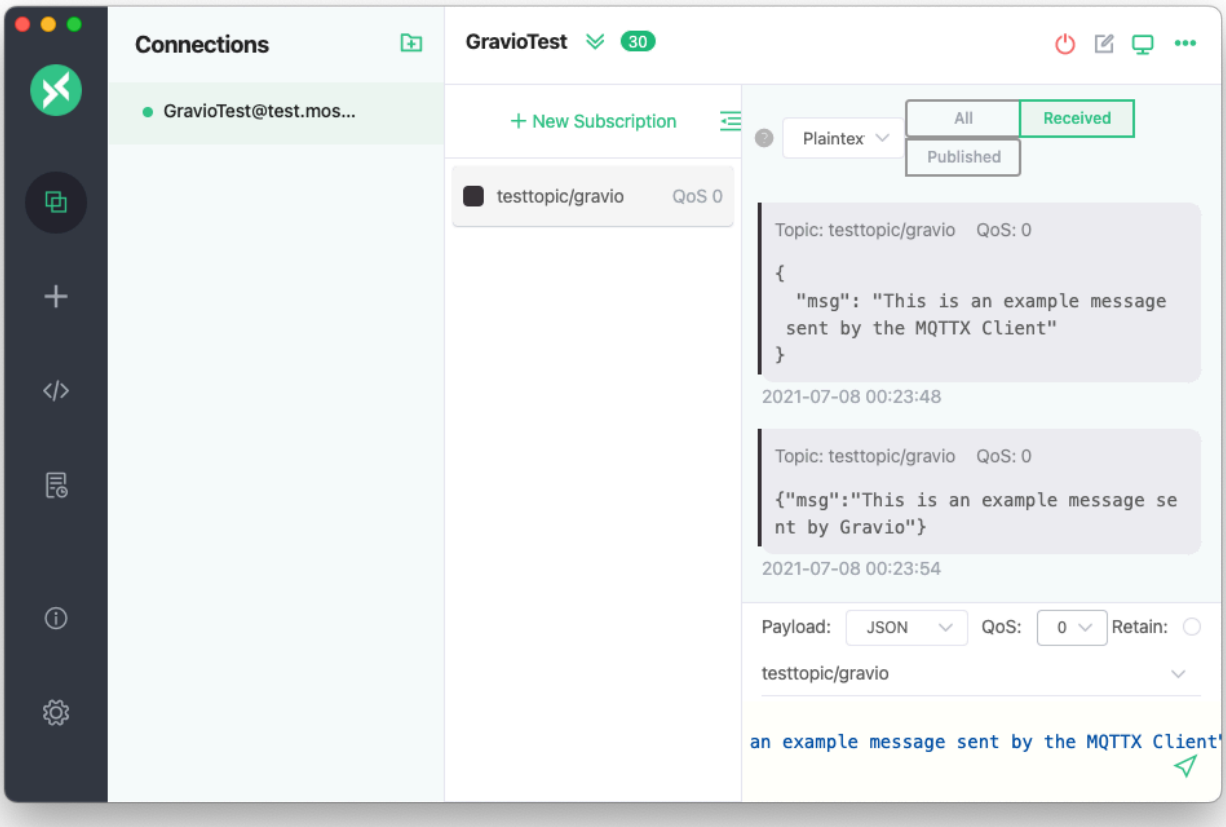
Client ID	Choose something Unique for Gravio and something different for MQTTX
QoS	0
Timeout	60000
URL	`test.mosquitto.org`
Port	`1883`
Username	*blank*
Password	*blank*
Certificate	*blank*
Private Key	*blank*
Private Key Passphrase	*blank*
Root CA	*blank*

Then set the contents to `cv.Payload = {"msg": "your test message"}`

Triggering the action from Gravio Studio:



You should see something similar to the following in MQTTX:

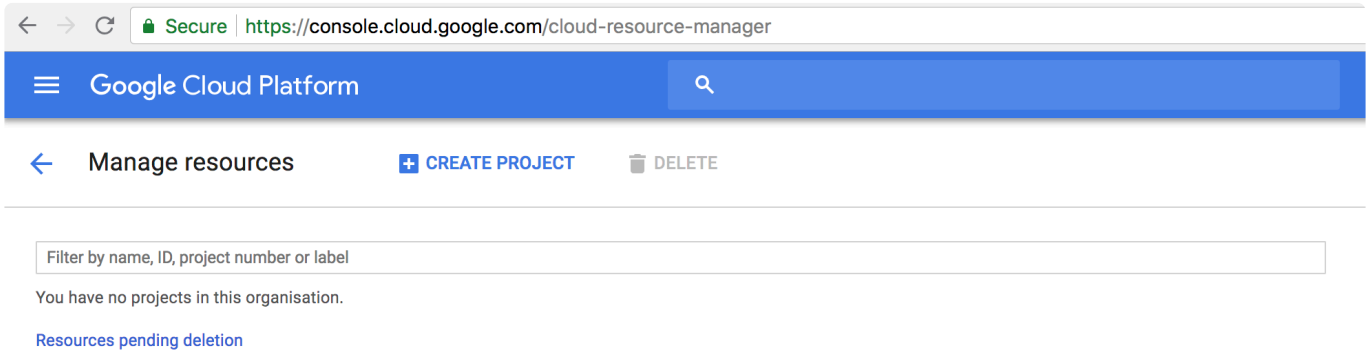


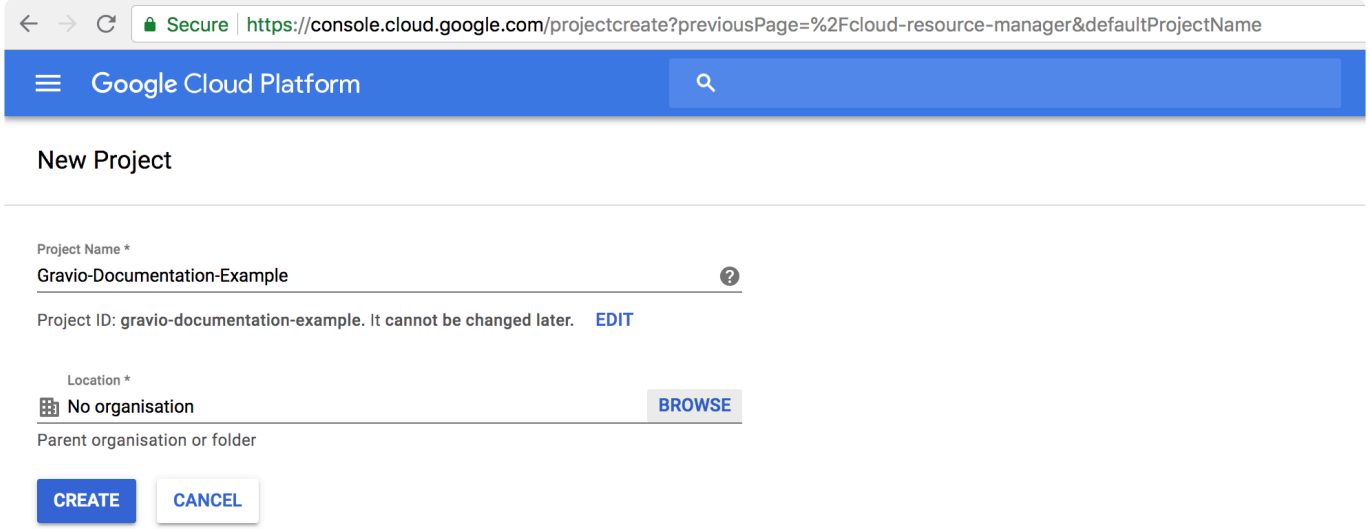
Note: Ensure that both, MQTTX and Gravio have different Client IDs as the MQTT broker will disconnect the clients if they share the same ID.

Google Cloud MQTT Example

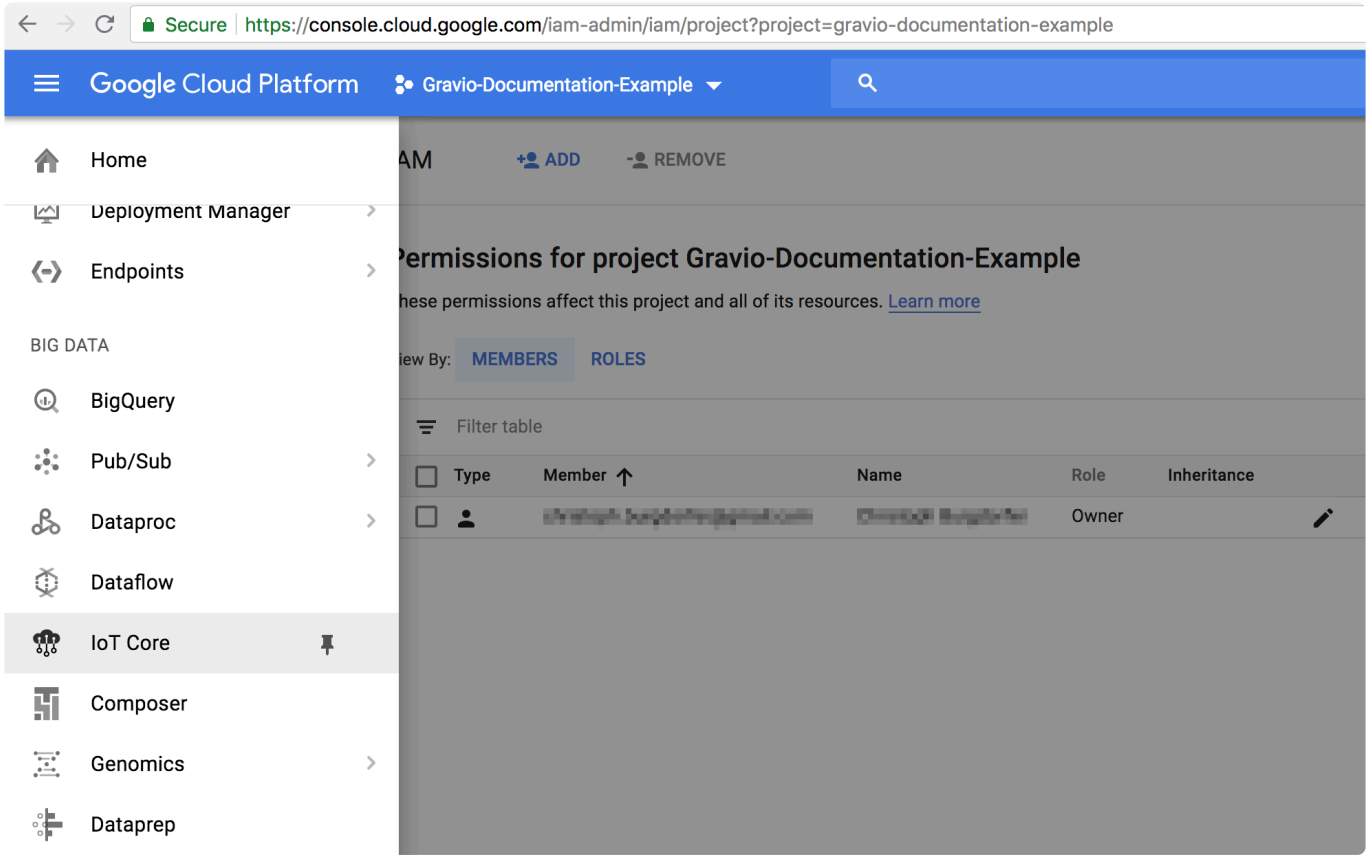
Here is an example of how to publish messages to a Google Cloud MQTT System:

Navigate to your Google Cloud console and create a project:



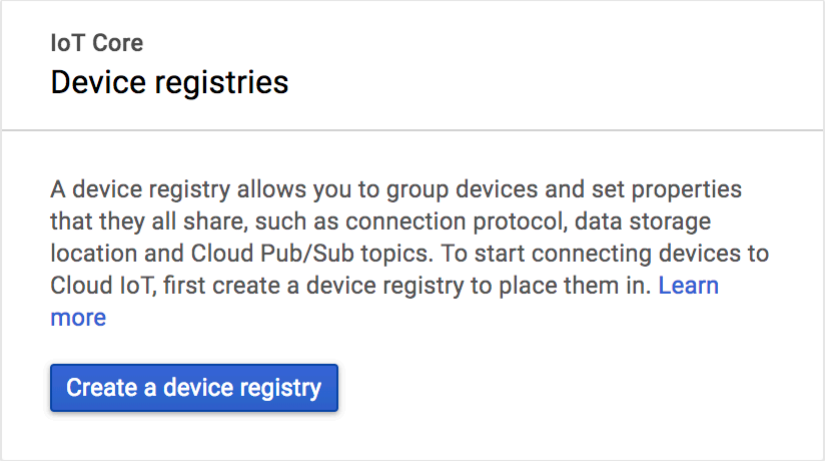


Navigate to Google Cloud Platform IoT Core

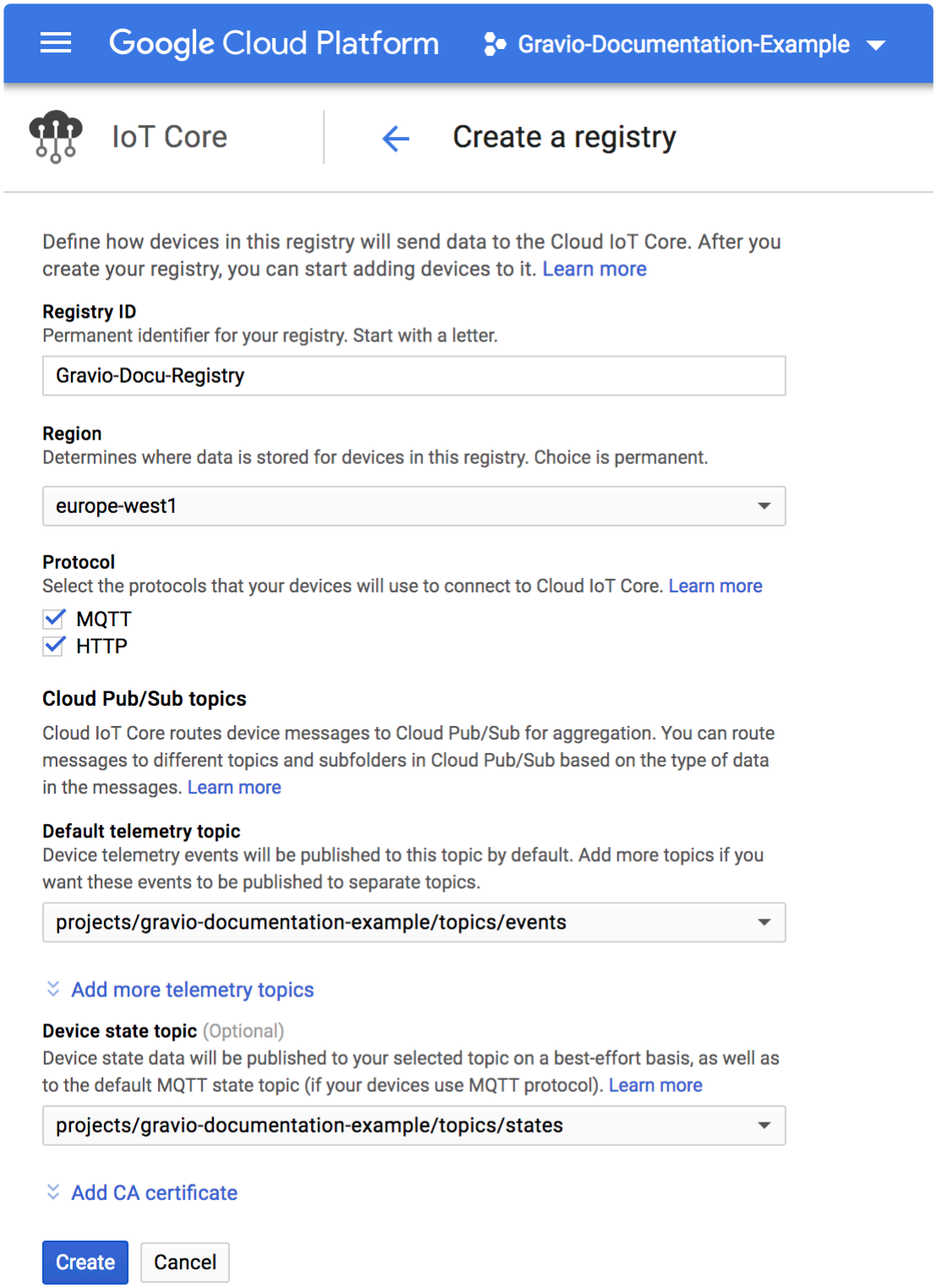


Enable the Google IoT Core API.

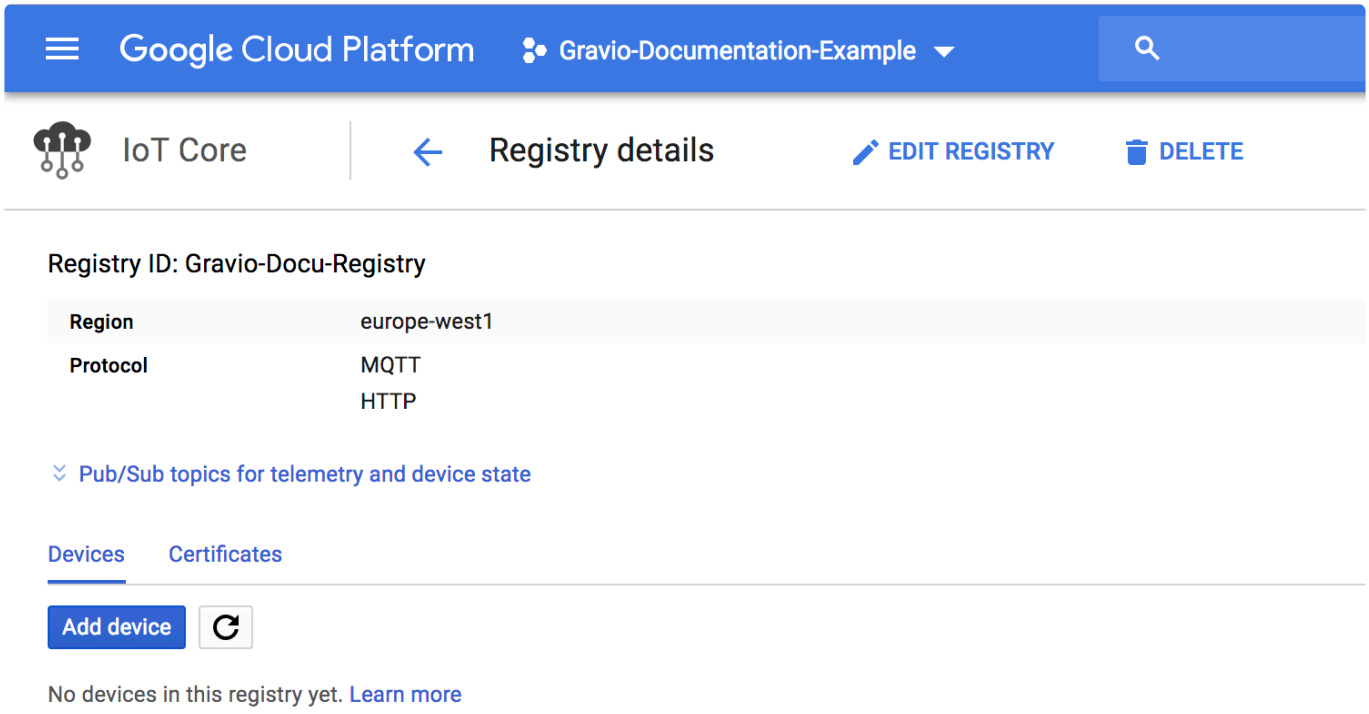
Create a Device Registry



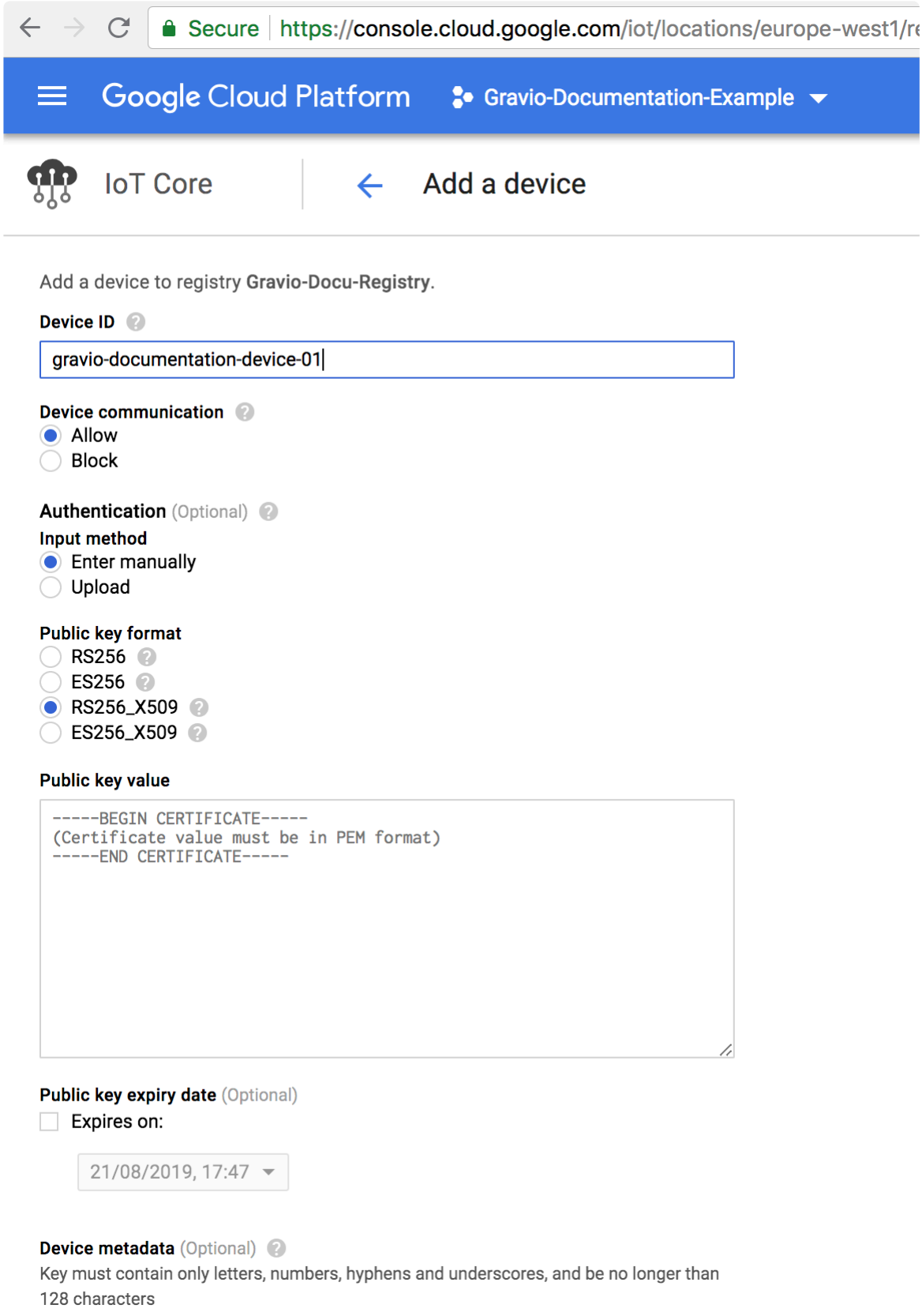
Enter the name of your registry, select the region and add the topics (here `events` and `states` for example):



Click on “Add Device” in this screen:



And you will see:



Create a certificate that you can paste into the field by entering this command in your command line:

```
openssl req -x509 -newkey rsa:2048 -keyout rsa_private.pem -nodes -out rsa_cert.pem -subj "/CN=unused"
```

This will create a file `rsa_private.pem` and a file called `rsa_cert.pem`

Copy the contents of the `rsa_cert.pem` file into the window:

Authentication (Optional)

Input method

- (●) Enter manually
- ( ) Upload

Public key format

- ( ) RS256
- ( ) ES256
- (●) RS256_X509
- ( ) ES256_X509

Public key value


```
-----BEGIN CERTIFICATE-----
MIICnjCCAYYCCQC5xZT3Rlw/gzANBgkqhkiG9w0BAQsFADARMQ8wDQYDVQQDDAZ1
bnVzZWQwHhcNMTgwODIxMTcxMDA0WhcNMTgwOTIwMTcxMDA0WjARMQ8wDQYDVQQD
DAZ1bnVzZWQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDPfNfvb44G
pb7xq4ajLhEpQn8OTtlm8hIDXpYVe0GLr9RtkceFJ25GP8+P4+8b2aikrX2DRmx9
t0Q5D0cKOQ/77cI9pBK9qBwoiXqv7YMu50JjZxPHB8nikJKh41ymqrU/vWGRlVVZ
aRcTzmHg03nAhomyCNG0vVA8jXi0YpPJiz/BuoVApobjfH17RXReuxj5IOYeFEg6
CYYbg3exS9phGXwAVJvSRb6iHHEf47advXOrkzg/A0ZUfiAr/lA4aa0+GCZ5nMMN
YL7TV+2VmR4/m2ydcWIMlD7sFe79khLJYbEsm4H14Zle+rm4MAulVmMpwa1sIpAI
FQcfS8hPAIyfAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAHDJJaHEF+1hkXJorEb0
DzQlb3NbMVQrcU4P2l+rZp6tUNgi2igb8kF5gV1t/5pds4GnIKW0UaoSaU4rjcP9
7EPIkP1m0GyTDfY9h44Id6EUnJnnIrklmQAQuZ/Mml04cE6DGyGyXvxP03Mi+DQ7
cPU4SyiKmXWU7H3Jgm6OZLk35Addz9US0WjCExYmCNsg17vXHl7epuAto4VVXB60
```


Public key expiry date (Optional)

- [ ] Expires on:

21/08/2019, 17:47

Device metadata (Optional)

Key must contain only letters, numbers, hyphens and underscores, and be no longer than 128 characters

Key	Value	✕

+ Add attribute

Add Cancel

We’re now ready to take on messages from Gravio.

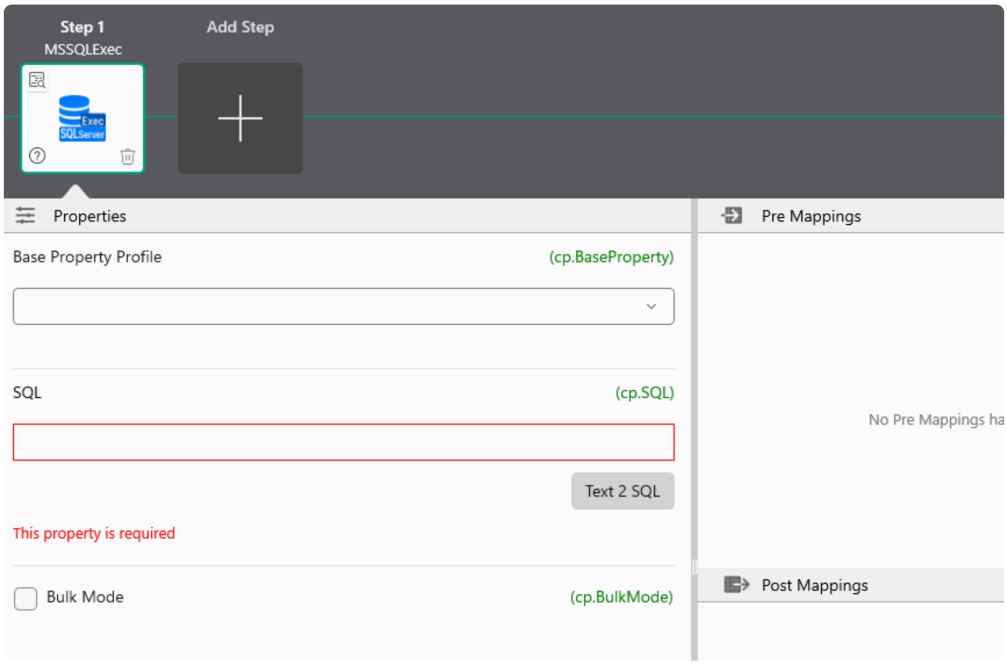
Note: The MQTT Publish component will take the variables as payload and make one MQTT publish per variable. And if you use JSON or CSV the variable name will be the key/column.

Typically you can use the CreateRecord component to inject a variable into the MQTTPublish component.

To see how to retrieve MQTT messages, please refer to the MQTT subscription section of this documentation.

6.3.9.32. SQL Server Exec

The SQL Server Exec component allows you to execute SQL statements against SQL Server to update/delete data.
It is compatible with SQL Server 2005 and above.



Output Payload

Output Payload	Description
`cv.Payload`	In addition to the input payload, it passes LastInsertId or RowsAffected by the executed SQL statement

LastInsertId is returned when the executed SQL statement was an INSERT statement and the ID generated by the DB could be obtained.
RowsAffected returns the number of records affected by the executed SQL statement. For example, it returns the number of rows deleted when the statement was a DELETE statement.

Component Properties

Property Name	Description
SQL	Specifies the SQL statement to execute
Bulk Mode	Choose whether to use bulk mode in executing the SQL statement
Bulk Size	Specifies the size when using bulk mode
Host	Specifies the host of SQL Server
Port	Specifies the port of SQL Server

Username	Specifies the username of SQL Server
Password	Specifies the password of SQL Server
Database	Specifies the database
Encrypt	Specifies whether to disable encryption (not connect with SSL/TLS), true (connect with SSL/TLS), or false (encrypt login only)
TrustServerCertificate	Select whether to trust the SQL Server's server certificate. If on, it trusts the certificate sent by SQL Server. If off, it verifies the SQL Server certificate against the trust store certificate
Root Certificate	Specifies the root certificate to use. If using AWS's Aurora or RDS, you will need to use the server certificate issued by AWS. This will be the root certificate of that server certificate

Placeholders

You can use placeholders in the SQL statement to be executed and use the value of cv.Playload in the SQL statement.

```
UPDATE table1 SET name='Updated name' WHERE id=?
```

The value of cv.Payload is used for the placeholder specified by ?. When cv.Payload contains 1, it updates the record where id is 1.
You can specify more than one placeholder.

```
UPDATE table1 SET name=? WHERE id=?
```

To specify values for more than one placeholder, specify the values in an array in cv.Payload.

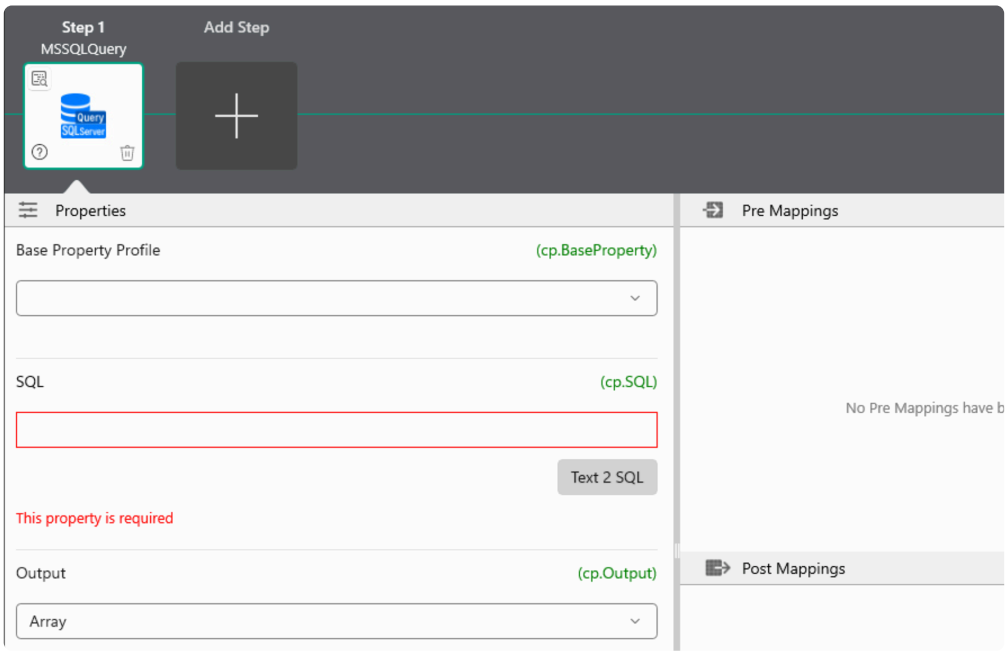
```
["First Last", 20]
```

Text2SQL

Text2SQL is a function that can generate SQL statements from Japanese entered into the prompt using the functionality of OpenAI's ChatGPT.
For how to use it, please refer to here.

6.3.9.33. SQL Server Query

The SQL Server Query component allows you to execute query statements against SQL Server to retrieve data.
It is compatible with SQL Server 2005 and above.



Output payload

Output Payload	Description
`cv.Payload`	Outputs a record in the output format specified by the property for each record that matches the condition.

Example of output in array format

When table1 table has column1 (integer type) and column2 (string type), specifying `SELECT * FROM table1` in the SQL property and “array” in the output format property, the output payload is generated as follows.

```
Output Example
[1, "abc"]
[2, "def"]
[3, "hij"].
```

Example of output in object format

When table1 table has column1 (integer type) and column2 (string type), specifying `SELECT * FROM table1` in the SQL property and “object format” in the output format property, the output payload is generated as follows.

```
Output Example
{"column1": 1, "column2": "abc"}
{"column1": 2, "column2": "def"}
{"column1": 3, "column2": "hij"}
```

Component properties

|_. Property Name |_. Description | |_. SQL | Specify SQL statement to execute | |_. Output Format | Select array or object format | |_. Host | Specify the SQL Server host. | |_. Port | Specify the SQL Server port | |_. User Name | Specify the SQL Server user name. | |_. Password | Specify the SQL Server password. | |_. Database | Specify the database. | |_. Encrypt | Specify communication encryption from disable (no SSL/TLS connection), true (SSL/TLS connection), and false (encryption for login only) | |_. TrustServerCertificate | Select whether to trust SQL Server’s server certificate. If on, trusts the certificate sent by SQL Server. If Off, validates SQL Server certificates against trust store certificates | |_. Root Certificate | Specify the root certificate to use; if you are using AWS Aurora or RDS, you must use a server certificate issued by AWS. This will be the root certificate for that server certificate. |

Placeholder

You can use a placeholder in the SQL statement to be executed to use the value of cv.Playload in the SQL statement.

```
SELECT * FROM table1 WHERE id=?
```

? The value of cv.Payload is used for the placeholder specified in cv. When cv.Payload contains 1, the record with id 1 can be retrieved.
More than one placeholder can be specified.

```
SELECT * FROM table1 WHERE name=? AND age=?
```

To specify values for more than one placeholder, specify the values in an array in cv.

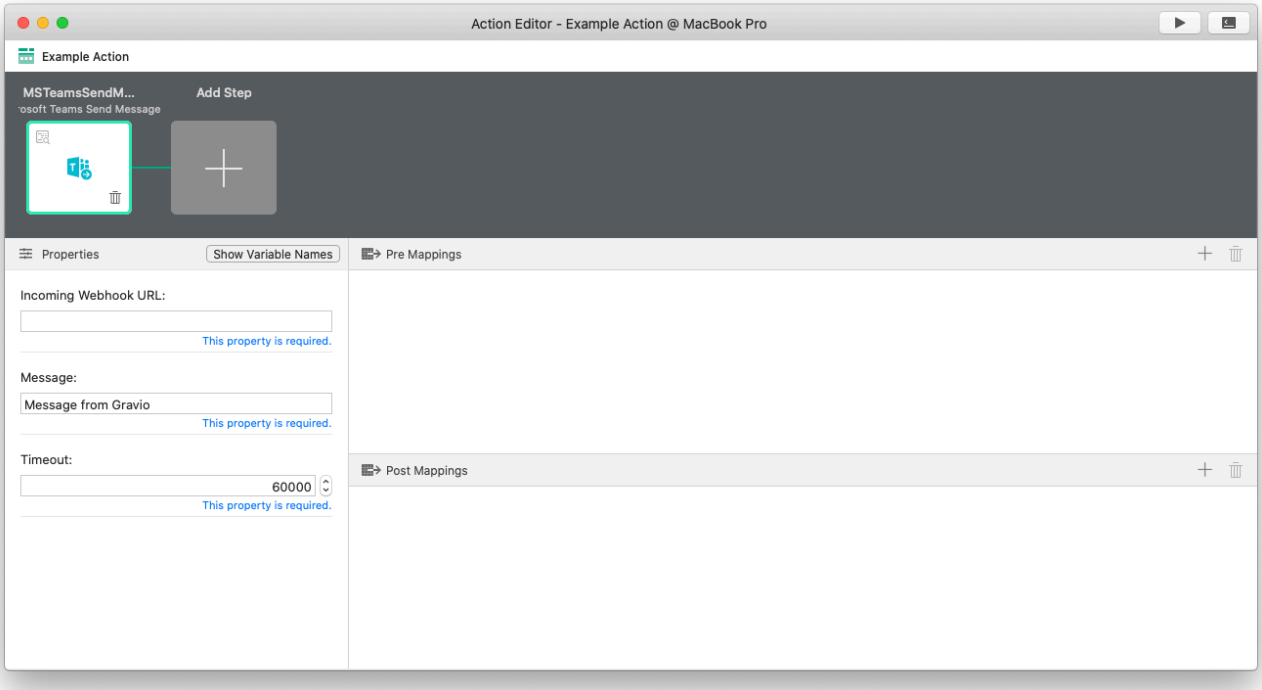
```
["First Last", 20]
```

Text2SQL

Text2SQL is a function that allows you to generate SQL statements from the Japanese you enter in the prompt for the data you wish to retrieve using OpenAI's ChatGPT functionality.
Please refer to here for how to use it.

6.3.9.34. MSTeamsPostMessage

The MSTeamsPostMessage action component can use the Microsoft Teams API to post a message to the channel specified in the component's properties.



How to get the Incoming Webhook URL:

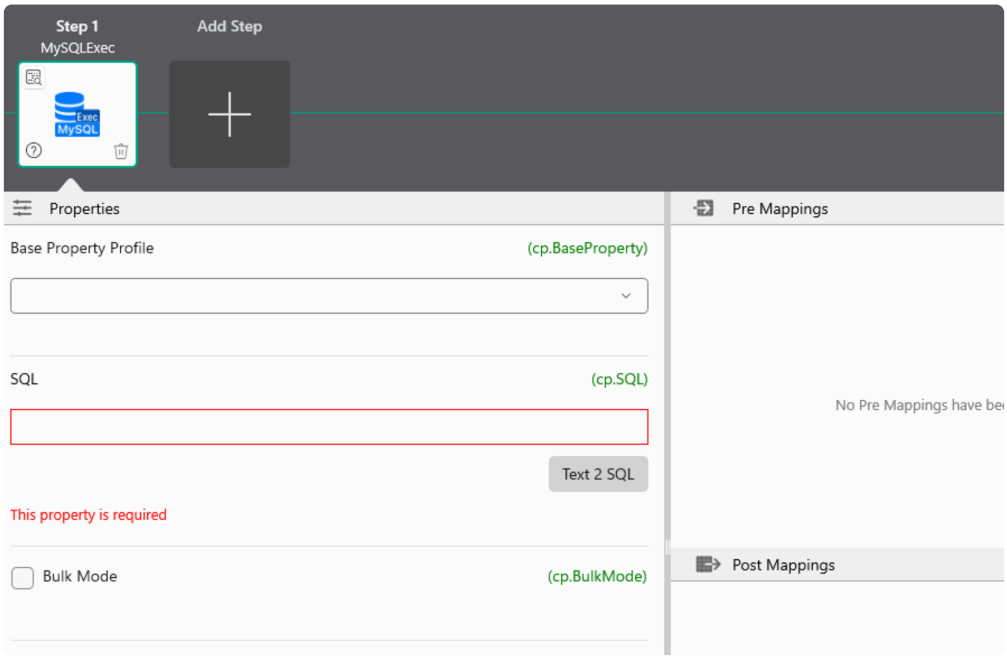
- Select "Connector" from the "More Options" menu in the channel where you want to post your Teams message.
- Select "Incoming Webhook" from the connector settings screen and press the "Configure" button.
- Enter the name of the Incoming Webhook and press the "Create" button.
- When an Incoming Webhook is created, a URL is automatically created.
- Specify this URL in the component's Incoming Webhook URL property.
- In the Incoming Webhook URL field, enter the URL created in Teams.

Component Properties

Name	Required	Description
Incoming Webhook URL	True	The URL provided by your MS Team API account.
Message	True	Message to be sent to the API.
Timeout	True	API timeout – typically you would leave it on 30s.

6.3.9.35. MySQL Exec

The MySQL Exec component allows you to execute SQL statements against MySQL to update/delete data.
It is compatible with MySQL 5.6 and above.



Output Payload

Output Payload	Description
`cv.Payload`	In addition to the input payload, it passes LastInsertId or RowsAffected by the executed SQL statement

LastInsertId is returned when the executed SQL statement was an INSERT statement and the ID generated by the DB could be obtained.
RowsAffected returns the number of records affected by the executed SQL statement. For example, it returns the number of rows deleted when the statement was a DELETE statement.

Component Properties

Property Name	Description
SQL	Specifies the SQL statement to execute
Bulk Mode	Choose whether to use bulk mode in executing the SQL statement
Bulk Size	Specifies the size when using bulk mode

Host	Specifies the host of MySQL
Port	Specifies the port of MySQL
Username	Specifies the username of MySQL
Password	Specifies the password of MySQL
Database	Specifies the database
Timezone	Specifies the timezone of the database
TLS	Specifies whether to encrypt communication, options are: true (connect with SSL/TLS), false (do not connect with SSL/TLS), skip-verify (connect with SSL/TLS but do not verify the server certificate), preferred (try to connect with SSL/TLS first, if unable to connect, try to connect without SSL/TLS)
Root Certificate	Specifies the root certificate to use. If using AWS's Aurora or RDS, you will need to use the server certificate issued by AWS. This will be the root certificate of that server certificate

Placeholders

You can use placeholders in the SQL statement to be executed and use the value of cv.Playload in the SQL statement.

```
UPDATE table1 SET name='Updated name' WHERE id=?
```

The value of cv.Payload is used for the placeholder specified by ?. When cv.Payload contains 1, it updates the record where id is 1.
You can specify more than one placeholder.

```
UPDATE table1 SET name=? WHERE id=?
```

To specify values for more than one placeholder, specify the values in an array in cv.Payload.

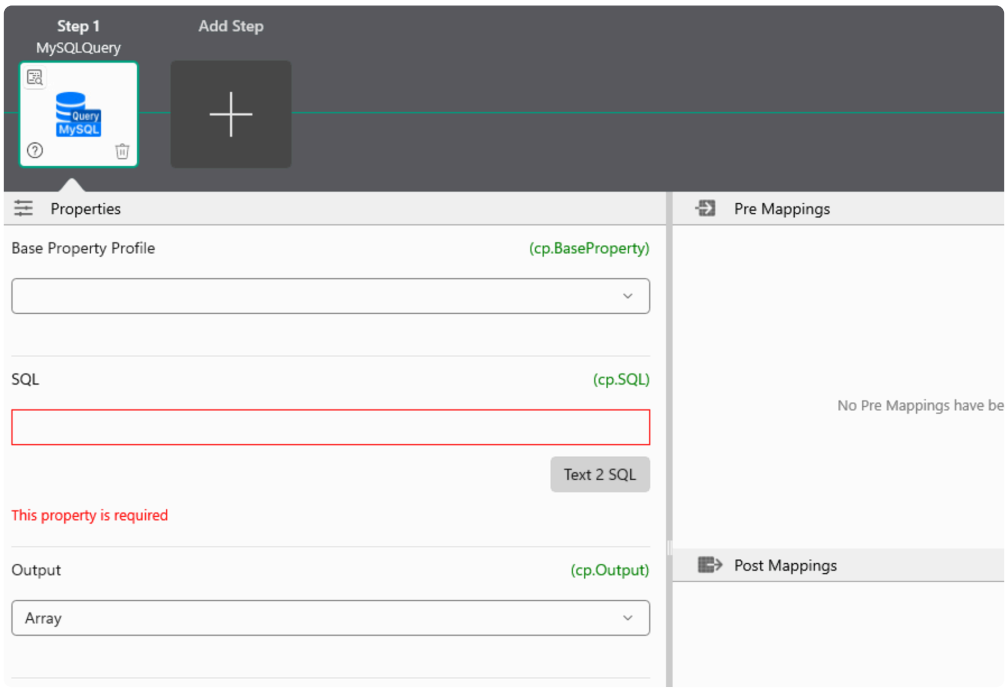
```
["First Last", 20]
```

Text2SQL

Text2SQL is a function that can generate SQL statements from Japanese entered into the prompt using the functionality of OpenAI's ChatGPT.
For how to use it, please refer to here.

6.3.9.36. MySQL Query

The MySQL Query component allows you to execute query statements against MySQL to retrieve data. It is compatible with MySQL 5.6 and above.



Output Payload

Output Payload	Description
`cv.Payload`	Outputs records that match the conditions in the output format specified by the property

Output example in array format

When table1 has column1 (integer type) and column2 (string type), if you specify `SELECT * FROM table1` in the SQL property and specify 'Array' in the output format property, the output payload will be generated as follows:

```
Output Example
[1, "abc"]
[2, "def"]
[3, "hij"]

p.

Output example in object format

When table1 has column1 (integer type) and column2 (string type), if you speci
fy `SELECT * FROM table1` in the SQL property and specify 'Object Format' in t
```

```
he output format property, the output payload will be generated as follows:
```

```
Output Example
{"column1": 1, "column2": "abc"}
{"column1": 2, "column2": "def"}
{"column1": 3, "column2": "hij"}

p.
```

Component Properties

Property Name	Description
SQL	Specifies the SQL statement to execute
Output Format	Choose between array format or object format
Host	Specifies the host of MySQL
Port	Specifies the port of MySQL
Username	Specifies the username of MySQL
Password	Specifies the password of MySQL
Database	Specifies the database
Timezone	Specifies the timezone of the database
TLS	Specifies whether to encrypt communication, options are: true (connect with SSL/TLS), false (do not connect with SSL/TLS), skip-verify (connect with SSL/TLS but do not verify the server certificate), preferred (try to connect with SSL/TLS first, if unable to connect, try to connect without SSL/TLS)
Root Certificate	Specifies the root certificate to use. If using AWS's Aurora or RDS, you will need to use the server certificate issued by AWS. This will be the root certificate of that server certificate

Placeholders

You can use placeholders in the SQL statement to be executed and use the value of cv.Playload in the SQL statement.

```
SELECT * FROM table1 WHERE id=?

p.

The value of cv.Payload is used for the placeholder specified by ?. When cv.Pa
yload contains 1, it retrieves the record where id is 1.
```

```
You can specify more than one placeholder.
```

```
SELECT * FROM table1 WHERE name=? AND age=?

p.

To specify values for more than one placeholder, specify the values in an arra
y in cv.Payload.
```

```
["First Last", 20]

p.
```

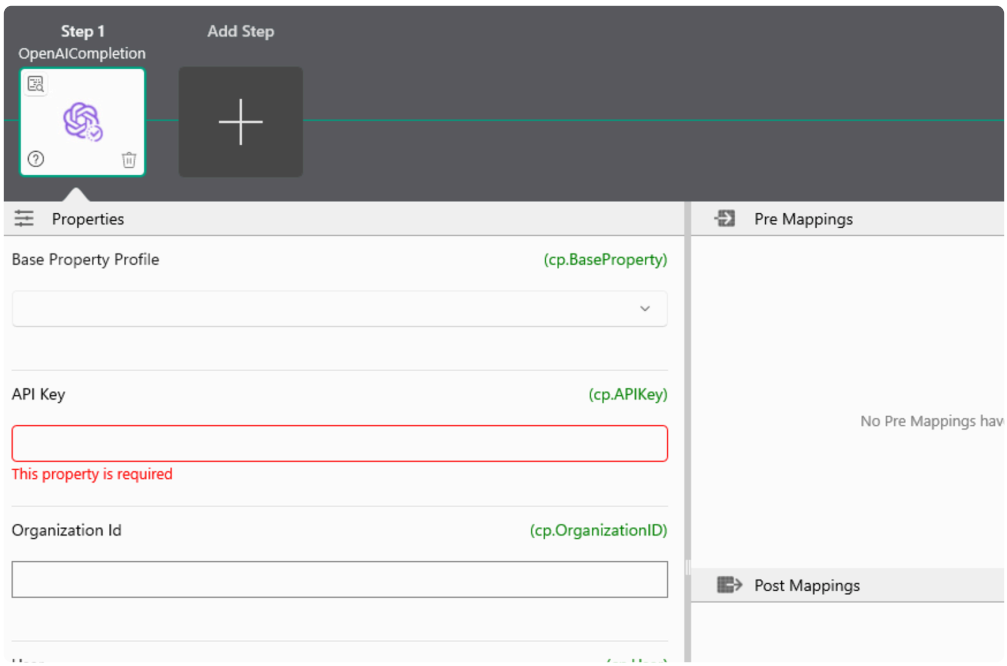
Text2SQL

Text2SQL is a function that can generate SQL statements from Japanese entered into the prompt using the functionality of OpenAI's ChatGPT.
For how to use it, please refer to here.

6.3.9.37. OpenAICompletion

The OpenAICompletion component is a component that integrates with the AI features provided by OpenAI. OpenAICompletion returns completions for given prompts. For chat-based functions, please use the OpenAIChat Component.

- To use this OpenAICompletion component, you need to obtain an API Key in advance to call OpenAI's AI function. For how to get an API Key on OpenAI's site, please refer to here.



Component Properties

Property Name	Description	Example
Basic Properties	Specify the basic properties of the 'OpenAI' category	
API Key	API Key *1	
Organization Id	Organization ID *1	
User	Username	
Model	Choose or manually input the model name to use	`Text Davinci 003` `Text Davinci 002` `Text Curie 001` `Text Babbage 001` `Text Ada 001`
Temperature	Specify the sampling temperature between 0 and 1	
Prompt	Prompt	

Suffix	Suffix	
Max Tokens	The maximum number of tokens to be generated	
Top-P	Specify nucleus sampling between 0 and 1. Default is 1	
N	The number of answers generated per prompt	
Log Probabilities	Include the log probabilities of the most likely token and the selected token	
Echo	Add the prompt to the answer	
Stop	String to stop token generation	
Presence Penalty	Specify whether to use a word that has already appeared again, from -2.0 to 2.0. Default is 0	
Frequency Penalty	Specify whether to give a penalty the more often it appears, from -2.0 to 2.0. Default is 0	
Best Of	Control the number of candidate answers. You specify the number to return, but best_of must be greater than N	
Logit Bias	Parameter to change the likelihood of a specified token appearing	

- Please refer to the OpenAI API Reference for details on each OpenAI-related property.
- To use the OpenAICompletion component, you need to obtain an API key. For how to get an API key, please refer to here.

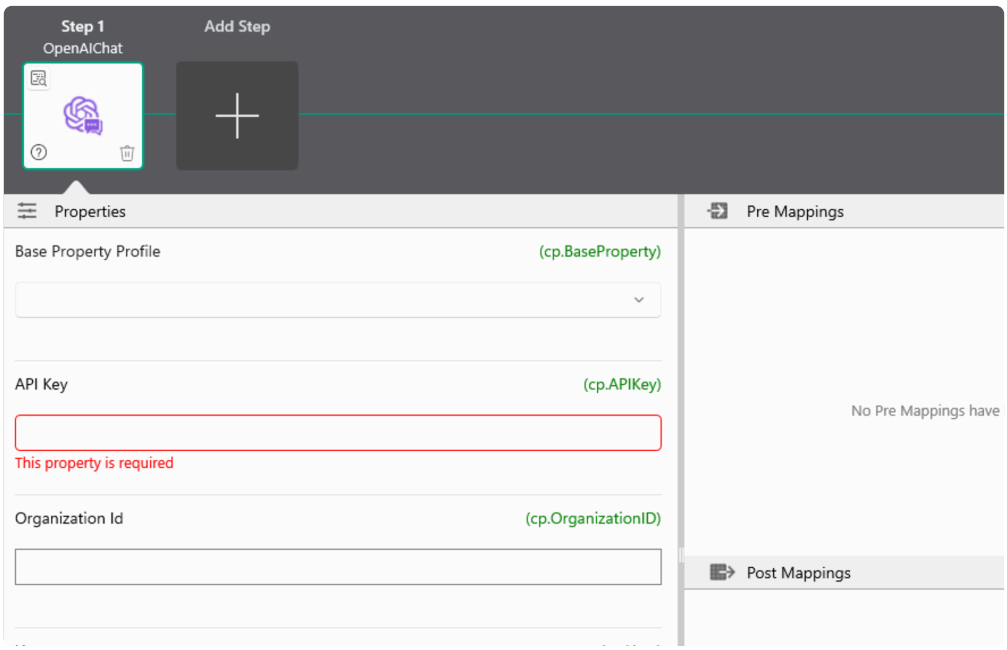
Output Payload

Output Payload	Description
`cv.Payload`	It stores the response from the OpenAI Completion API as is

6.3.9.38. OpenAIChat

The OpenAIChat component is a component that collaborates with the AI functions provided by OpenAI. OpenAIChat uses a language model specialized in chat completion. It is designed to simplify conversations involving multiple interactions.

※To use this OpenAIChat component, you need to obtain an API Key in advance to call OpenAI's AI functions. For how to obtain an API Key on the OpenAI site, please refer to here.



Component Properties

Property Name	Description	Example
Basic Properties	Specifies the basic properties of the 'OpenAI' category	
API Key	API Key *1	
Organization Id	Organization ID *1	
User	User name	
Model	Select or manually enter the model name to be used. The models that can be used vary depending on the contract with OpenAI.	`GPT 3.5 Turbo` `GPT 3.5 Turbo 16k` `GPT 4` `GPT 4 32k`
System Message	Role Setting	
User Message Sample Input	Specifies examples of questions from the user	

Assistant Message Sample Input	Specifies examples of answers	
User Message	Question from the user	
Temperature	Specifies the sampling temperature between 0 and 1	
Max Tokens	Maximum number of tokens in the response	
Top-P	Specifies nucleus sampling between 0 and 1. The default is 1	
N	Number of responses generated per prompt	
Log Probabilities	Includes the log probability of the most likely token and the selected token	
Stop	String to stop token generation	
Presence Penalty	Specifies whether to reuse words that have already appeared, from -2.0 to 2.0. The default is 0	
Frequency Penalty	The more often it appears, the more penalty it gets, specified from -2.0 to 2.0. The default is 0	
Logit Bias	Parameter to change the likelihood that the specified token will appear	

※ For details of each property related to OpenAI, please refer to OpenAI API Reference.
※1 To use the OpenAIChat component, you need to obtain an API key. For how to obtain an API key, please refer to here.

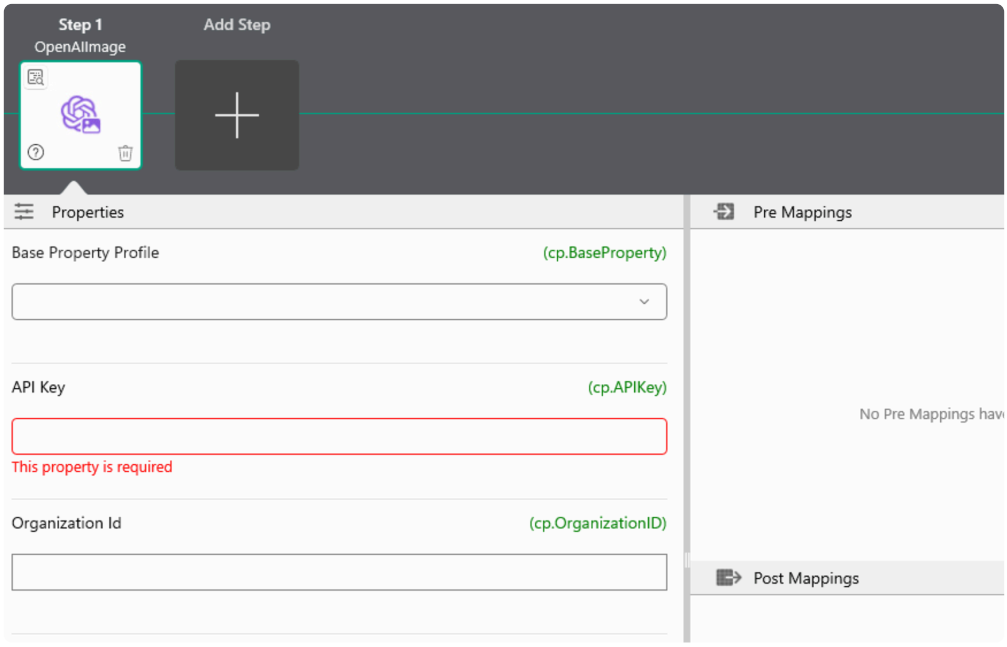
Output Payload

Output Payload	Description
`cv.Payload`	Stores the response from the OpenAI Chat API as is

6.3.9.39. OpenAIImage

The OpenAIImage component is a component that collaborates with the AI functions provided by OpenAI. OpenAIImage can invoke a function to generate images from the entered prompts.

※To use this OpenAIImage component, you need to obtain an API Key in advance to call OpenAI's AI functions. For how to obtain an API Key on the OpenAI site, please refer to here.



Component Properties

Property Name	Description	Example
Basic Properties	Specifies the basic properties of the 'OpenAI' category	
API Key	API Key *1	
Organization Id	Organization ID *1	
User	User name	
Operation	Select the operation from Image Creation, Image Editing, Similar Image Generation	`Create Image` `Edit Image` `Make Image Variation`
Prompt	Prompt (only for Image Creation or Image Editing operations)	
Image File	Image PNG file (only for Image Editing or Similar Image Generation operations)	
Mask File	Mask PNG file less than 4MB, use an image of the same size as	

	the Image File (only for Image Editing operations)	
n	Number of responses generated per prompt	
Image Size	Size of the image to be generated	`1024px by 1024px` `512px by 512px` `256px by 256px`
Response Format	Generated data format File URL Base64	`File` `URL` `Base64`
Filename	Output filename (only for File format data)	
File Overwrite	Overwrite output file (only for File format data) If you do not overwrite the file and a file with the same name exists, it will be saved with “-N” appended to the end	

※ For details of each property related to OpenAI, please refer to OpenAI API Reference.
※1 To use the OpenAIImage component, you need to obtain an API key. For how to obtain an API key, please refer to here.

Operation Property

・When specifying Create Image
Use a square image less than 4MB for the Image File

・When specifying Edit Image
Use a square image less than 4MB for the Image File, if specifying a Mask File, use an image of the same size as the Image File.

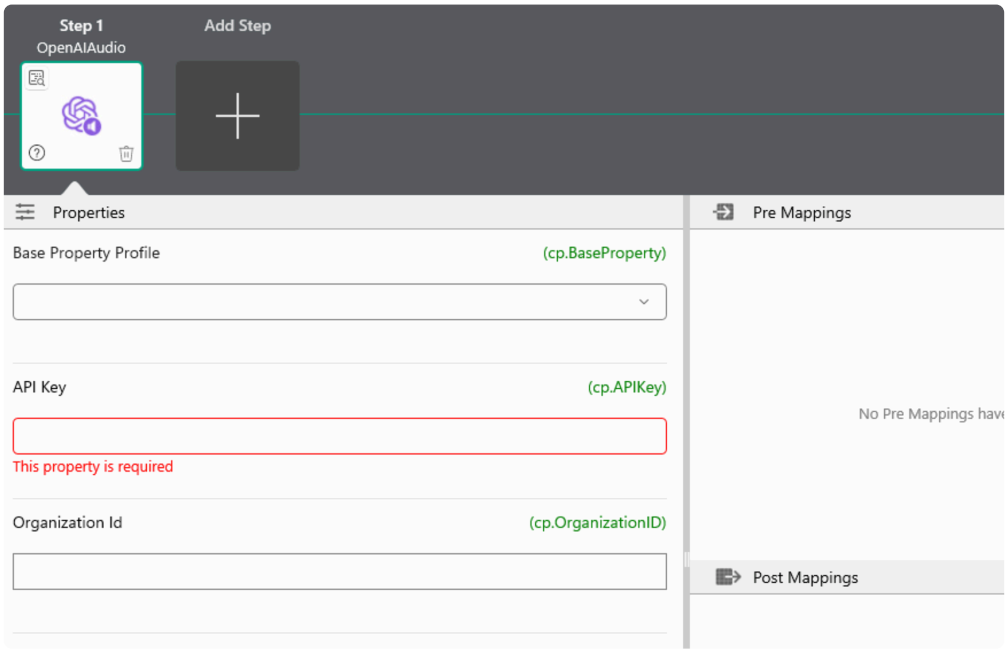
Output Payload

Output Payload	Description
`cv.Payload`	If the Response Format property is a file, the filepath of the written file is stored, if URL, the URL is stored, if Base64, the generated data encoded in Base64 is stored

6.3.9.40. OpenAIAudio

OpenAIAudio component is a component that works with OpenAI's AI functionality.

To use the OpenAIAudio component, you need to obtain an API Key in advance in order to call OpenAI's AI functions. Please refer to here for how to obtain an API Key.



Component properties

Property Name	Description	Example
Basic Properties	「Specify the basic properties of the "OpenAI" category	
API Key	API Key *1	
Organization Id	Organization ID *1	
User	User Name	
Operation	Choose a process from Transcription Translation	`Transcribe` `Translation`
Model	Select the model name you want to use or enter it manually	`Whisper-1`
Audio File	Sound filename	
Language	Language of speech Specify in ISO-639-1 format. Default is English.	`ja` `en`
Prompt	Prompt	
Temperature	Specify sampling temperature from 0 to 1.	

Response Format	Select output data format from JSON Detailed JSON Text SRT File VTT File	`JSON` `Verbose JSON` `Text` `srt` `vtt`

- Refer to OpenAI API Reference for details of each OpenAI related property.
  1 To use OpenAIAudio components, you need to obtain an API key, see here for how to obtain an API key.

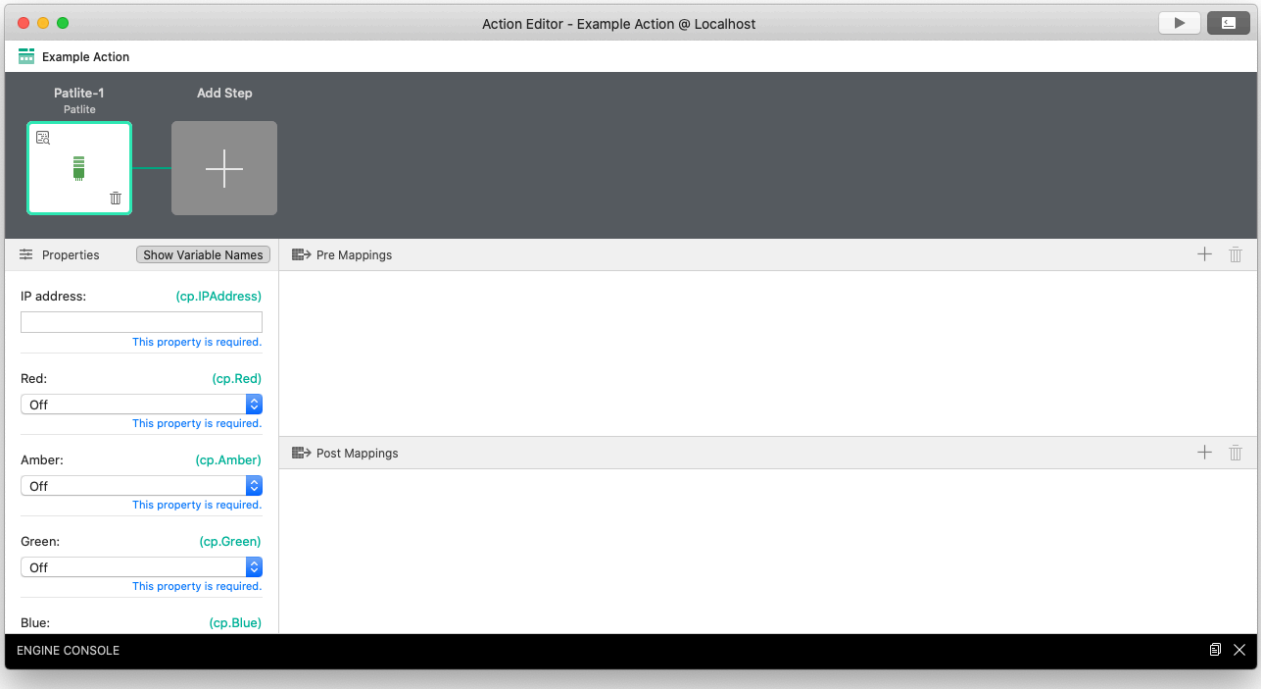
Output payload

Output payload	Description
`cv.Payload`	Response Format property Stores the contents specified in the output data format.

6.3.9.41. PATLITE

The Patlite Action Component is to control a PATLITE network device including audio. The watchtower must support HTTP commands. For more information, please visit the Patlite website.
https://www.patlite.co.jp/

PATLITEs are useful to control the display according to the information from the sensors.



Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

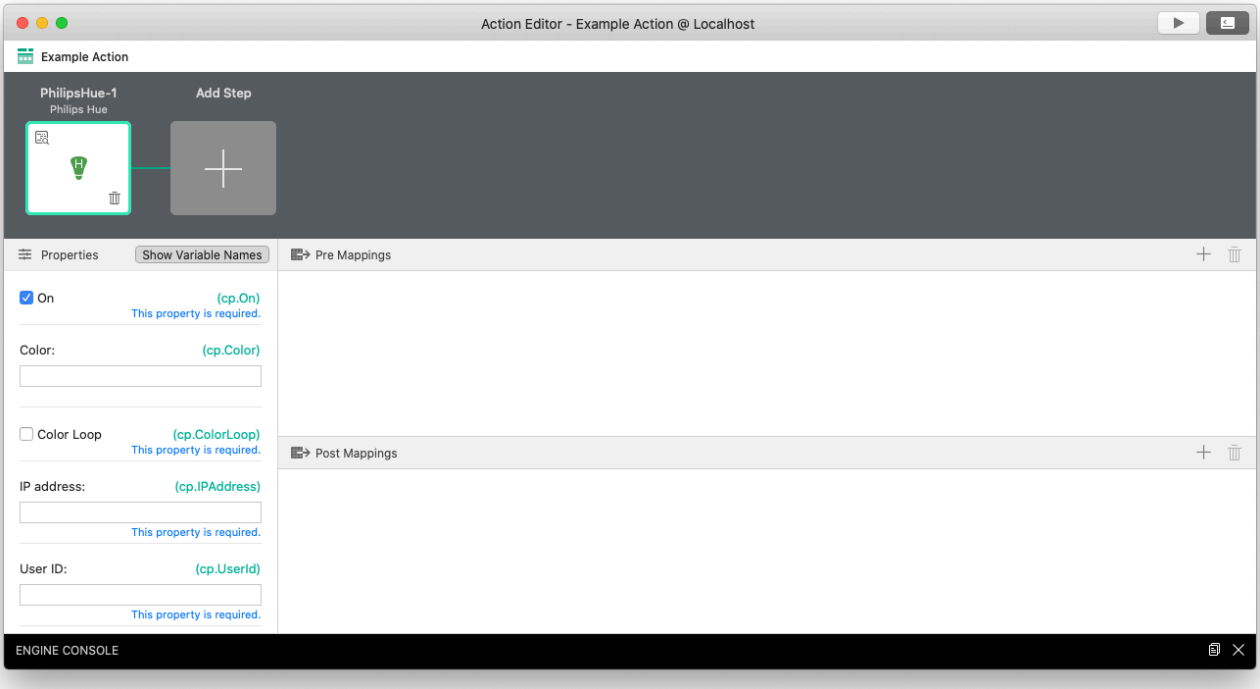
Component Properties

Name	Required	Description
IP address	True	The IP address of the watchtower
Red	True	Set to either On/Off/Blink1/Blink2/No Change. [The values to set for cp.Red are “off”, “on”, “blink-1”, “blink-2”, “no-change”].
Amber	True	Set to either On/Off/Blink1/Blink2/No Change [The values to set for cp. Amber are “off”, “on”, “blink-1”, “blink-2”, “no-change”].
Green	True	Set to either On/Off/Blink1/Blink2/No Change [The values to set for cp. Green are “off”, “on”, “blink-1”, “blink-2”, “no-change”].
Blue	True	Set to either On/Off/Blink1/Blink2/No Change [The values to set for cp.Blue are

		“off”, “on”, “blink-1”, “blink-2”, “no-change”].
White	True	Set to either On/Off/Blink1/Blink2/No Change [The values to set for cp.White are “off”, “on”, “blink-1”, “blink-2”, “no-change”].
Play Sound	True	Play Sound
Sound Channel	True	Specify the channel to be played back using numbers. For details, see the manual for the device.
Repeat	True	Specifies the number of additional repetitions. If you don’t want to repeat it, leave it blank.
Timeout	False	Timeout (ms)

6.3.9.42. PhilipsHue

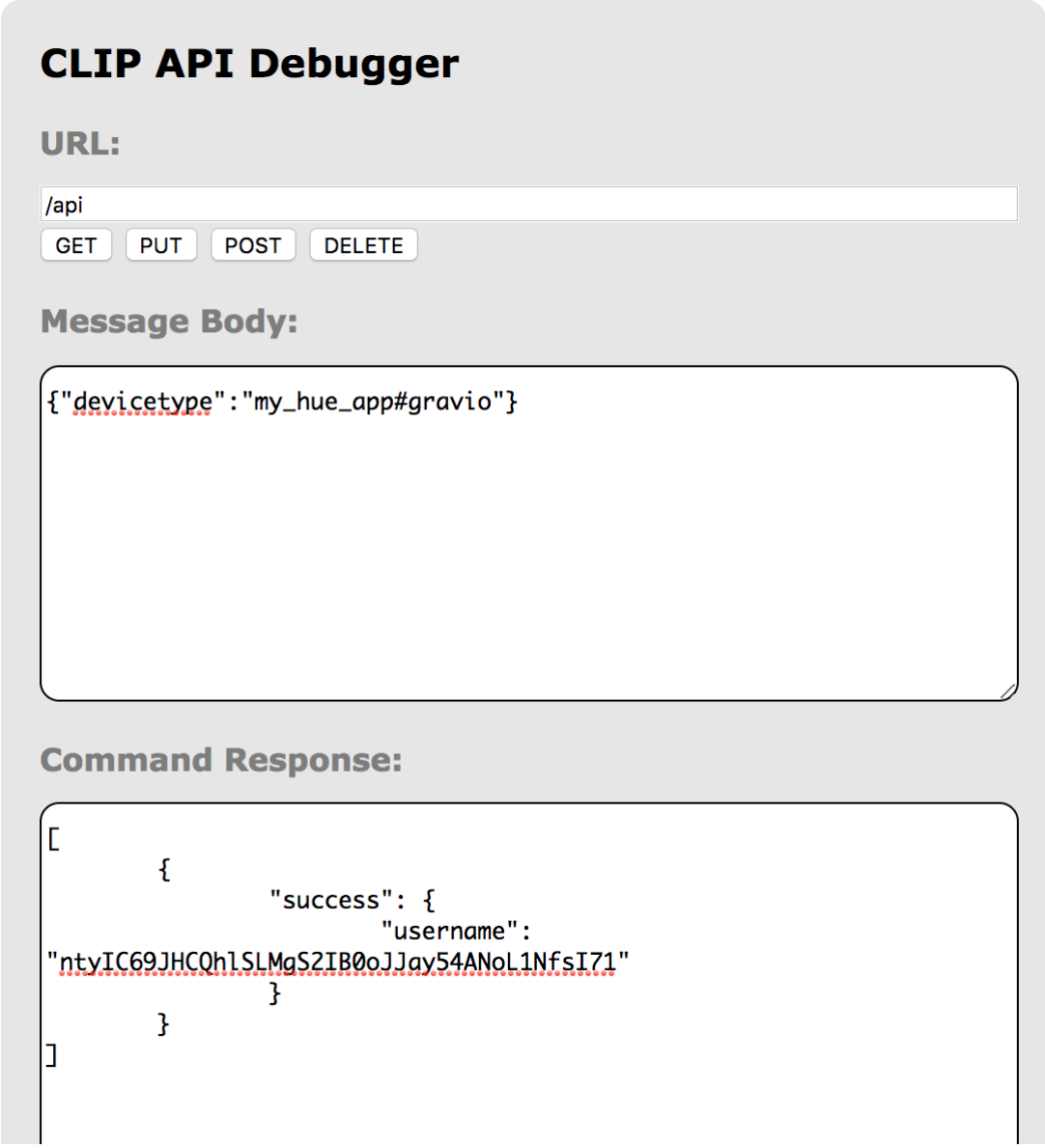
The Philips Hue Action Component is to send commands to a Philips Hue light device. You need to specify the color and opacity and set the IP address, user ID, and light ID.



The IP address can be found by going to https://discovery.meethue.com/.

With this information, the debugger API simulator can be found at `http://<your_ip>/debug/clip.html`

Before you send the `POST` command to the API using JSON, you need to get the user ID, which you can only get if you “pair” the device by pressing the bridge button once. `{"devicetype": "my_hue_app#gravio"}`



For detailed instructions on how to use the debugger, see here: https://developers.meethue.com/documentation/getting-started

Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

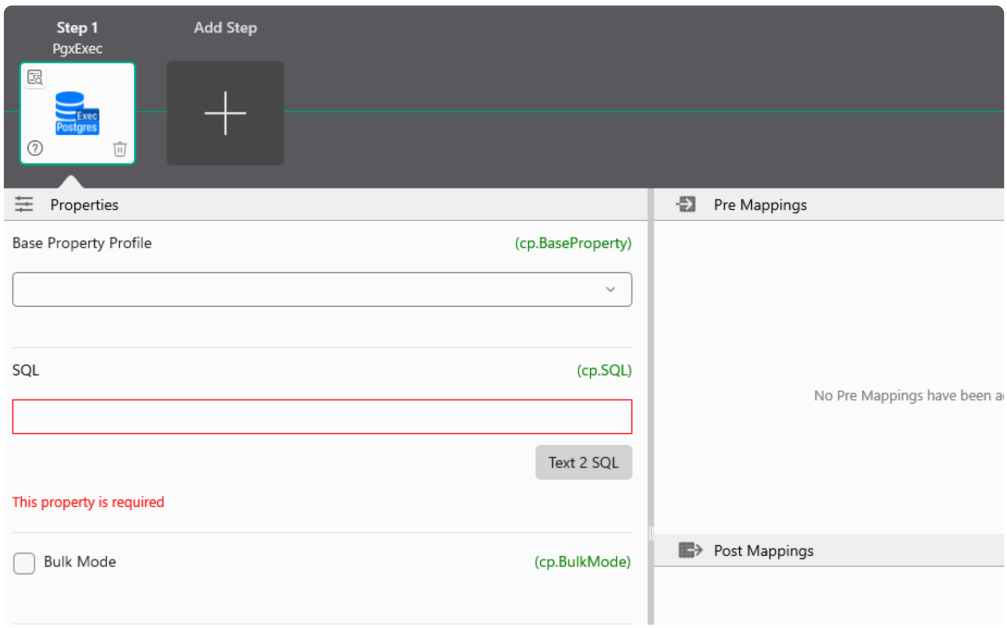
Component Properties

Name	Required	Description
On	True	Specify on/off. The value specified for cp.On is “true” or “false”.
Color	False	R,G,B,Opacity in hexadecimal, for example AABBCCDD
Color Loop	True	Switch colors in regular intervals

IP address	True	IP address of Philips Hue bridge
User ID	True	Philips Hue User ID
Light ID	False	Specify the Philips Hue light IDs which is a number starting with 1. All lights if empty.

6.3.9.43. PostgreSQL Exec

The PostgreSQL Execc component allows you to execute SQL statements against PostgreSQL to update/delete data.
It supports PostgreSQL 11 and above.



Output payload

Output Payload	Description
`cv.Payload`	Pass LastInsertId or RowsAffected depending on the SQL statement executed in addition to the input payload

LastInsertId is returned if the SQL statement executed was an INSERT statement and the ID generated by the DB can be retrieved.
RowsAffected returns the number of records affected by the SQL statement executed. For example, if it was a DELETE statement, the number of rows deleted is returned.

h3. Component property

Property Name	Description
SQL	Specify SQL statement to execute
Bulk Mode	Select whether to use bulk mode for SQL statement execution

Bulk Size	Specify the size if bulk mode is used
Host	Specify PostgreSQL host.
Port	Specify the PostgreSQL port.
Username	Specify a PostgreSQL username.
Password	Specify the PostgreSQL password.
Database	Specifies a database.
Timezone	Specify the timezone for the database
SSLMode	The encryption of the communication can be disable (no SSL/TLS connection), allow (first try a connection without SSL/TLS connection and if the connection fails, then try a SSL/TLS connection), prefer (first try a connection with SSL/TLS connection and if the connection fails, then try a connection without SSL/TLS connection), or require (try a connection with SSL/TLS connection). connection), prefer (first try an SSL/TLS connection and if it fails to connect, then try a non-SSL/TLS connection), require (make an SSL/TLS connection. (makes an SSL/TLS connection and verifies the server certificate if specified, but not if not specified), verify-ca (makes an SSL/TLS connection and verifies the server certificate), verify-full (verifies the SSL/TLS connection and the server certificate, and also verifies the server name)
Root Certificate	Specify the root certificate to use; if you are using AWS Aurora or RDS, you must use a server certificate issued by AWS. This will be the root certificate for that server certificate.

Placeholder

You can use a placeholder in the SQL statement to be executed to use the value of `cv.Playload` in the SQL statement.

```
UPDATE table1 SET name='update name' WHERE id=?
```

The value of `cv.Payload` is used for the placeholder specified in `cv.` When `cv.Payload` contains `1`, the record with id `1` is updated.

More than one placeholder may be specified. For example:

```
UPDATE table1 SET name=$1 WHERE id=$2
```

To specify values for more than one placeholder, specify the values in an array in cv.

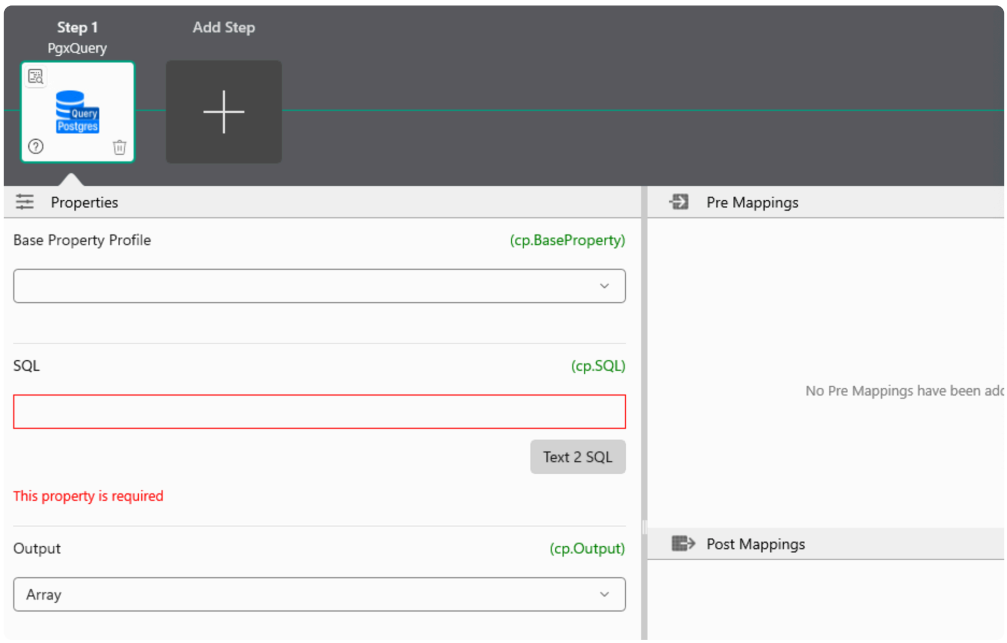
```
["First Last", 20]
```

Text2SQL

Text2SQL is a function that allows you to generate SQL statements from the Japanese you enter in the prompt for the data you wish to retrieve using OpenAI's ChatGPT functionality.
Please refer to here for how to use it.

6.3.9.44. PostgreSQL Query

The PostgreSQL Query component allows you to execute query statements against PostgreSQL to retrieve data.
It supports PostgreSQL 11 and above.



Output payload

Output Payload	Description
`cv.Payload`	Outputs a record in the output format specified by the property for each record that matches the condition.

Example of output in array format

When table1 table has column1 (integer type) and column2 (string type), specifying `SELECT * FROM table1` in the SQL property and “array” in the output format property, the output payload is generated as follows.

```
Output Example
[1, "abc"]
[2, "def"]
[3, "hij"].
```

Example of output in object format

When table1 table has column1 (integer type) and column2 (string type), specifying `SELECT * FROM table1` in the SQL property and “object format” in the output format property, the output payload is generated as follows.

```
Output example
{"column1": 1, "column2": "abc"}
{"column1": 2, "column2": "def"}
{"column1": 3, "column2": "hij"}
```

Component properties

Property Name	Description
SQL	Specify SQL statement to execute
Output Format	Select array or object format
Host	Specify PostgreSQL host.
Port	Specify the PostgreSQL port.
Username	Specify a PostgreSQL username.
Password	Specify the PostgreSQL password.
Database	Specifies a database.
Timezone	Specify the timezone for the database
SSLMode	The encryption of the communication can be disable (no SSL/TLS connection), allow (first try a connection without SSL/TLS connection and if the connection fails, then try a SSL/TLS connection), prefer (first try a connection with SSL/TLS connection and if the connection fails, then try a connection without SSL/TLS connection), or require (try a connection with SSL/TLS connection). connection), prefer (first try an SSL/TLS connection and if it fails to connect, then try a non-SSL/TLS connection), require (make an SSL/TLS connection. (makes an SSL/TLS connection and verifies the server certificate if specified, but not if not specified), verify-ca (makes an SSL/TLS connection and verifies the server certificate), verify-full (verifies the SSL/TLS connection and the server certificate, and also verifies the server name)
Root Certificate	Specify the root certificate to use; if you are using AWS Aurora or RDS, you must use a server certificate issued by AWS. This will be the root certificate for that server certificate.

Placeholder

You can use a placeholder in the SQL statement to be executed to use the value of cv.Playload in the SQL statement.


```
SELECT * FROM table1 WHERE id=?
```

? The value of cv.Payload is used for the placeholder specified in cv. When cv.Payload contains 1, the record with id 1 can be retrieved.
More than one placeholder can be specified.

```
SELECT * FROM table1 WHERE name=? AND age=?
```

To specify values for more than one placeholder, specify the values in an array in cv.

```
["First Last", 20]
```

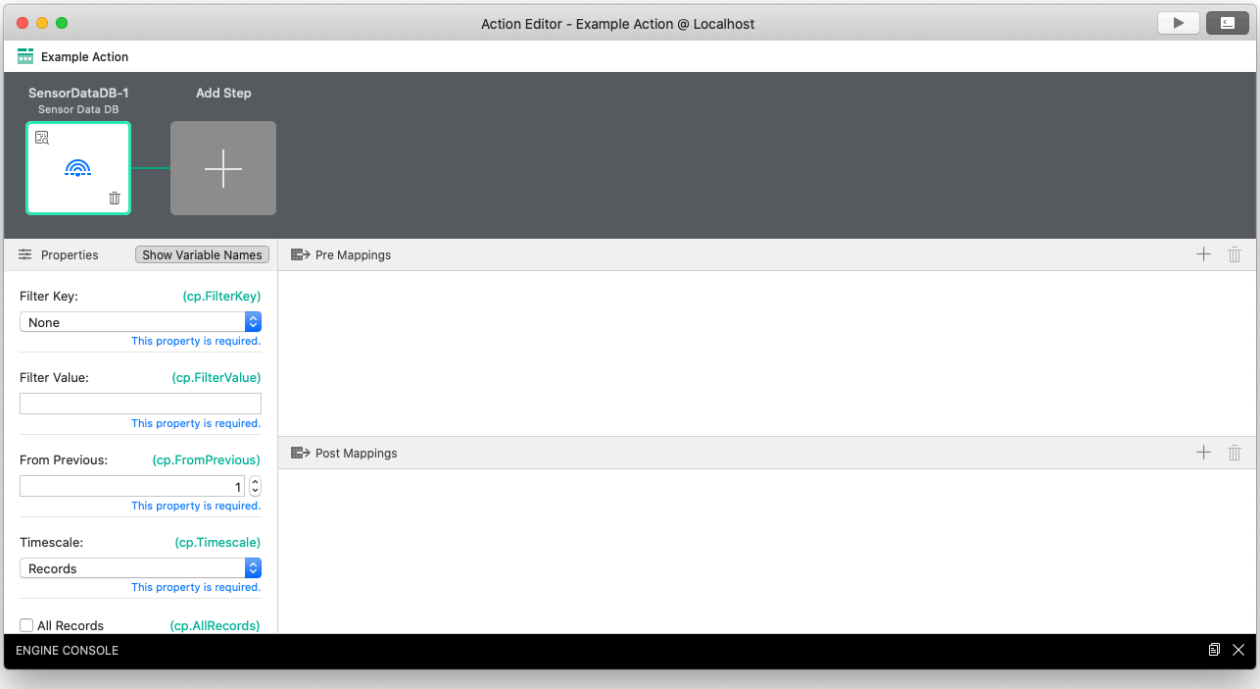
Text2SQL

Text2SQL is a function that allows you to generate SQL statements from the Japanese you enter in the prompt for the data you wish to retrieve using OpenAI's ChatGPT functionality.
Please refer to here for how to use it.

6.3.9.45. SensorDataDB

The Sensor Data DB Action Component is to read from the Sensor Data DB. Each recorded piece of data gets stored in the Sensor Data DB. With this component, you can query that database.
The SensorDataDB component queries data from the SensorDataDB stored in HubKit and passes each record, one row at a time, to the next component.

With the content specified in the property, search the sensor DB and execute a filter expression for each matched record. The matched records can be referenced in the “Value” variable. If the result of the expression is true, the record will be passed through and the output Payload.



Output Payload	Description
`cv.Payload`	Outputs a JSON Object for each record that matches the conditions.

Output Example

```
{
   "AreaId":"c753b341-c856-42fc-9170-69edaa5d8246",
   "AreaName":"Entrance Area",
   "Data":460,
   "DataId":"f969129db906491e876cef43afb4fcb8",
   "DataType":"double",
   "KindId":"7e87a819-135e-40d3-9d5f-c0330f38ec4e",
   "KindName":"Gravio-CO2",
   "LayerId":"1efe01a1-41c7-4d21-bd98-d6bc87af74bd",
   "LayerName":"CO2",
```

```
    "PhysicalDeviceId":"DE-8D-06-FE-FF-57-B5-15",
    "PhysicalDeviceName":"CO2-Sensor-1",
    "Timestamp":"2021-09-13T00:26:42.594348934Z",
    "VirtualDeviceId":"2ce70be6-257b-4854-9680-af1f02d575b4"
}
```

Timestamp is output as a date/time type in the JSON Object type.
For each record, the component variable (`cv.Id`) is given a record ID, the component variable (`cv.Count`) is given the first record that matches the record, and the ID is incremented by 1, and the number of the ID is incremented in order.

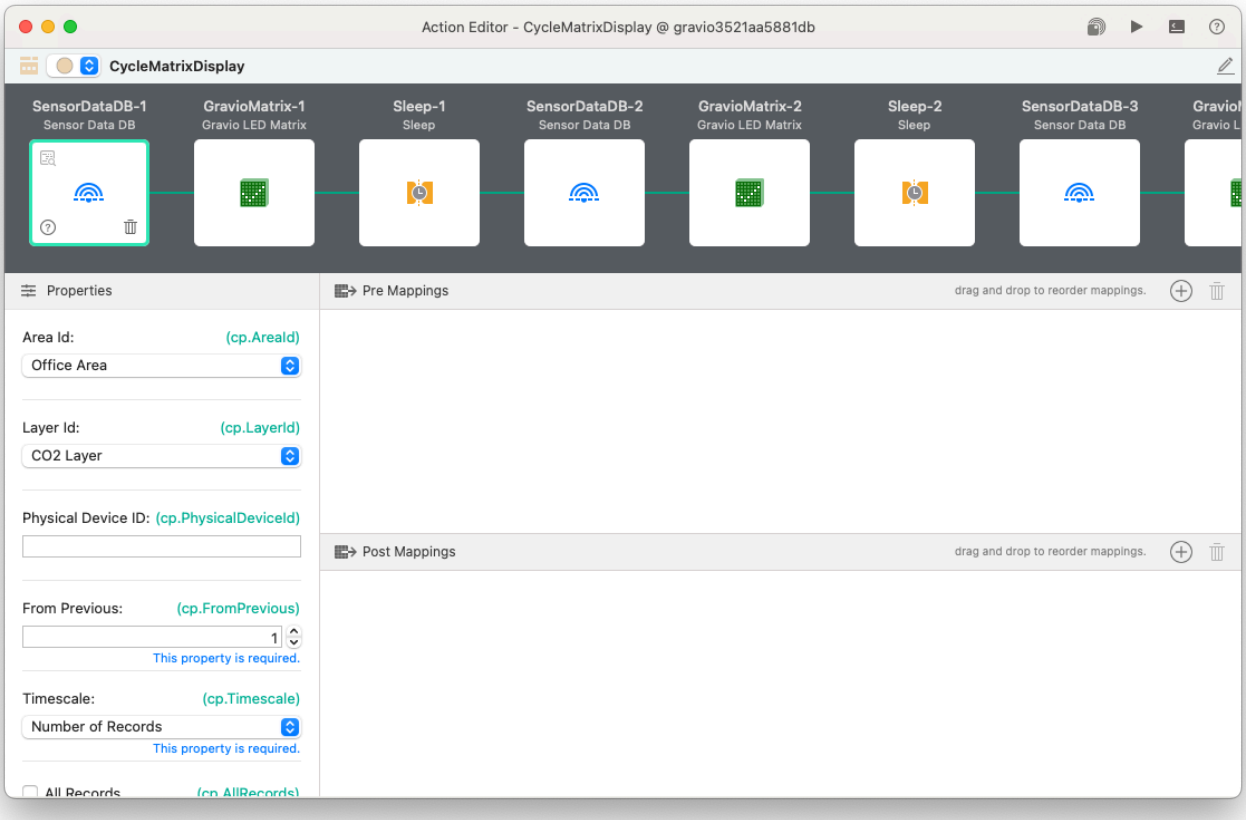
Component Properties

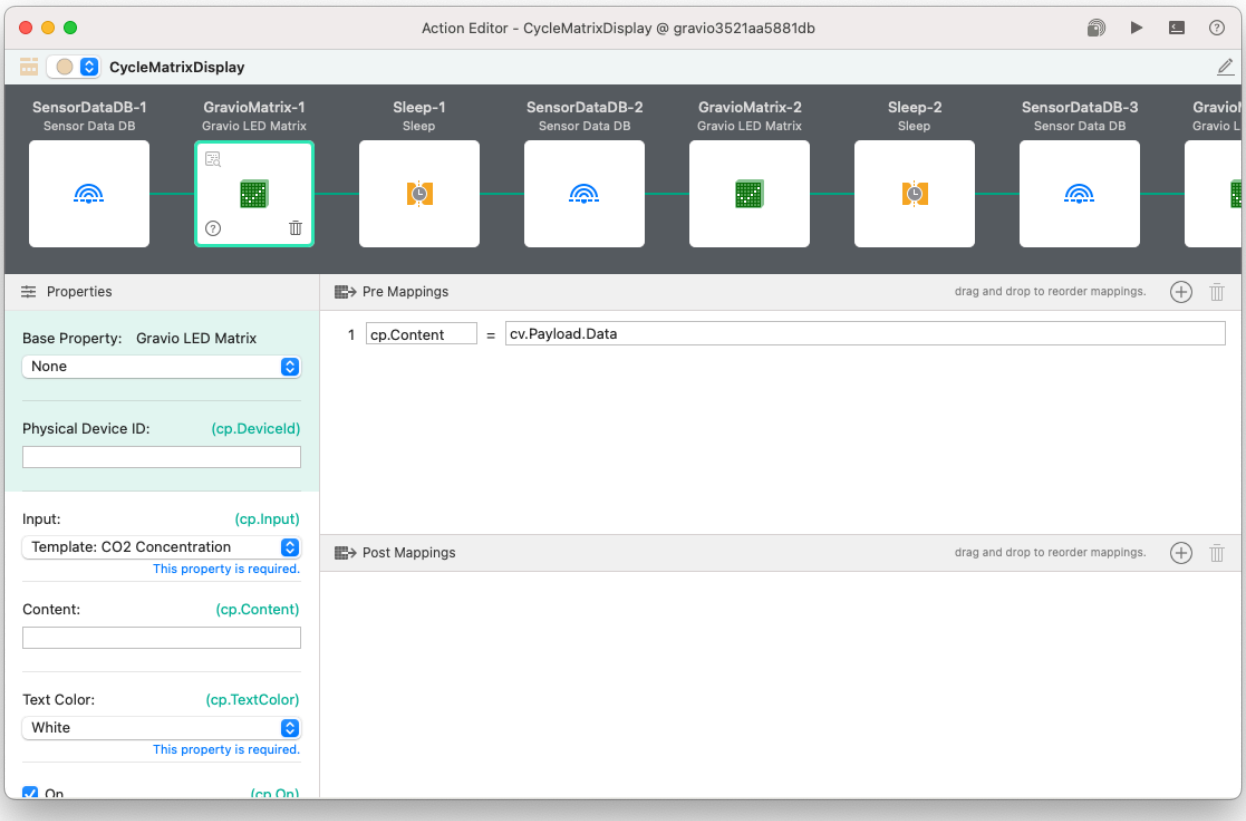
Name	Required	Description
Filter Key	True	The property to filter for
Filter Value	True	The value to filter for
From Previous	True	The number of records to retrieve (e.g., if From Previous is 10 and the timescale is a record, the latest 10 will be retrieved).
Timescale	True	Unit of the latest data to be acquired (choose from records, milliseconds, seconds, minutes, hours, days, weeks, months, and years). cp.Timescale can be one of the following values: “Records”, “Milliseconds”, “Seconds”, “Minutes”, “Hours”, “Days”, “Weeks”, “Months”, or “Years”.
All Records	True	Get all records by overriding the From Previous and Timescale conditions.
Time Rounding	True	Get data from the last second, minute, hour, day, day, week, month, and year rounded up to the last time the component was executed.

Filter Expressions	false	Allow you to choose whether to output the payload in an expression, like the Filter component Filter Expressions. Filter Formula Example
		`Value.Data > 123`
		`Value.Data.key1 == "abc" && Value.Data.key2 > 123`

Example

If you like to use the data from the sensor DB in the next step, for example a Matrix Display that rotates various latest database values, you will have to access the data using `cv.Payload.Data`:





In this example, each SensorDataDB step reads a particular last data set of a layer, the Gravio Led

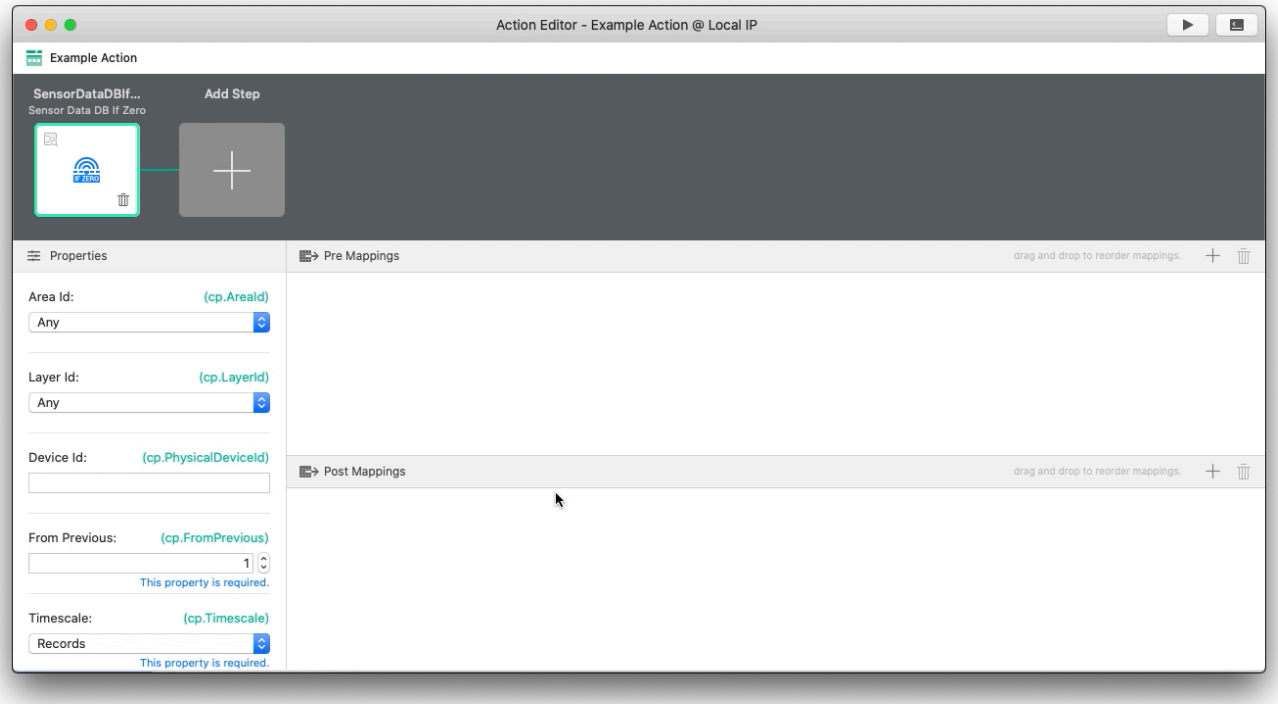
Matrix shows it on the display and the Sleep step pauses the display for 15 seconds.

6.3.9.46. Sensor Data DB If Zero

The SensorDataDBIfZero component outputs the input payload as the output payload if no rows are retrieved; if any rows are found, it outputs nothing and omits the Action and all subsequent components. If no rows are found, the Action flow continues.

With the content specified in the property, it searches the sensor DB and executes a filter expression for each matched record. The matched records can be referenced by the “Value” variable. If the result of executing the expression is true, it passes, but there is no output payload, because it corresponds to a single record in the DB, and there is no output payload. Outputs the input payload even if the execution result is all false.

The SensorDataDBIFZero component is useful for “keep alive” signals or to trigger an action if something “has NOT happened” rather than “has happened”.



Output Payload	Description
`cv.Payload`	Output payload as the output payload if no lines are retrieved.

Output Example

```
{"AreaId":"..", "LayerId":"", "KindId":"", "Timestamp": ..., "VirtualDeviceId": ..., "Data": }
```

Timestamp is output as a date/time type in the JSON Object type.
For each record, the component variable (`cv.Id`) is given a record ID, the component variable (`cv.Count`) is given the first record that matches the record, and the ID is incremented by 1, and the number of the ID is incremented in order.

Filter Formula Example

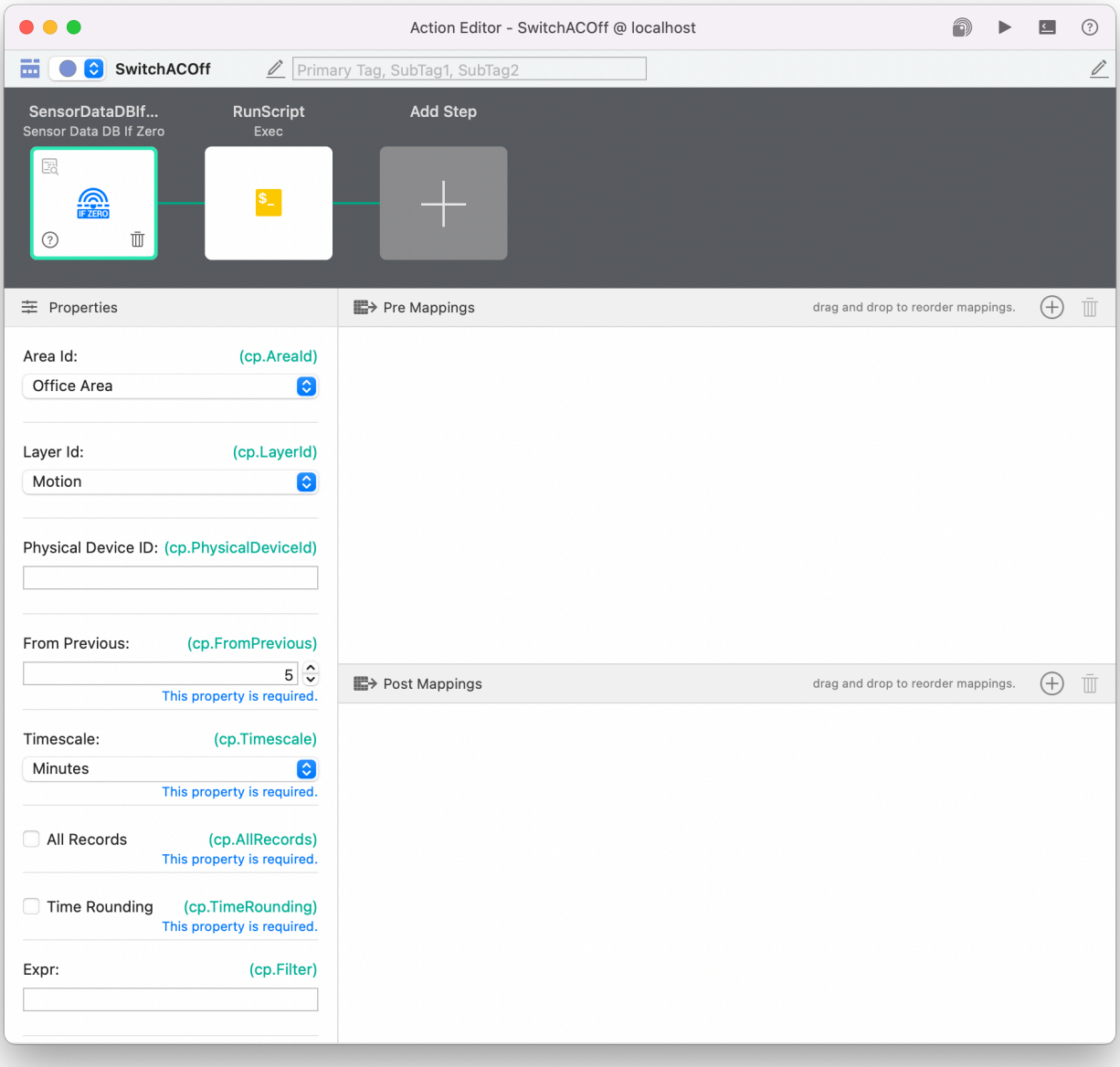
```
Value.Data > 123
Value.Data.key1 == "abc" && Value.Data.key2 > 123
```

Component Properties

Name	Required	Description
Filter Key	True	The property to filter for
Filter Value	True	The value to filter for
From Previous	True	The number of records to retrieve (e.g., if From Previous is 10 and the timescale is a record, the latest 10 will be retrieved).
Timescale	True	Unit of the latest data to be acquired (choose from records, milliseconds, seconds, minutes, hours, days, weeks, months, and years). cp.Timescale can be one of the following values: “Records”, “Milliseconds”, “Seconds”, “Minutes”, “Hours”, “Days”, “Weeks”, “Months”, or “Years”.
All Records	True	Get all records by overriding the From Previous and Timescale conditions.
Time Rounding	True	Get data from the last second, minute, hour, day, day, week, month, and year rounded up to the last time the component was executed.
Filter Expressions	False	Allow you to choose whether to output the payload in an expression, like the Filter component Filter Expressions.

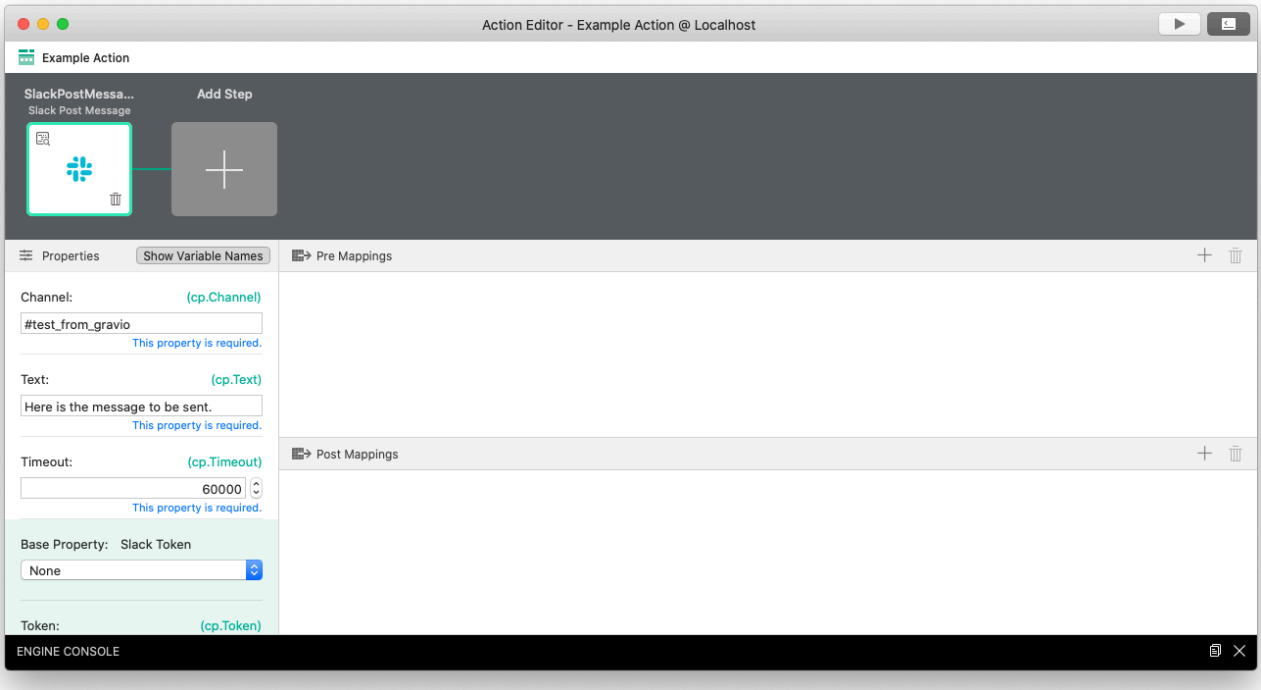
Example Use Case

You may want to trigger a script if no motion is detected for 5 minutes. Create a time trigger that executes every 1 minute, then execute the below action. If the last 5 minutes of the Motion layer are **empty**, the next component (RunScript) will run. If they are not empty, the Action will abort before running the script.


Action Editor - SwitchACOff @ localhost
SwitchACOff
Primary Tag, SubTag1, SubTag2
SensorDataDBIf...
Sensor Data DB If Zero
RunScript
Exec
Add Step
Properties
Pre Mappings
drag and drop to reorder mappings.
Area Id:
(cp.AreaId)
Office Area
Layer Id:
(cp.LayerId)
Motion
Physical Device ID:
(cp.PhysicalDeviceId)
From Previous:
(cp.FromPrevious)
5
This property is required.
Timescale:
(cp.Timescale)
Minutes
This property is required.
All Records
(cp.AllRecords)
This property is required.
Time Rounding
(cp.TimeRounding)
This property is required.
Expr:
(cp.Filter)
Post Mappings
drag and drop to reorder mappings.

6.3.9.47. Slack Post Message

The Slack Post Message Action Component is to post a message to Slack using the Slack API. It can be useful to debug Actions or notifying people. It can be used for sending messages to #channels as well as @people .



Component Properties

Name	Required	Description
Channel	True	The #channel or `user you like to post your message to. If you like to include sensor data, wrap the text into double quotes and append it with the sensor data. For example:` “The temperature is “av.Data” degrees Celsius”@
Text	True	The text you like to send
Timeout	True	Timeout in
Slack Token	False	Slack Token from the Base Property Profile
Token	True	Your Slack Token, if not loaded from the Base Property Profile

Use the base Base Property Profile property to read the values such as the token from the settings rather than specifying the details within the component.

Apps on Slack

First you will need to create an app on Slack in order to interact with your workspace. You can follow the instructions here to create your Slack app: https://api.slack.com/apps

You will need to:

- Create an App on Slack
- Add the `chat:write` (or any other scope you like to use) to “Scopes” under “OAuth & Permissions” in the Features menu.
- Hit the “Install App to Workspace” button in “OAuth Tokens & Redirect URLs”
- Add the app to the channel you like it to post to

For more complex scopes and permissions, please refer to the Slack Documentation.

slack api

Search

Settings

- Basic Information
- Collaborators
- Install App
- Manage Distribution
- Submit to App Directory

Features

- App Home
- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- Workflow Steps
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- Beta Features
- Where's Bot User

Slack ♥
Help
Contact
Policies
Our Blog

OAuth & Permissions

OAuth Tokens & Redirect URLs

Tokens for Your Workspace

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. Learn more.

Bot User OAuth Access Token

xoxb- Copy

Reinstall App

Redirect URLs

You will need to configure redirect URLs in order to automatically generate the Add to Slack button or to distribute your app. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. Learn more.

Redirect URLs

You haven't added any Redirect URLs

Add New Redirect URL

Save URLs

Scopes

A Slack app's capabilities and permissions are governed by the scopes it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
chat:write	Send messages as @

Add an OAuth Scope

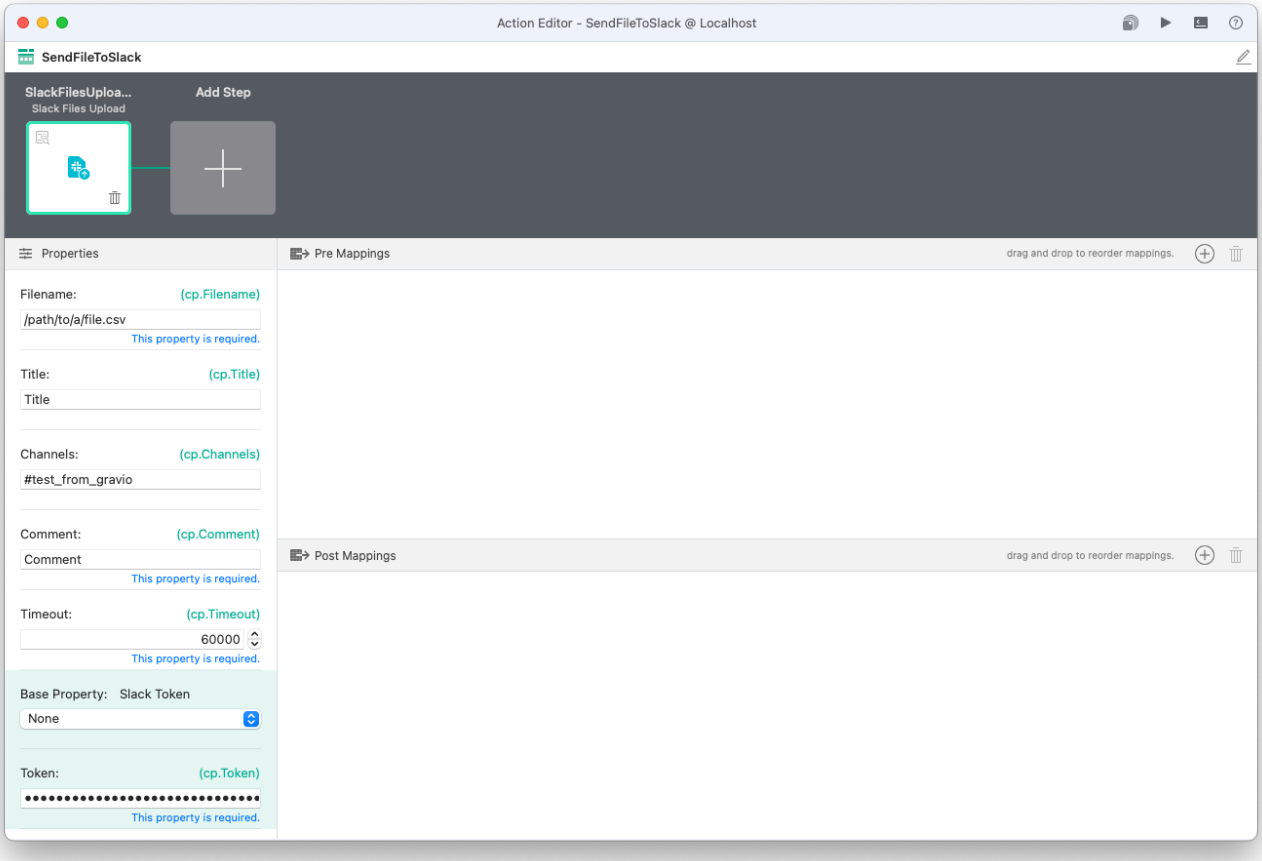
6.3.9.48. Slack Files Upload

The Slack Files Upload action component can use the Slack API to post messages and files to a Slack channel. The authentication happens via the Slack authentication token, which you can retrieve from your Slack Admin console.

The Slack Files Upload component can be useful to debug actions and notify people. You can send messages to #channels or @people directly.

To get started, create a new app in your Slack API admin and add “chat: write: bot” and “files: write: bot” to “Scopes” under “OAuth & Permissions” in the Features menu. You can follow the instructions from the Slack Post Message Component for more details on how to retrieve the authentication token.

Once you have your token, enter it in the Token property as seen below.



For the paths required, you can refer to the documentation how to handle paths in Gravio

You can also refer to a Base Property profile if you wish to define the settings on HubKit level rather than Component level.

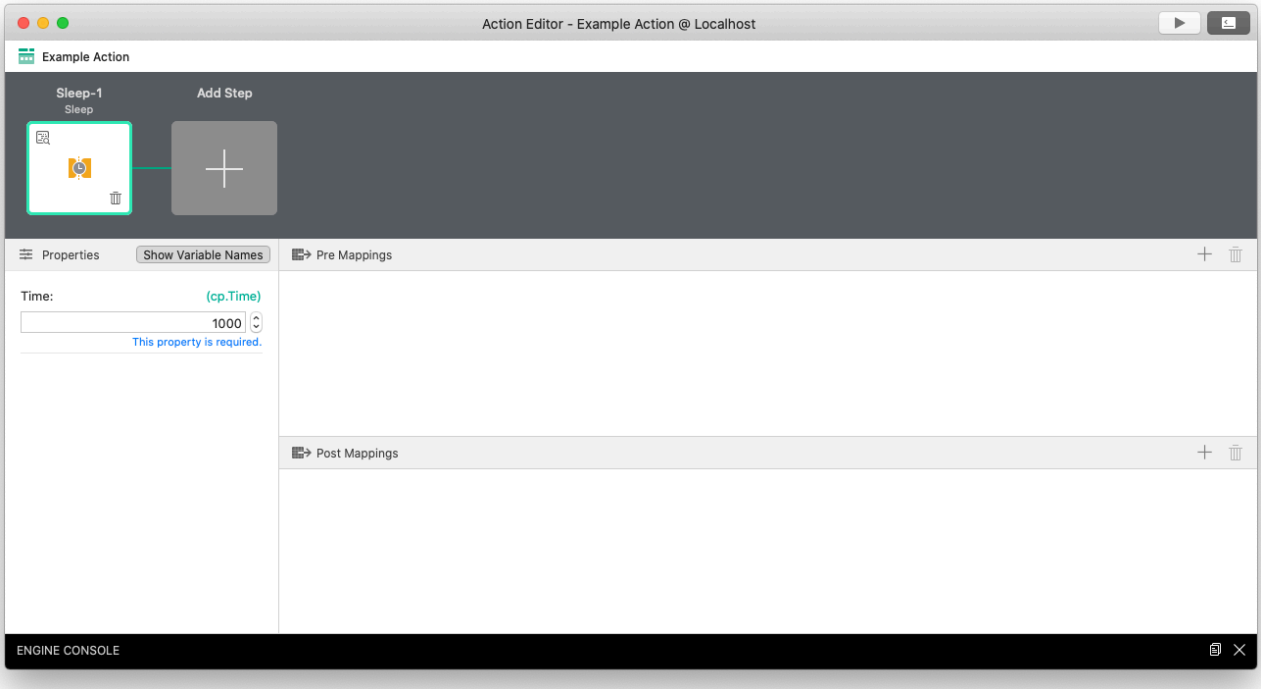
Output payload	Description
cv.Payload	Outputs the output payload of the previous component as is. (Pass through)

Component properties

Property name	Description
file name	Specify the file to send with the message
title	Specify the title of the file
Channel	Specify Slack channel
comment	Specify the message text you want to send
time out	Specifying the transmission timeout time
token	OAuth Access Token set in Slack

6.3.9.49. Sleep

The Sleep Action Component is to insert a pause between the two steps of the action.



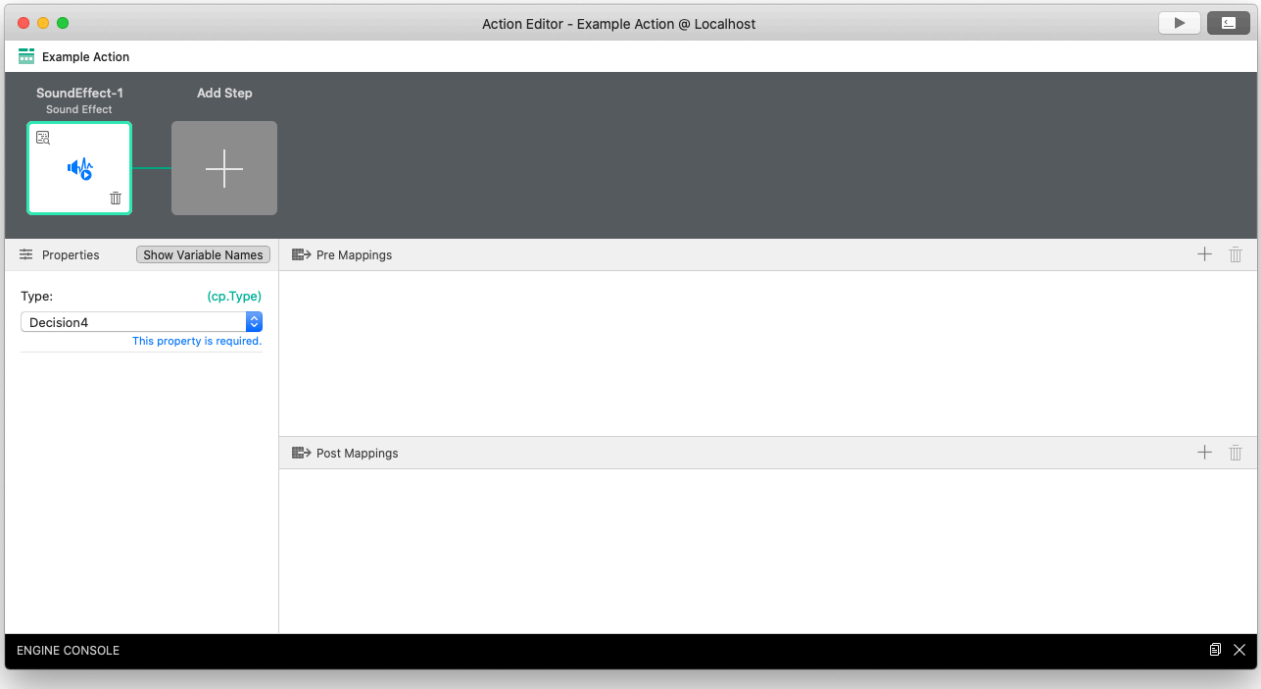
Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

Component Properties

Name	Required	Description
Time	True	Specify pause time in milliseconds

6.3.9.50. Sound Effect

The Sound Effect Action Component is to play a sound effect.



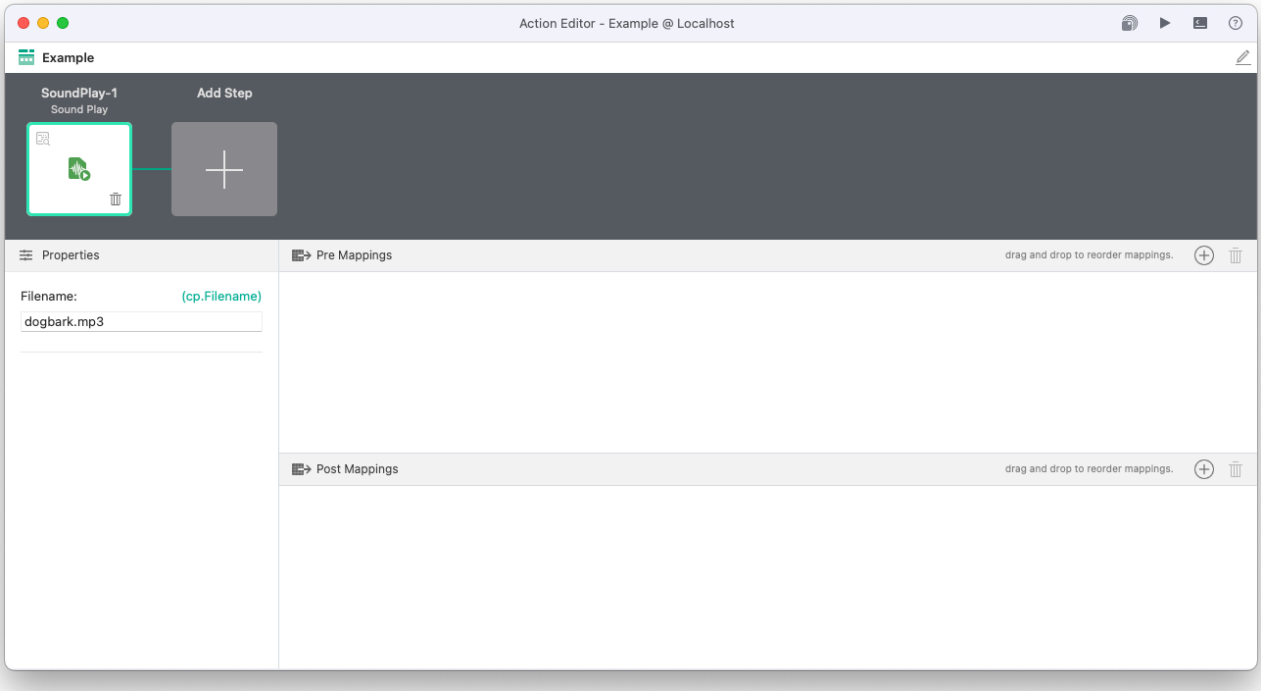
Output Payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

Component Properties

Name	Required	Description
Type	True	Type of Sound Effect. When setting by cp.Type value, it can be one of “warning1”, “warning2”, “decision3”, or “decision4”.

6.3.9.51. Sound Play

The Sound Play component allows you to play a specified mp3 file.



Output payload	Description
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass-through)

Component Properties

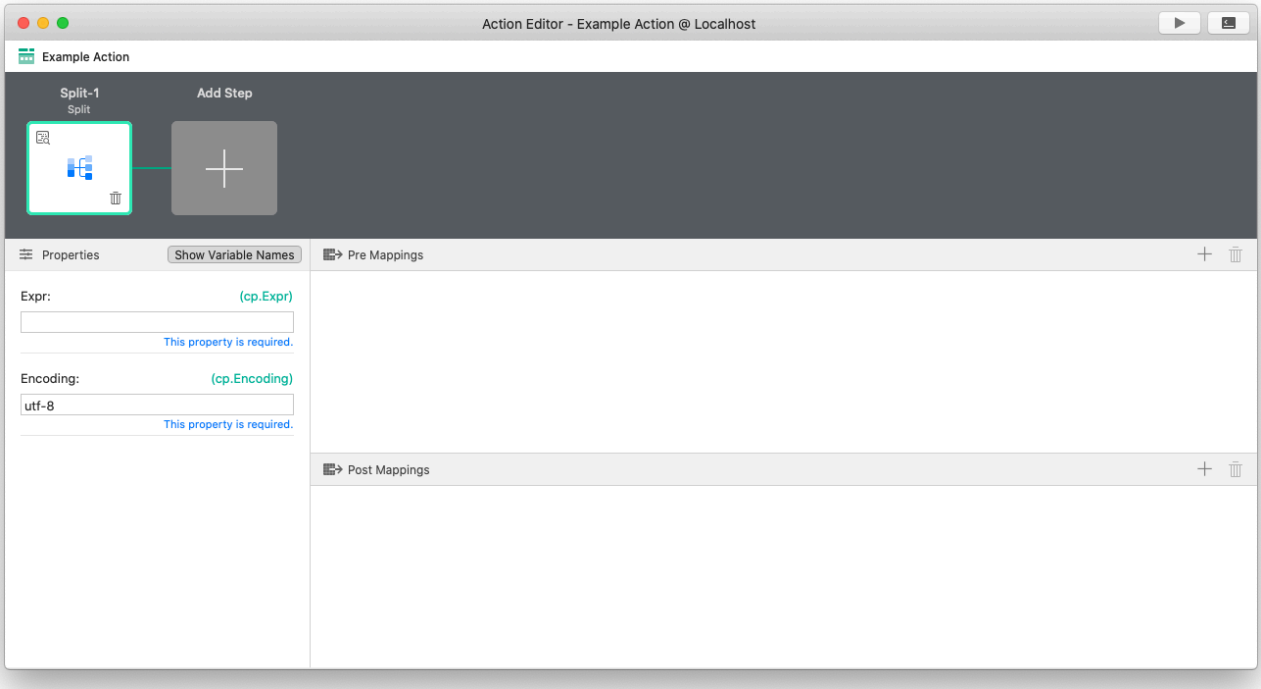
Property name	Description
File name	Specify the file name in case of file input

If the file name is not specified, the referenced filename refers to a file in the `actmgr/data` folder, but for how to refer to a file outside this folder, please see here.

- Sound Play is currently supported by HubKit Windows, Linux, RaspberryPI and the Gravio Hub, Mac is not supported due to OS limitations.

6.3.9.52. Split

The Split component is a component that splits the input Payload by line feed if it is a string, or by element if it is a JSON array, and passes it to the next component.



The Split component provides the ability to filter whether the output data after the split is passed to the next component or not.
The conditions can be entered as expressions.
The following two variables are available in this component's expression. Note, since they are locally used variables, we do not require the `cv.` prefix.

Variable name	Meaning of the variable
Value	The actual data value after the splitting process
Index	Indicates the index of the element in case there were multiple elements

Conditional Example	Formula
Pass only elements that begin with abc to the next component	`Value =~ "^abc"`
Pass only the even-numbered elements.	`Index % 2 == 0`

Parameters

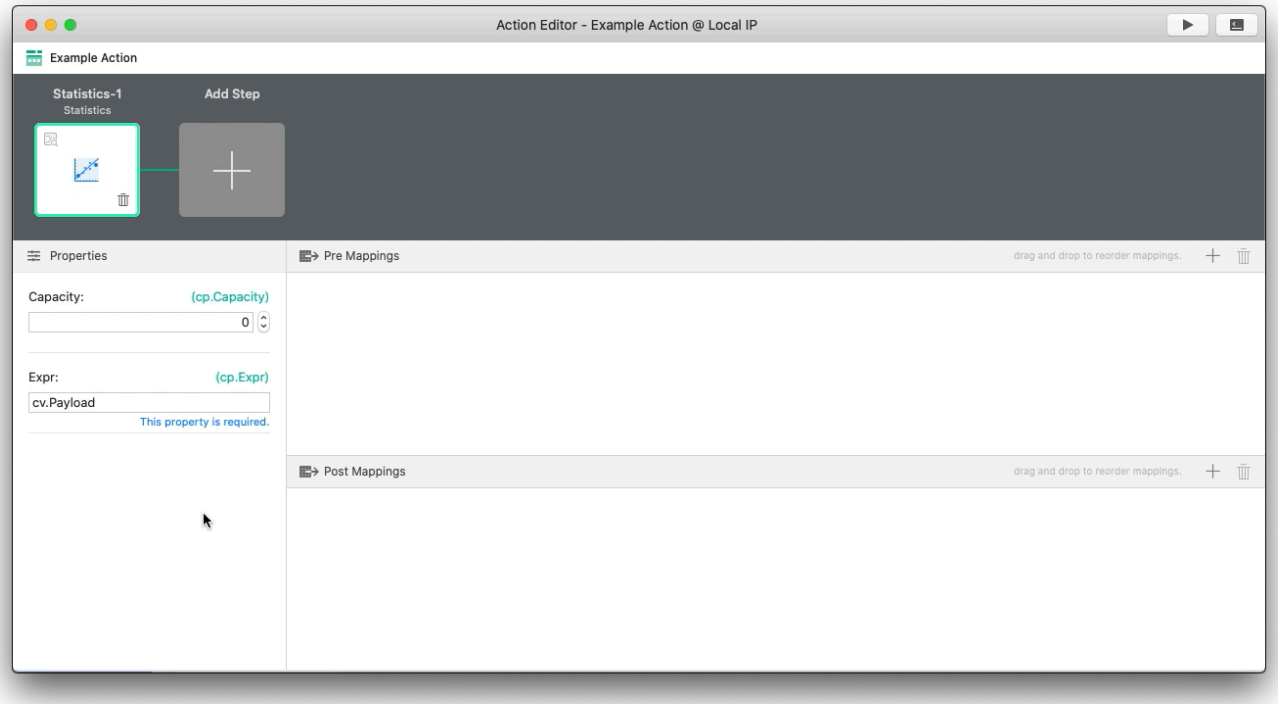
Expr	Required	Expr
Encoding	Required	Encoding

Example

If the incoming `cv.Payload` is `["a", "b", "c", "d", "e"]`, the Split component will split it into 5 payloads each submitted to the next component if the expression is `true` for all of them. In the above case, the Value is `a` for Index `0` and the Value is `b` for Index `1`.

6.3.9.53. Statistics

The Statistics component stores the statistics (Min, Max, Mean/Average, Median, Standard Deviation, Sum, Mode) in the component variables with the specified values.



If the value specified is an improper value for a number, that value will be excluded from the statistics. The number of values that enter the statistics without being excluded will be output to the component variables as Count.

Example.
If the test scores are flowing in as a Payload, suppose you specify “cv.Payload” for Expr and “0” for Capacity.
When the 10 values have flowed in and finished, Statistics will output an array of values as a payload.
At the same time, Statistics outputs the statistics data in the Statistics component variable.

cv.Payload `[100 100 80 80 80 70 70 30 10 0]`
cv.Statistics `{"Average":62 "Count":10 "Max":100 "Mean":62 "Median":75 "Min":0 "Mode":[80] "StandardDeviation":34 "Sum":620}`

The value expression to specify the value can be a numeric expression.
In the above example, if you specify “cv.Payload * 100”, the statistics will be taken as each value multiplied by 100.
If you want to obtain multiple statistics from a single value, or if you want to extract multiple values from the JSON Object type to obtain statistics, for example, you want to obtain statistics on multiple values at once, specify the values in an array of value expressions.
`[cv.Payload, cv.Payload*100]`
Assuming that the JSON Object type `{"val1":123, "val2":456}` flows

`[cv.Payload.val1, cv.Payload.val2]`
It is also possible to specify that the value expression is an array.
If you specify an array as a value expression, then the component variable Statistics will also be an array and the respective statistics will be taken.

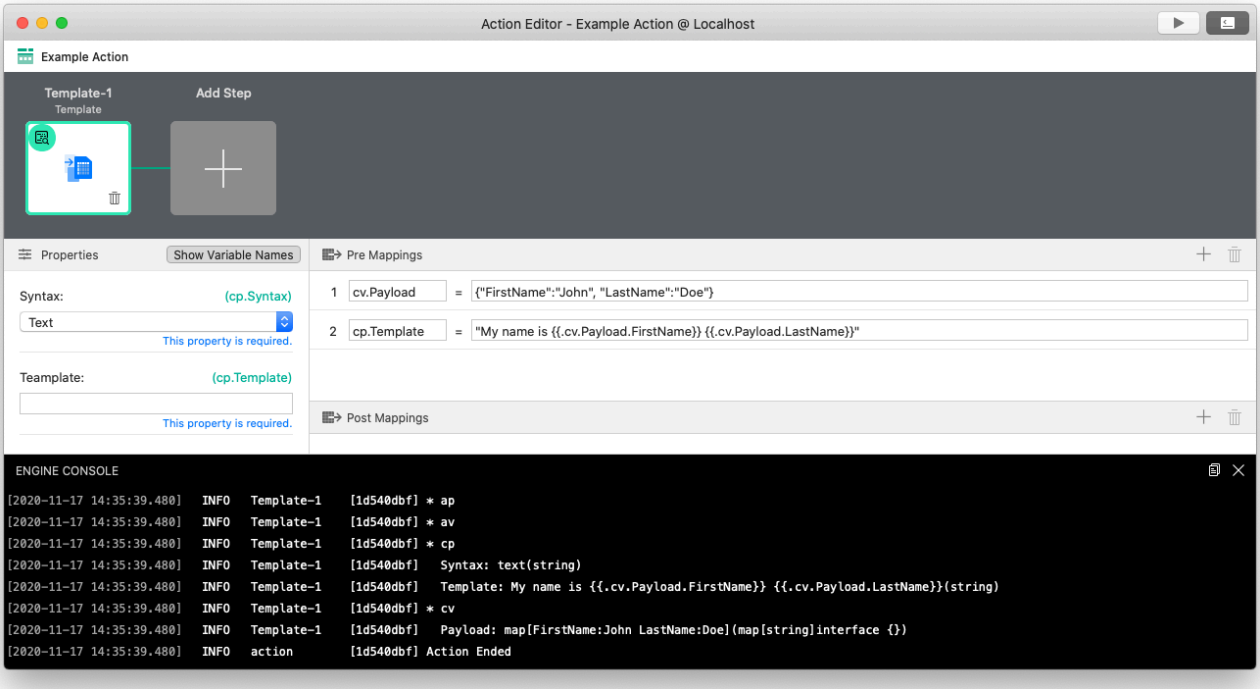
To get statistics on sensor values in the SensorDataDB component, you can create statistics based on sensor data values by specifying the value expression property “cv.Payload.Data”.

If the data is sent in excess of the number of capacity properties, the data will be cleared up to that point once the Capacity is reached.
For example, if you set the capacity to `10` and 12 data come in, the first time you will get statistics on the data from 1-10, then these 10 data will be discarded and calculated on the 11-12th data.

Output Payload and Variable	Description
`cv.Payload`	Payload to output an array of values as an array
`cv.Statistics.Average`	Average
`cv.Statistics.Count`	The number of values that entered the statistics without being excluded
`cv.Statistics.Max`	Max value
`cv.Statistics.Mean`	Mean
`cv.Statistics.Median`	Median
`cv.Statistics.Min`	Minimum value
`cv.Statistics.Mode`	Most Frequent value
`cv.Statistics.StandardDeviation`	Standard deviation
`cv.Statistics.Sum`	Total

6.3.9.54. Template

The Template component is a component that generates a string (text) from a specified template using placeholders.



The template syntax is a GoLang standard described on https://golang.org/pkg/text/template/ . `tp.` / `tv.` / `ap.` / `av.` / `cp.` / `cv.` are all referable to the template, so if these variables are referenced in the template, they will be replaced by their values.

Example:

Template	Description
`{{.cv.Payload.LastName}} {{.cv.Payload.FirstName}}`	Each variable's value replaces the double-curly-braced content containing a variable prefixed with a `.`

Output Payload	Description
`cv.Payload`	Template string generated by the contents of a template string with variable parts replaced by variable values (text)

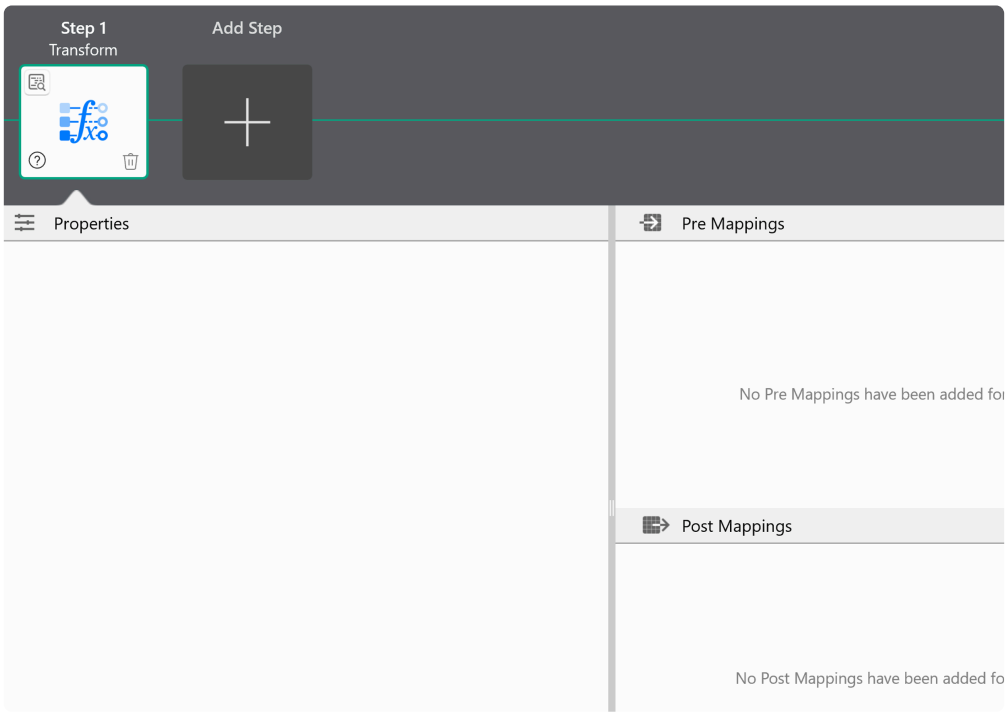
Tip: If you write data into an `av.` variable (Action Variable) it will be available throughout the lifetime of the action also in other components.

Component Properties

Name	Description
Syntax	Specify text or HTML. The value of cp.Syntax can be either “text” or “html”.
Template	A string containing the variables to be replaced

6.3.9.55. Transform

The Transform component is the component that handles Pre Mapping and Post Mapping.
A dedicated component used to transform input and output data.



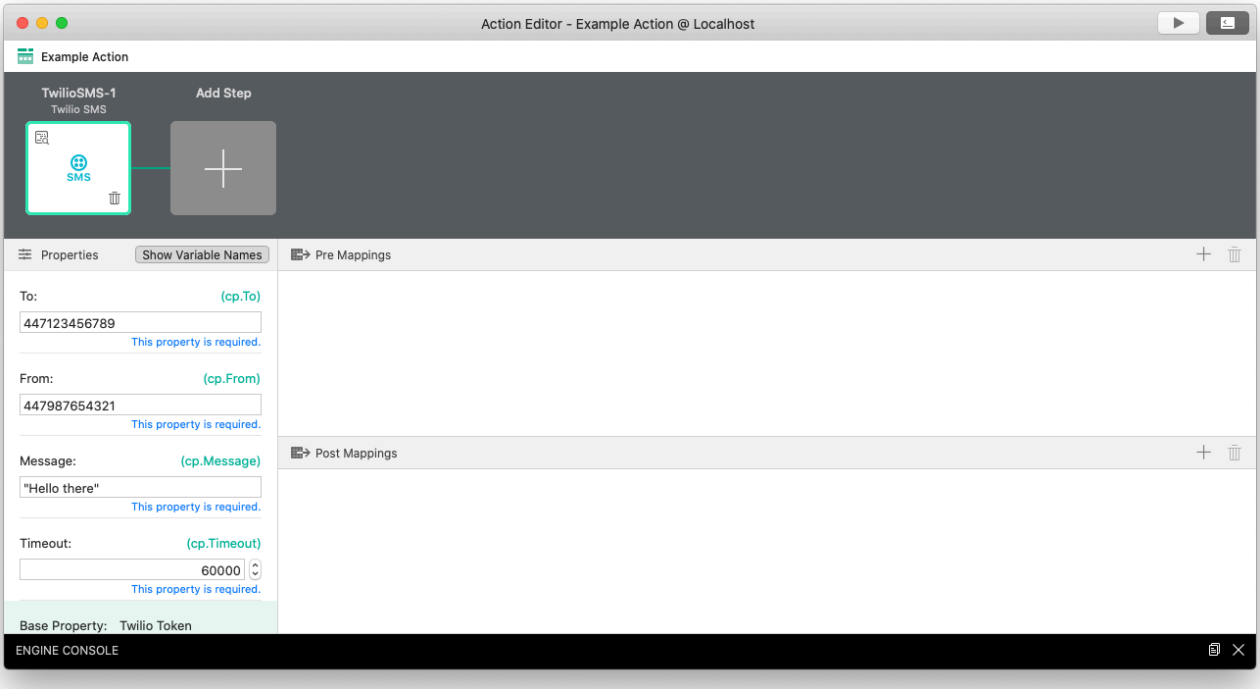
Output Payload	Explanation
`cv.Payload`	Outputs the output payload of the previous component as is. (Pass through)

Component Properties

no Properties.

6.3.9.56. TwilioSMS

The Twilio SMS Action Component is to send an SMS text message via Twilio. It is useful to notify people of information from sensors.



Component Properties

Name	Required	Description
To	True	The number to send the message to, starting with the country code, for example (`44` in UK) `447123456789`
From	True	From
Message	True	Message to be sent, for example `"The current temperature is" + cv.Payload + "°C"`
Timeout	True	Timeout (ms)
Twilio Token	False	Twilio Token from the Base Property Profile
AccountSID	True	AccountSID if not taken from the Base Property Profile
AuthToken	True	AuthToken if not taken from the Base Property Profile

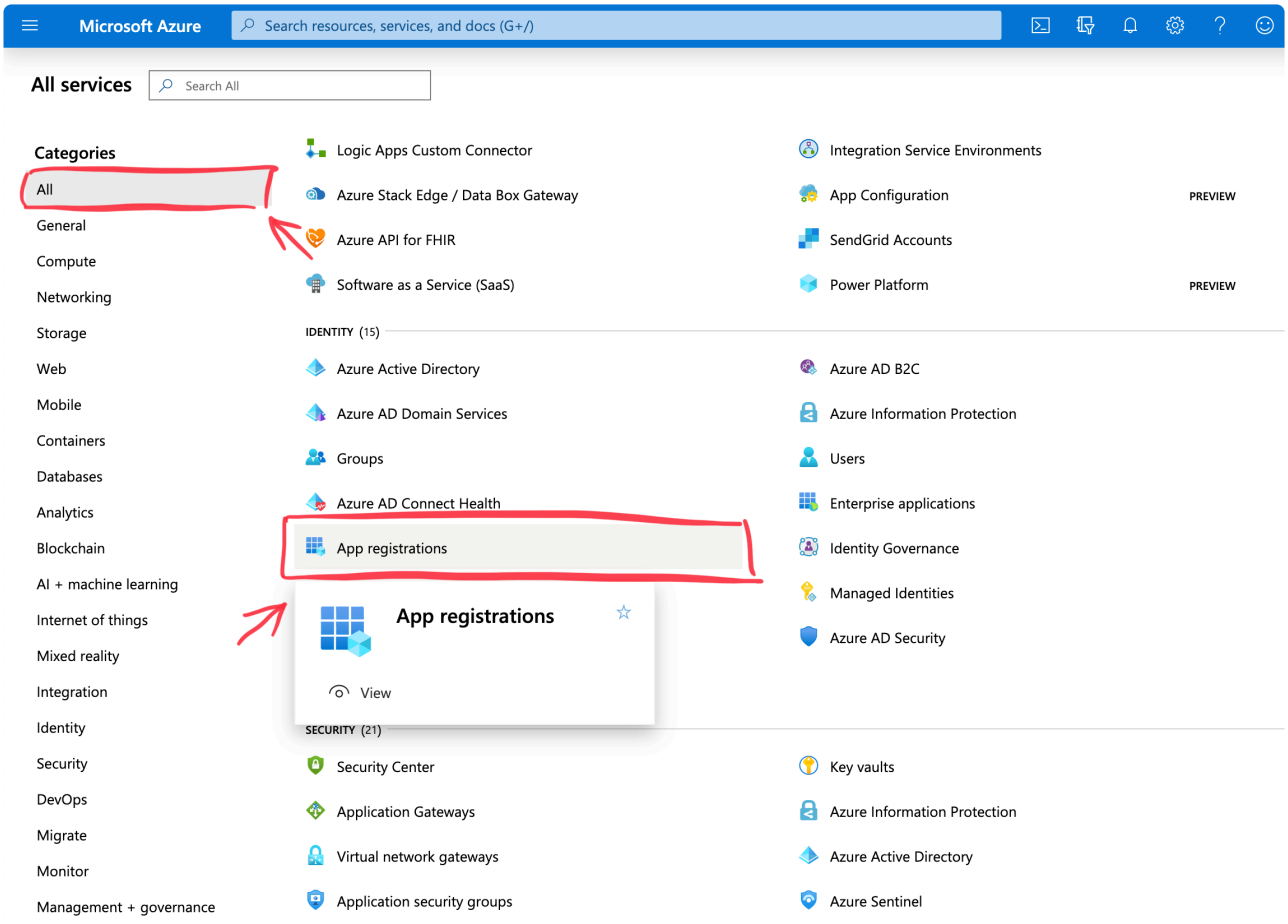
6.3.9.57. WriteToExcel365

The WriteToExcel365 component can output data to a OneDrive in Excel format.

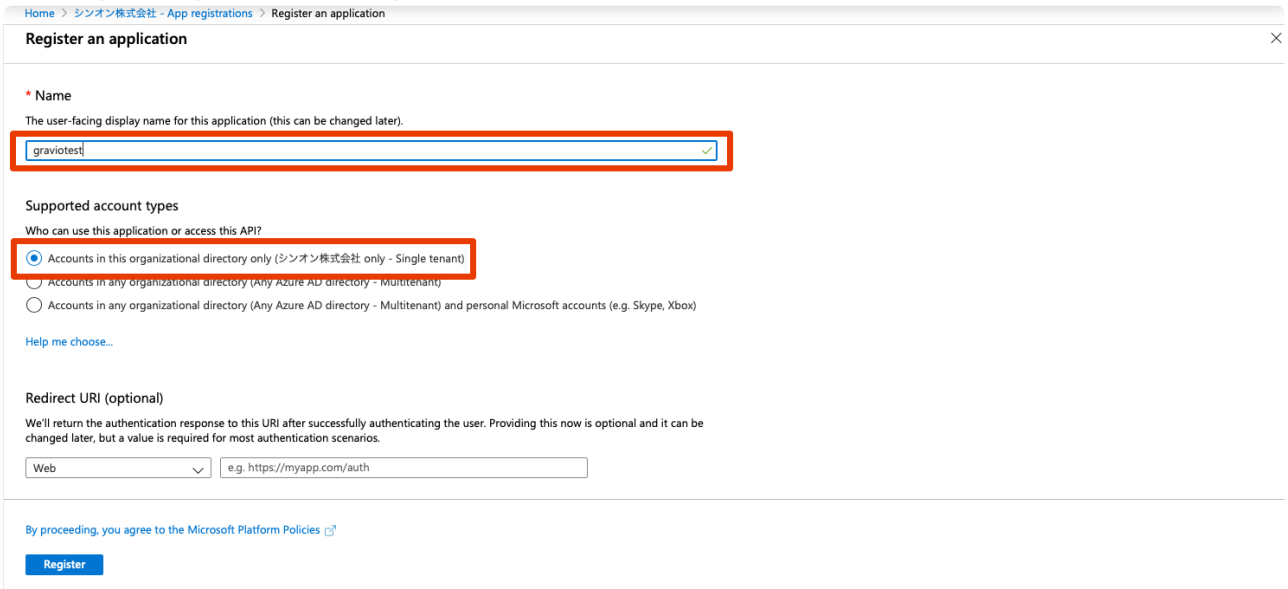
The use of this component requires a contract and pre-configuration of Office365 for Business and OneDrive for Business.

How to prepare your Microsoft account

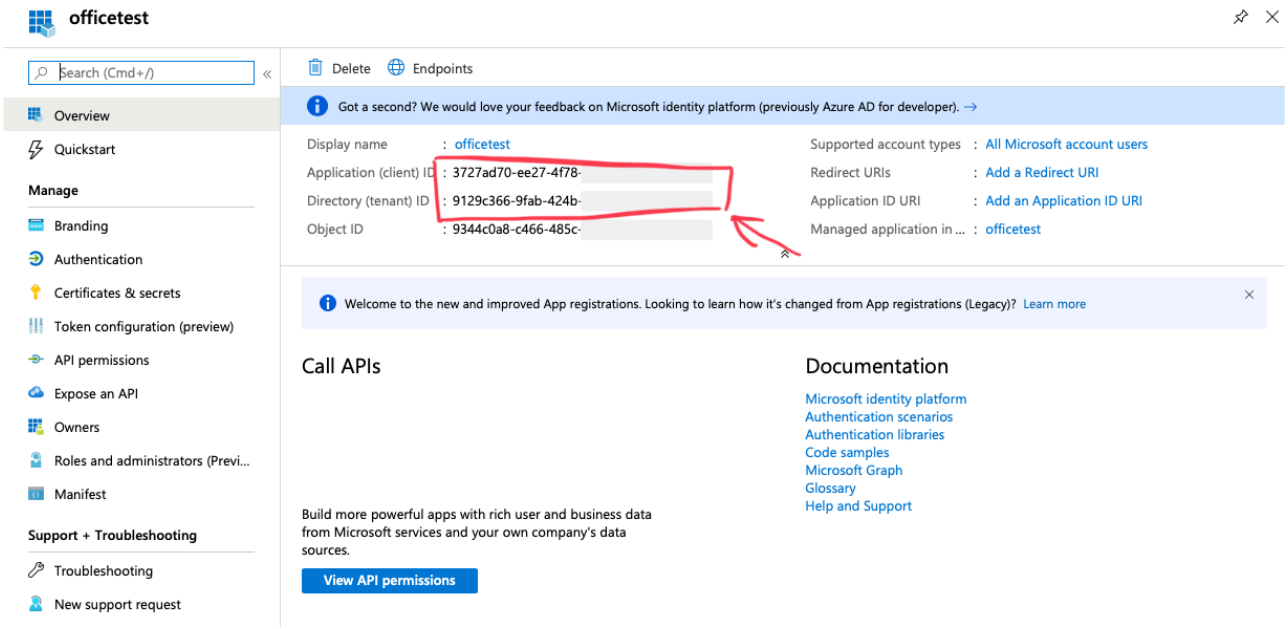
1. Prepare an Office365 account and log in to the Azure portal .
2. Search for *App registration* in the *Identity* section to create an application.



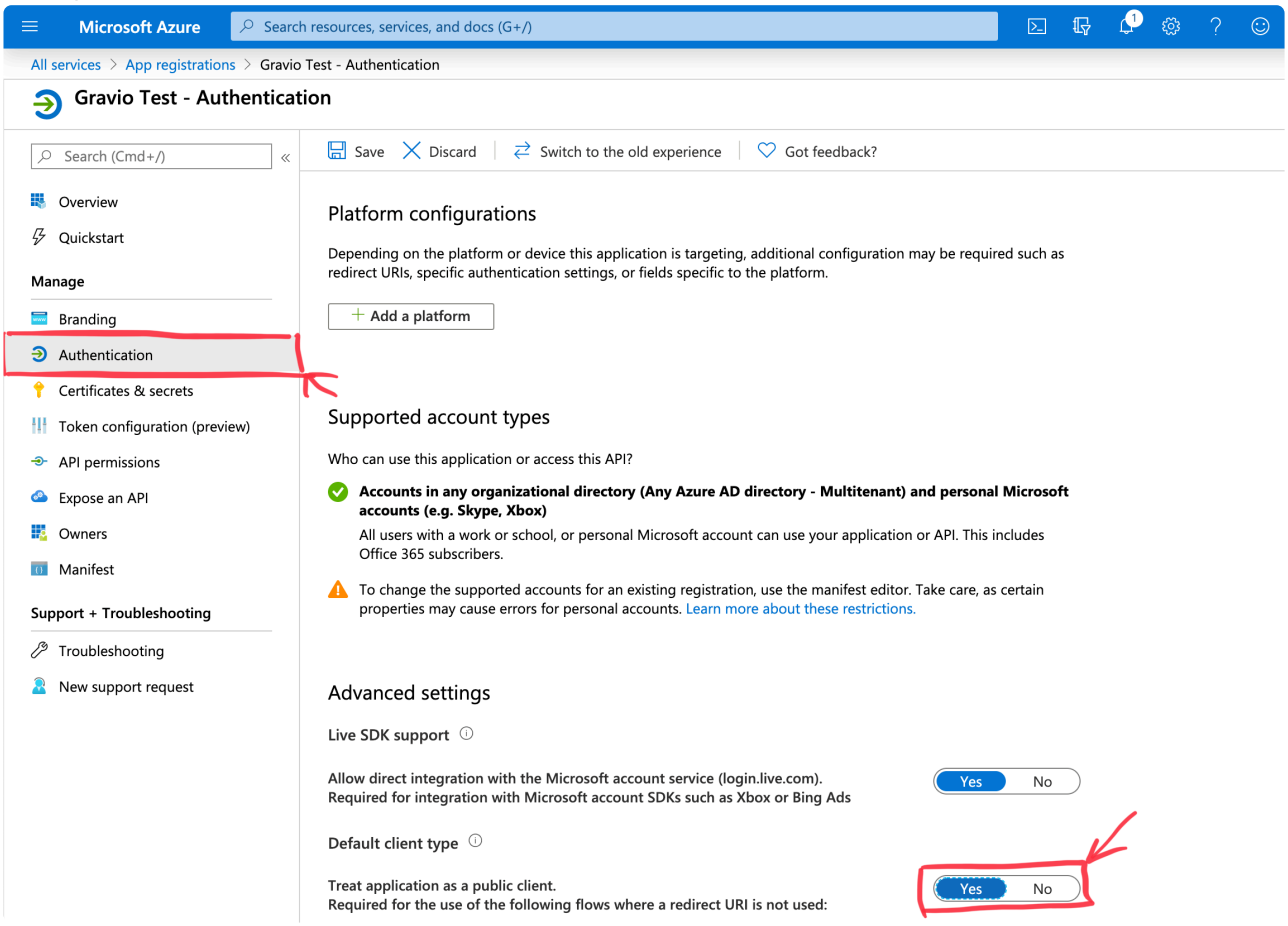
3. Give the app a meaningful name, and pick *single tenant*. (Note: if you don't have this *single tenant* option, you may not have a registered business account)



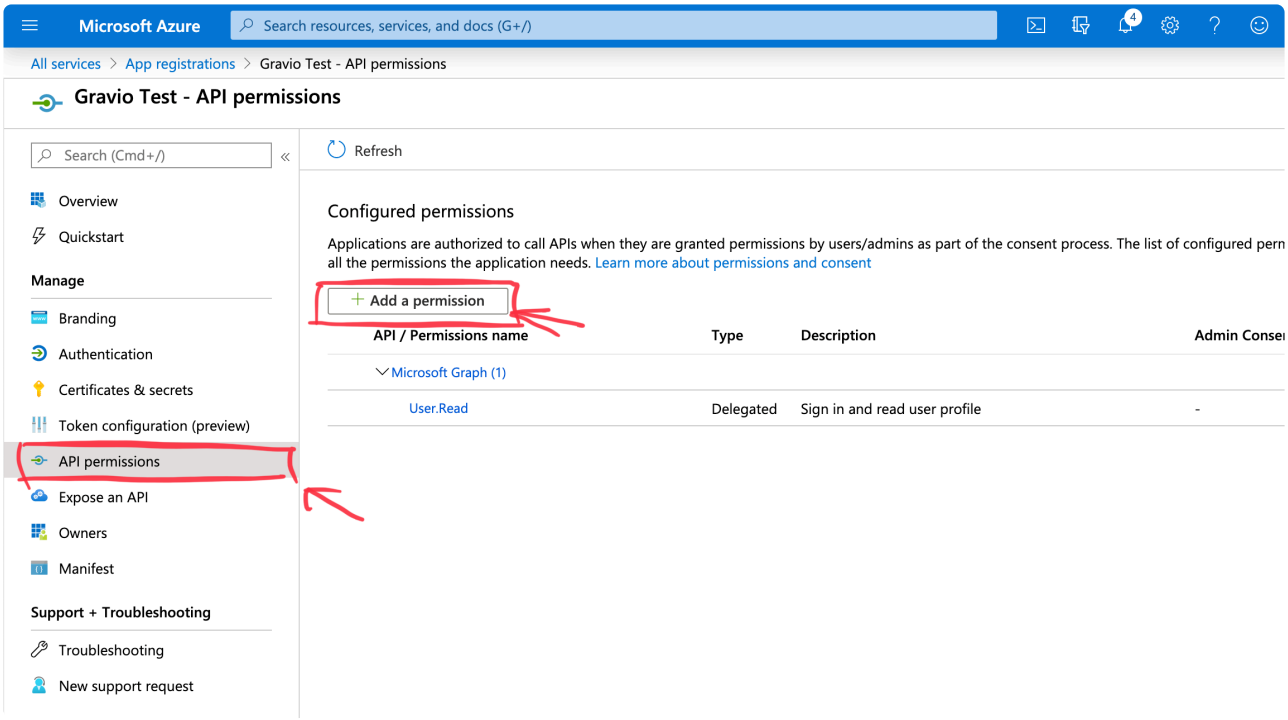
4. Open the details screen after creating the application and note the displayed *Application (Client) ID* and Directory (Tenant) ID_. You will need to enter them in Gravio.

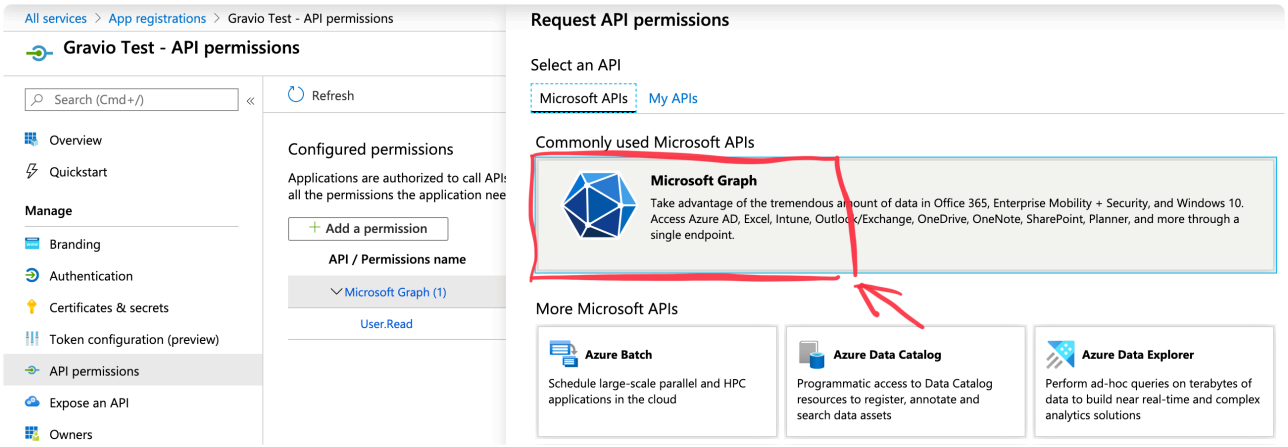


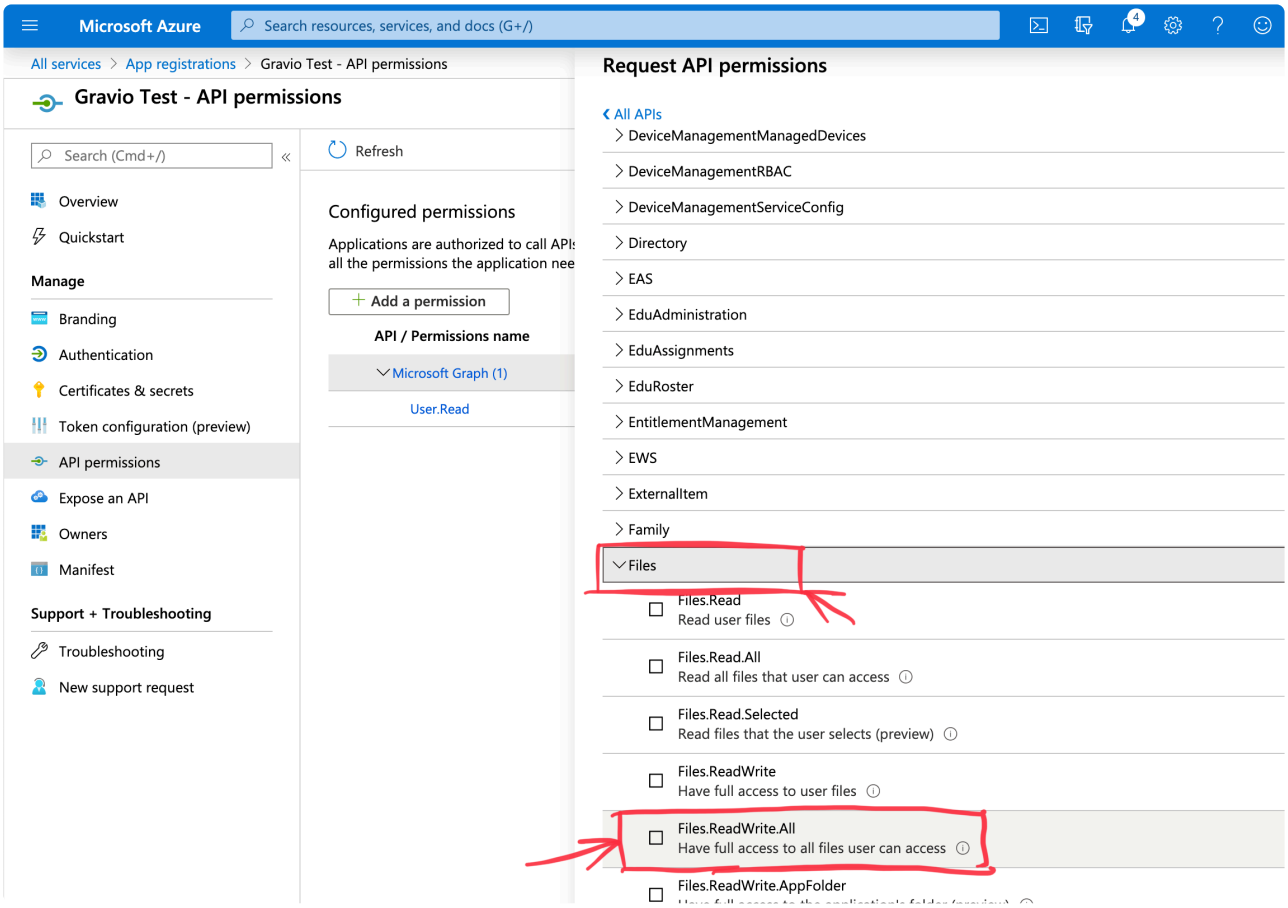
5. Put these values in the properties of the WriteToExcel365 component along with the email address and password of the Office365 account.
6. Now go to *Authentication* and make sure the app is treated as a *public client*:

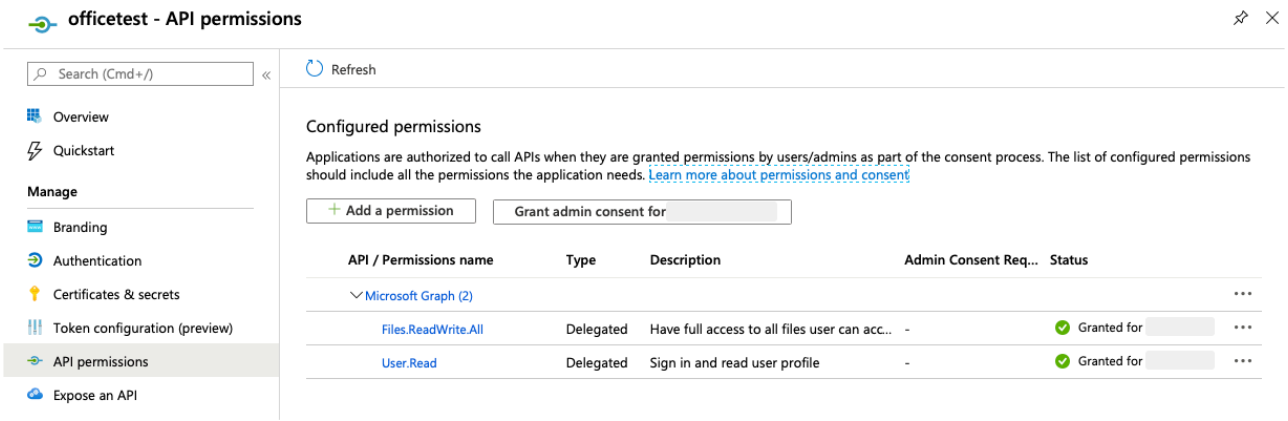


7. Then open *API Permissions* and add *Files.ReadWrite.All* permissions, which you can find under the *Microsoft Graph* section.


Microsoft Azure
Search resources, services, and docs (G+/)
All services > App registrations > Gravio Test - API permissions
Gravio Test - API permissions
Search (Cmd+/)
Refresh
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
Token configuration (preview)
API permissions
Expose an API
Owners
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Configured permissions
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured perm
all the permissions the application needs. Learn more about permissions and consent
Add a permission
API / Permissions name
Type
Description
Admin Consei
Microsoft Graph (1)
User.Read
Delegated
Sign in and read user profile
-


All services > App registrations > Gravio Test - API permissions
Gravio Test - API permissions
Search (Cmd+/)
Refresh
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
Token configuration (preview)
API permissions
Expose an API
Owners
Configured permissions
Applications are authorized to call APIs
all the permissions the application nee
Add a permission
API / Permissions name
Microsoft Graph (1)
User.Read
Request API permissions
Select an API
Microsoft APIs
My APIs
Commonly used Microsoft APIs
Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.
More Microsoft APIs
Azure Batch
Schedule large-scale parallel and HPC applications in the cloud
Azure Data Catalog
Programmatic access to Data Catalog resources to register, annotate and search data assets
Azure Data Explorer
Perform ad-hoc queries on terabytes of data to build near real-time and complex analytics solutions





8. The preparation is now complete.

Component Properties

File Name	Specify Excel file name to output
Sheet Name	Specify sheet name to output data
Write Mode	Specify whether to overwrite (Replace) the file or append the data (Append). The value of cp.WriteMode can be either “Append” or “Overwrite”.
Output header	Select whether to output header when outputting data

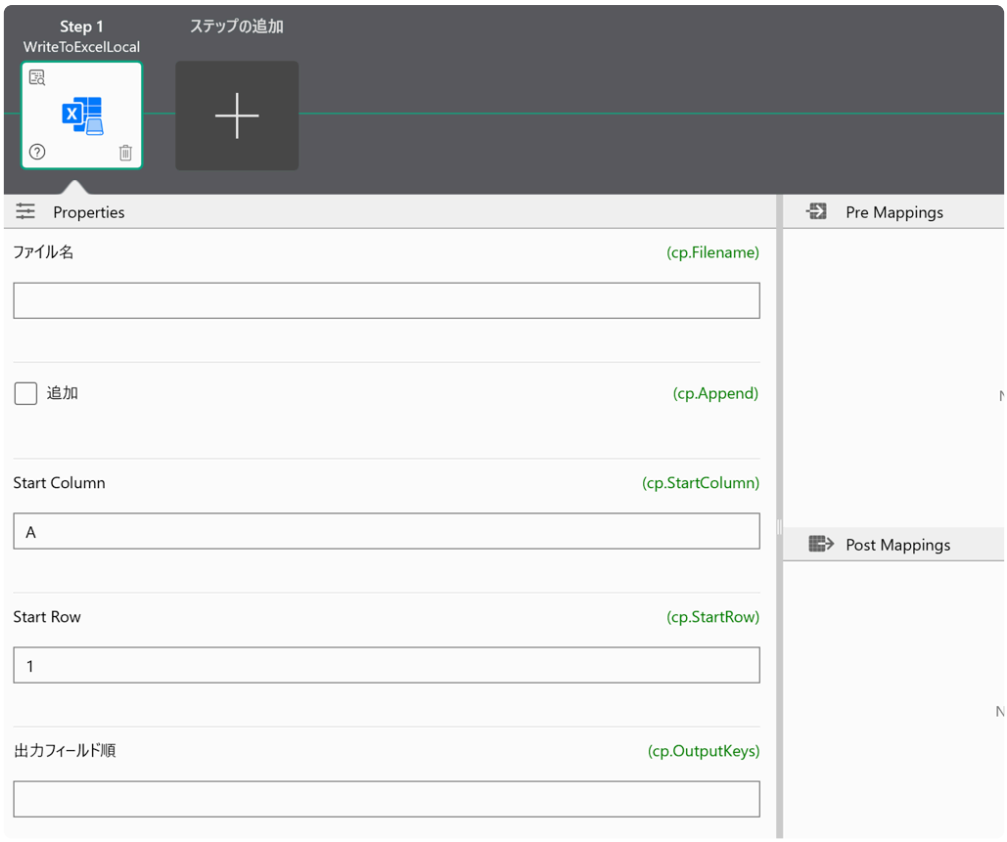
Column Mapping	Specify which column of Excel to output among the columns of step output. More than one can be specified. (Example: To output Step_a of step output to column A of the Excel and Step_c to column B of Excel, specify `Step_a: A, Step_c: B` and the field of step output and the column of Excel with `:`, If you specify more than one, separate with `,` comma)
Client Id	Specify Application (client) ID
Tenant Id	Enter the Directory (tenant) ID
Instance	Only change if your Azure account is hosted on a custom domain, as opposed to on Microsoft's domain
Username or Email	Specify Office365 Business account
password	Specify password for Office365 Business account

6.3.9.58. WriteToExcelLocal

The WriteToExcelLocal component can output data in Excel format to a local xlsx file.

The input payload (`cv.Payload`) can be an array of JSON arrays or an array of JSON Objects. Otherwise, it is converted to a string and output as a single line, single column Excel file.

Input payload	Example
Example JSON array payload	`[[123, "abc", "xxx"],[456, "def", "yyyy"],[789, "hij", "zzz"]]`
Example JSON Object payload	`[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]`



Component Properties

Property name	Description
File name	Specify the file name in case of file output

Append	Specifies whether the file is overwritten (Replace) or the data is appended (Append). The value of cp.WriteMode can be one of `"Append"` , `"Overwrite"`
SheetName	Specifies the name of the sheet on which the data is to be output.
Start Column	Specify the first column to output.
Start Row	Specify the first row to output.
Output Header	Select whether headers are also output when outputting data
Output field order	If the input payload is a JSON Object, list the field names to be output in the order in which you want them to be output, separated by commas.

The destination of the file without specifying the file path is actmgr/data, but see here for how to write a file name that specifies the file path.

how to specify output field order

When the payload of a JSON Object is as follows, an Excel file is output if field3 and field1 are specified in the output field order.

Payload of JSON Object

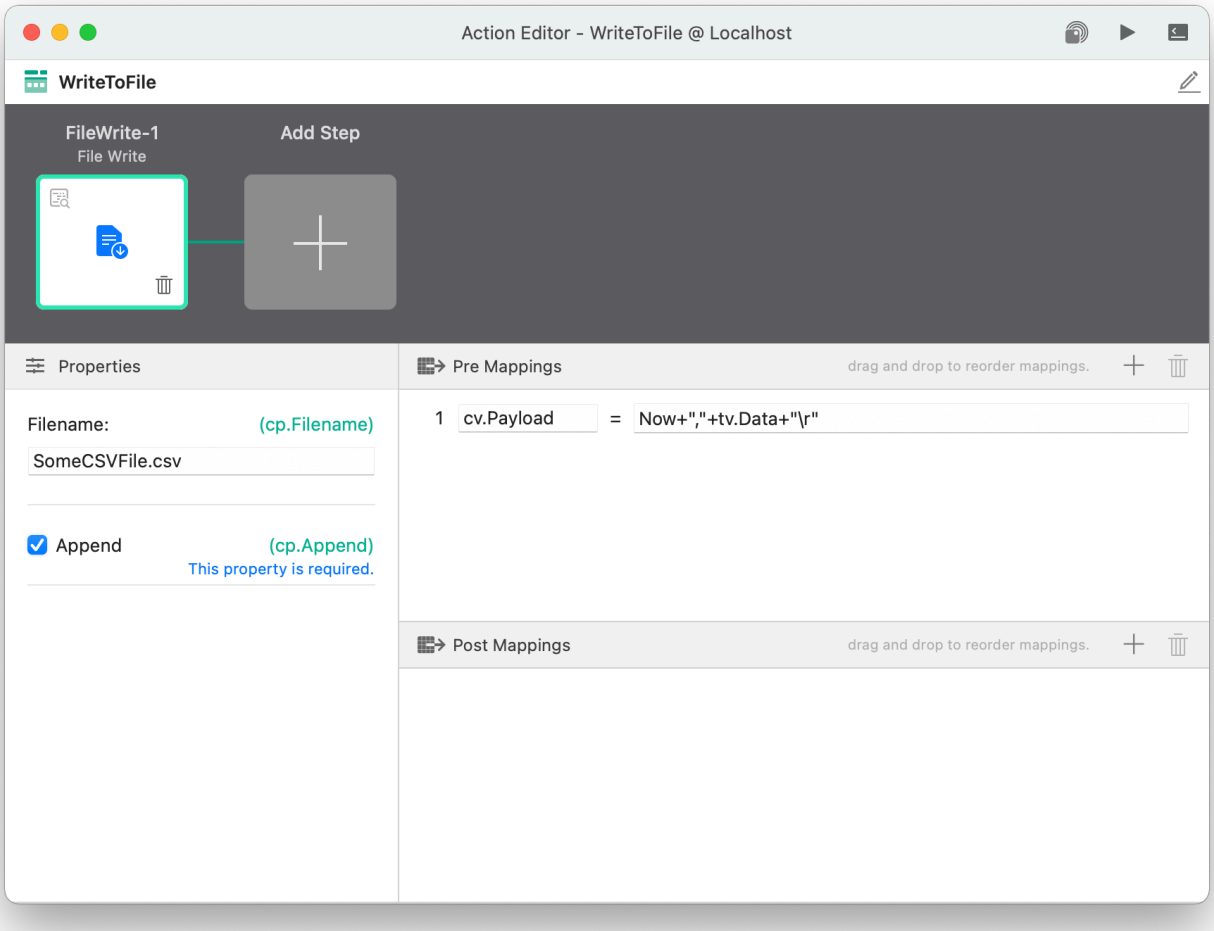
```
[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2":
"def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]
```

Output payload (Excel file)

```
field3, field1
xxx,123
yyy,456
zzz,789
```


6.3.10. Functions

Functions available on the right-hand side of expressions in Pre Mappings / Post Mappings are described separately in this section. A simple example of using a function is the usage of the `Now` function to output the current date, like in this example that writes the timestamp to a local CSV file alongside with the trigger value `tv` Data:



6.3.10.1. String Functions

The following functions are available.

Function Name	Arguments	Description	Example
`Len`	Len(s)	Returns the length of a string (in bytes). s returns the number of elements if s is an array or an object.	Len(“abcdefghijk”) -> 11
`Contains`	Contains(s, substr)	Return by Bool to see if the substring is included in the s	Contains(“abc”, “a”) -> true
`RuneCount`	RuneCount(s)	Returns the number of characters in the s string.	RuneCount(“abc012漢字”) -> 8
`HasPrefix`	HasPrefix(s, prefix)	s begins with a prefix or returns with a Bool	HasPrefix(“https://www.asteria.com/”, “https://”) -> true
`HasSuffix`	HasSuffix(s, suffix)	s ends with a suffix or returns with a Bool	
`Index`	Index(s, substr)	Returns the position of the first occurrence of the substring in s. If not, it is -1. If not, it returns -1	Index(“abcdefg”, “cdef”) -> 3
`Join`	Join(a, sep)	Make the elements of the array “a” into one string with a sep in between.	Join([“abc”,“def”], “-”) -> “abc-def”
`LastIndex`	LastIndex(s, substr)	Returns the position of the last occurrence of the substring in s. If not, it is -1. If not, it returns -1	
`Repeat`	Repeat(s, count)	Returns a string of s repeated count times	Repeat(“abc”, 3) -> “abcabcabc”
`Replace`	Replace(s, old, new[, n])	Replace the old string in s with the new string up to the nth time. n is less than 0 for all (default is -1).	
`Split`	Split(s, sep[, n])	Return an array of s separated by sep to a maximum of n elements, if n is less than 0, all of them (default is -1)	Split(“a,b,c”, “,”) -> [“a”, “b”, “c”]
`ToLower`	ToLower(s)	Convert s to lowercase	
`ToUpper`	ToUpper(s)	Convert s to uppercase	
`Trim`	Trim(s[, cutset])	Returns a string minus the letter at the beginning of the “s” and the characters in the last “cutset”, where “cutset” is a	Trim(” abc “) -> “abc”

		space character by default	
`TrimLeft`	TrimLeft(s[, cutset])	Returns a string excluding the characters in the s-head cutset, where cutset is a space character by default	
`TrimRight`	TrimRight(s[, cutset])	Returns a string except for the characters in the last cutset of s. Cutset defaults to a space character	
`TrimPrefix`	TrimPrefix(s, prefix)	If s begins with a prefix, it returns the string without it. If it doesn't start, it returns s as it is.	TrimPrefix("https://www.asteria.com", "https://") -> "www.asteria.com"
`TrimSuffix`	TrimSuffix(s, suffix)	If the s is terminated by a suffix, return a string with the stripped string. If the s is not terminated, it returns the s as it is.	
`JSONPath`	JSONPath(path, s)	Returns the value as a string, specified by path from `s`.	JSONPath("$.store.book[*].author", cv.Payload)
`Sprintf`	Sprintf(format, s)	Formats a formatted string and returns a string. Formatting can be specified as % followed by a conversion specifier and options flags, width, and precision.	Sprintf("Number of messages: %d", ToInt(cv.Payload))
`IFS`	IFS (condition 1, value 1[, condition 2, value 2, …])	Evaluates the conditions in the order of the parameters from the 1st parameter and returns the respective value that matches. If you specify `true` for the condition, it will always match, therefore if you specify `true` for the last condition-value pair, you can specify a condition matching everything else.	IFS (ToFloat(cv.Payload) >= 10, "more than 10", true, "Less than 10")

6.3.10.1.1. JSONPath() Usage Examples

This section describes an example of using JSONPath().

If the sensor data (`cv.Payload`) or the argument of the action to be executed in the trigger condition (`tv.Data`) is in JSON format, you can use JSONPath() to refer to the value.

Sample Data

```
{ "store": {
    "book": [
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      { "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
      },
      { "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  }
}
```

For example, if you want to refer to store -> bicycle -> color , use

```
cv.Payload.store.bicycle.color
```

But if you use JSONPath() you can write

```
JSONPath("$.store.bicycle.color", cv.Payload)
```

It can also be written as

The first argument of JSONPath path

path	Description
$	root object/element
@	current object/element
. or []	child operator
…	recursive descending
*	wildcard
[]	subscript operator (native array operator)
[,]	union operator (alternative names or array indices can be used as sets)
[start:end:step]	array slice operator
?()	filter (script) expression
()	Script expression using the underlying script engine

Example of the first argument path of JSONPath

Example	Description
$.store.book[*].author	all authors in the books array in the store
$..author	all authors
$.store.*	the books array in the store and bicycle
$.store..price	all prices in the store
$..book [2]	the third book
$..book[(@.length-1)]$..book[-1:]	last book
$..book[0,1] $..book [:2]	first two books in book array
$..book[?(@.isbn)]	all isbn in book array
$..book[?(@.price<10)]	price less than 10 in book array
$..*	all nodes/elements

For more information on JSONPath, see the Goessner.net reference here.

6.3.10.1.2. Using Sprintf

Sprintf formats a formatted string and returns the string. The format can be specified by % followed by a conversion specifier and optionally a flag, width, and precision. Decimal

Specifier	Description
`%b`	Binary integer
`%d`	Decimal integer
`%x %X`	Hexadecimal
`%e %E`	Floating point, scientific notation
`%f %F`	Floating point, no exponent notation
`%g %G`	Floating point, %e if the exponent is large, %f otherwise
`%s`	String
`%%`	The character of % itself

Flag	Description
`+`	Signed
`-`	Left justified
`0`	If it is less than the width, fill it with 0 instead of space (left padding)

Length and Accuracy	Description
`%f`	Default length
`%9f`	Length 9
`%.2f`	Accuracy 2
`%9.2f`	length 9 and accuracy 2

Examples

```
Sprintf("Messages: %d", ToInt(cv.Payload))
Sprintf("Floating point to the second decimal place: %.2f", cv.Payload)
```

6.3.10.2. Type Conversion Functions

The following functions are available.

Function Name	Arguments	Description	Example
`ToBool`	ToBool(a)	Conversion to Bool type	
`ToInt`	ToInt(a)	Conversion to Integer type	
`ToFloat`	ToFloat(a)	Conversion to Floating-Point type	
`ToString`	ToString(a)	Conversion to String type	
`ToDate`	ToDate(a)	Conversion to Date and time	
`ToBinary`	ToBinary(a)	Conversion to Byte sequence	
`ToJSON`	ToJSON(a)	Conversion to JSON array type or JSON object type	split a string of numbers separated with `,` at `,` and turn the result into a JSON object so it can be processed with JSONPath to find out if any of the values exceeds 30 (for example in the trigger with the `Expr` setting): Len(JSONPath(“$[?@>30]”, ToJSON(Split(tv.Data, “,”, “-1”)))) > 0

6.3.10.3. Encode Functions

The following functions are available.

Function Name	Arguments	Description	Example
URLPathEscape	URLPathEscape(s)	Escaping s to a string that can be safely placed in a URL path segment	URLPathEscape(“test1?test2 test3”) -> “test1%3Ftest2%20test3”
URLPathUnescape	URLPathUnescape(s)	Undo any escaped string with URLPathEscape	
URLQueryEscape	URLQueryEscape(s)	Escaping s to a string that can be safely placed in a URL query	URLQueryEscape(“test1?test2 test3”) -> “test1%3Ftest2+test3”
URLQueryUnescape	URLQueryUnescape(s)	Undo the escaped strings with URLQueryEscape	
MD5	MD5	Return MD5 in byte sequence b	
SHA1	SHA1	Return SHA1 of byte sequence b	
SHA256	SHA256	Return SHA256 of byte sequence b	
BASE64	BASE64	Returns BASE64 of the byte sequence b. If true is specified with pad, =padding is performed (the default is true)	
DecodeBASE64	DecodeBASE64(s[, pad])	Decode BASE64s. pad=true = BASE64 with padding (the default is true)	
BASE64URL	BASE64URL	Returns the BASE64 used in the URL or file name for the byte sequence b. If true is specified in the pad, =padding is performed (the default is true)	
DecodeBASE64URL	DecodeBASE64URL(s[, pad])	Decode BASE64s. pad=true = BASE64 with padding (the default is true)	

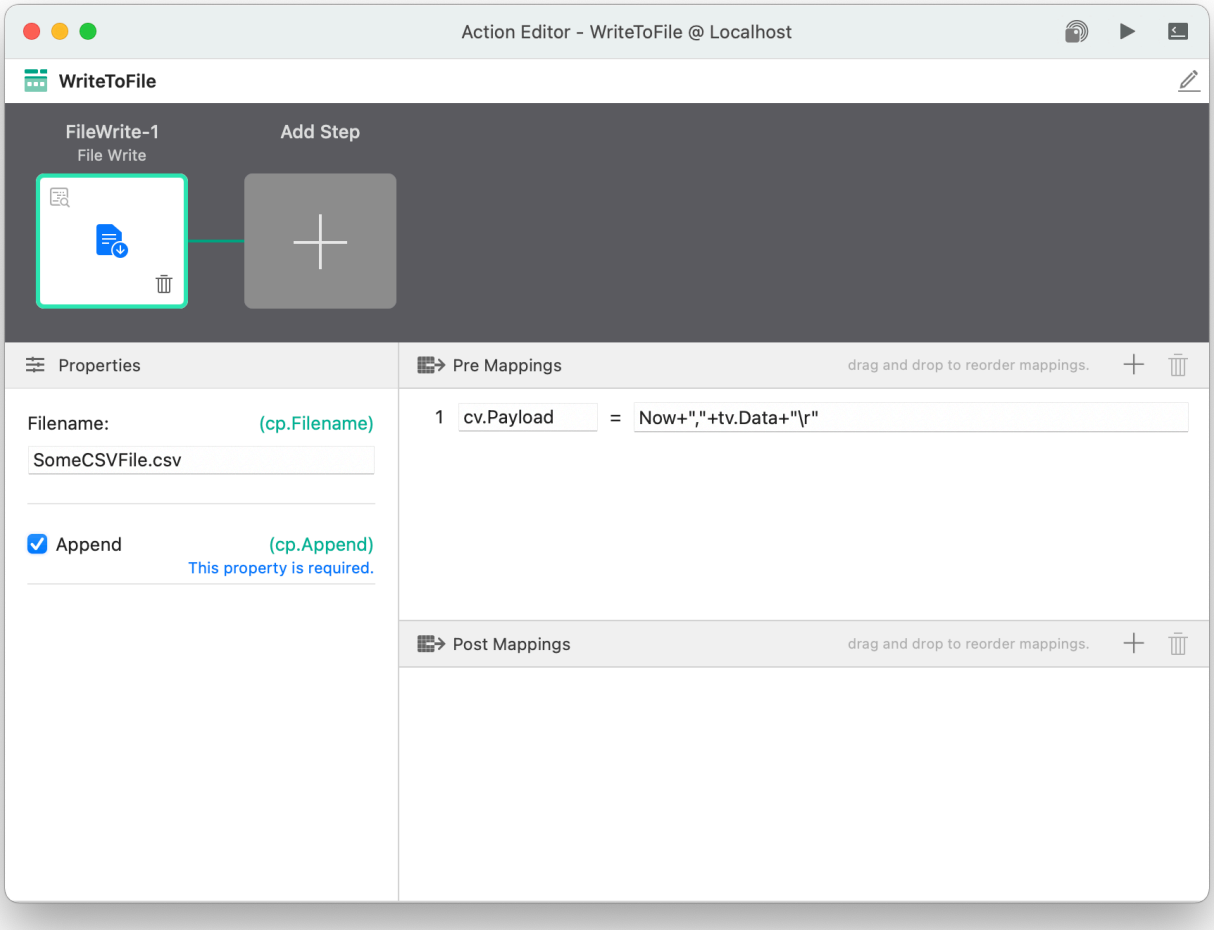
6.3.10.4. Date Functions

The following functions are available.

Function Name	Arguments	Description	Example
`Now`	Now([timezone])	Returns the date and time type for the current date and time.	
`Year`	Year(t[, timezone])	Returns the year of date and time type `t`. If timezone is specified, it is interpreted as the time zone.	
`Month`	Month(t[, timezone])	Returns the month of date and time type `t`. If timezone is specified, it is interpreted as the time zone.	
`Day`	Day(t[, timezone])	Returns a day of type `t`. If timezone is specified, it is interpreted as the time zone.	
`Hour`	Hour(t[, timezone])	Return the time of date and time type `t`. If timezone is specified, it is interpreted as the time zone.	
`Minute`	Minute(t[, timezone])	Return the minutes of date and time type `t`. If timezone is specified, it is interpreted as the time zone.	
`Second`	Second(t[, timezone])	Return seconds of date and time type `t`. If timezone is specified, it is interpreted as the time zone.	
`Weekday`	Weekday(t[, timezone])	Return the day of the week for date and time type `t`. If timezone is specified, it is interpreted as the time zone. (0 is Sunday.)	
`ToLocal`	ToLocal(t)	Convert date and time type `t` to system time zone	
`ToUTC`	ToUTC(t)	Convert date and time type `t` to UTC	
`ToTimezone`	ToTimezone(t, timezone)	Converts a date and time type `t` to a specified time zone	`ToTimezone(Now(), "America/New_York")`
`DateFormat`	DateFormat(t, layout)	Convert a date and time type `t` to a string with the specified layout. The key date that you can rearrange as you desire is `2006-01-02 15:04:05.999 MST` with `01` representing the month, `02` representing the day, `15` representing the hours, `04` representing the minutes `05` representing the seconds, `2006` representing the year in four digits and `999` representing the microseconds. MST is the timezone (Mountain Standard Time). This is based on the GoLang date formatting standard, which you can look up here .	Example to output the format Day Month Year Hour:Minute Timezone `DateFormat(Now(), "02 Jan 06 15:04 MST")`

`DateParse`	DateParse(layout, s[, timezone])	Interprets `s` along the layout and returns a date/time type	
`ToInt`	ToInt(timestamp)	Turn the timestamp into a unix timestamp	`ToInt(cv.Payload.Timestamp)`

Example usage of the `Now` function:



Gravio uses GoLang Date formatting. More details can be found [in the GoLang date format source code file](#).

6.3.10.5. Environment Functions

The following functions are available.

Function Name	Arguments	Description
`Env`	Env(s)	Return environment variables

6.3.10.6. Regular Expression Functions

The following functions are available.

Function Name	Arguments	Description	Example
`RegExpMatch`	RegexMatch(re, s)	Returns a Bool to see if the regular expressions re and s match	RegexMatch(“G.*o”, “Gravio HubKit”) -> true
`RegExpFind`	RegexFind(re, s)	Returns the first string that matches the regular expressions re and s	RegexFind(“i.”, “Gravio HubKit”) -> “io”
`RegExpFindSubmatch`	RegexFindSubmatch(re, s)	If the first string and parenthesized subexpressions of the regular expressions re and s match, return an array	RegexFindSubmatch(“(G.*o)(H.*t)”, “Gravio HubKit”) -> [“Gravio HubKit”, “Gravio”, “HubKit”]
`RegExpFindAll`	RegexFindAll(re, s[, n])	Returns an array of all strings where the regular expressions re and s match	RegexFindAll(“i.”, “Gravio HubKit”) -> [“io”, “it”]
`RegExpReplace`	RegexReplace(re, s, repl)	Returns a string with the regular expressions re and s replaced by a matched repl. repl can use subexpressions such as `$1`.	
`UUID`	UUID	Returns a UUID string, where sep is a separator (empty string by default)	

6.3.10.7. Arithmetic Functions

The following functions are available:

Function Name	Arguments	Description	Example
`Abs`	Abs(v)	Returns absolute value	
`Acos`	Acos(v)	Returns the inverse cosine	
`Acosh`	Acosh(v)	Returns the inverse hyperbolic cosine	
`Asin`	Asin(v)	Returns the inverse sine	
`Asinh`	Asinh(v)	Returns the inverse hyperbolic sine	
`Atan`	Atan(v)	Returns the inverse tangent	
`Atan2`	Atan2(y, x)	Returns the inverse tangent of y/x	
`Atanh`	Atanh(v)	Returns the inverse hyperbolic tangent	
`Cbrt`	Cbrt(v)	Return cubic root	
`Ceil`	Ceil(v)	Returns the value rounded up to the nearest whole number	
`Copysign`	Copysign(x, y)	Returns a value of sign y with magnitude x	
`Cos`	Cos(v)	Returns the cosine	
`Cosh`	Cosh(v)	Returns the hyperbolic cosine	
`Erf`	Erf(v)	Returns the error function	
`Erfc`	Erfc(v)	Returns the complementary error function	
`Exp`	Exp(x)	Returns e raised to the power x	
`Exp2`	Exp2(x)	Returns 2 raised to the power x	
`Expm1`	Expm1(x)	Returns e to the power of x, minus 1	
`Floor`	Floor(v)	Returns the value rounded down to the nearest whole number	
`Hypot`	Hypot(x, y)	Returns Sqrt(xx + yy)	
`Inf`	Inf(sign)	Returns positive infinity if sign>=0, otherwise returns negative infinity	
`IsInf`	IsInf(v, sign)	Returns true if v is infinity according to the sign	
`IsNaN`	IsNaN(v)	Returns true if v is Not a Number	
`Log`	Log(v)	Returns the natural logarithm	

`Log10`	Log10(v)	Returns the base-10 logarithm	
`Log1p`	Log1p(v)	Returns the natural logarithm of v + 1	
`Log2`	Log2(v)	Returns the base-2 logarithm	
`Max`	Max(v1[, v2, v3, …])	Returns the maximum value among the specified values	
`Min`	Min(v1[, v2, v3, …])	Returns the minimum value among the specified values	
`NaN`	NaN()	Returns Not a Number	
`PI`	PI()	Returns the value of pi	
`Pow`	Pow(x, y)	Returns x raised to the power y	
`Round`	Round(v[, n])	Returns the value rounded to the nearest whole number. If n is specified, it rounds to the nth decimal place	
`Signbit`	Signbit(v)	Returns true if the value is negative	
`Sin`	Sin(v)	Returns the sine	
`Sinh`	Sinh(v)	Returns the hyperbolic sine	
`Sqrt`	Sqrt(v)	Returns the square root	
`Random`	Random([max])	Returns a random number. If no argument is provided, it returns a floating-point number between 0 and 1. If an argument is provided, it treats it as the maximum value and returns an integer between 0 and that maximum value	
`Tan`	Tan(v)	Returns the tangent	
`Tanh`	Tanh(v)	Returns the hyperbolic tangent	
`Trunc`	Trunc(v[, n])	Returns the integer part of the value. If n is specified, it truncates at the nth decimal place	
`Degrees`	Degrees(v)	Converts radians to degrees	
`Radians`	Radians(v)	Converts degrees to radians	
`FtoC`	FtoC(v)	Converts Fahrenheit to Celsius	
`CtoF`	CtoF(v)	Converts Celsius to Fahrenheit	
`GtoLBM`	GtoLBM(v)	Converts grams to pounds	
`LBMtoG`	LBMtoG(v)	Converts pounds to grams	
`GtoOZM`	GtoOZM(v)	Converts grams to ounces	
`OZMtoG`	OZMtoG(v)	Converts ounces to grams	
`MtoIN`	MtoIN(v)	Converts meters to inches	

`INtoM`	INtoM(v)	Converts inches to meters	
`MtoFT`	MtoFT(v)	Converts meters to feet	
`FTtoM`	FTtoM(v)	Converts feet to meters	

7. Using the Gravio AI Option

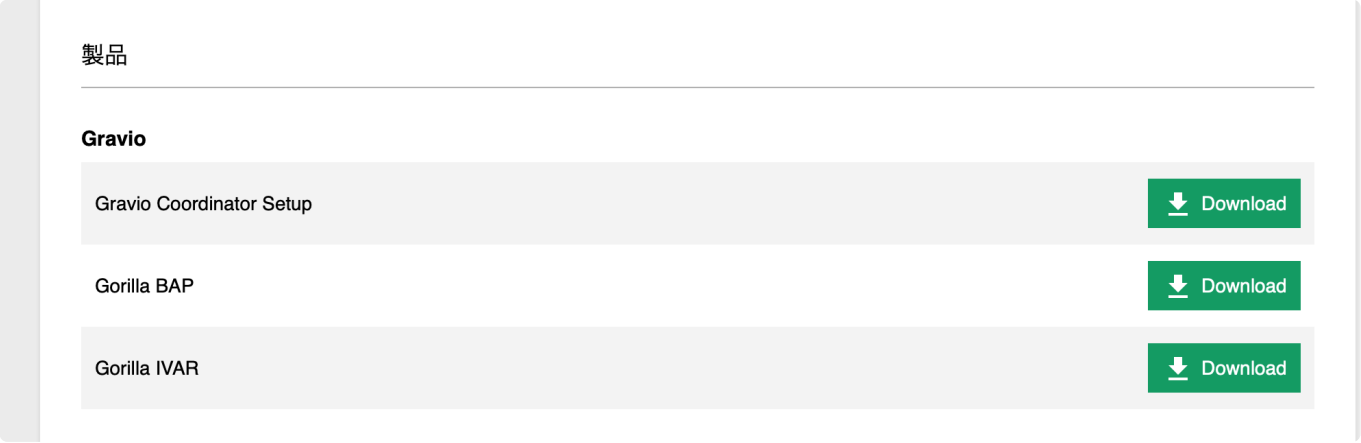
This section describes how to install and use the Gorilla IVAR (Intelligent Video Analytics Recorder) and Gorilla BAP (Biometric Authentication Provider) provided by Gorilla Technology for Gravio Enterprise AI Edition Gold / Platinum.

Gorilla Technology is a privately held company established in 2000, that specializes in video intelligence and IoT technology, supporting a wide range of video-centric and content management applications for retail, enterprise and surveillance. Their machine learning and deep learning video analysis algorithms identifies, analyses and extracts information from digital content to drive business intelligence solutions and automation.

7.1. Gorilla IVAR and BAP Modules

Gorilla IVAR (Intelligent Video Analytics Recorder), BAP (Biometric Authentication Provider) installer and documentation can be found on your Asteria customer portal, depending on your agreement and license.

If you have access, after logging in and selecting the “Download” tab, you should see the following view:



Each download button is displayed to the right of the solution available. Both IVAR and BAP can be downloaded in zip file format and each zip file contains the installer and documentation.

7.2. Installation of Gorilla BAP

Things to prepare in advance

・BAP account Root, Administrator, Application user password
・BAP license key

Installation environment

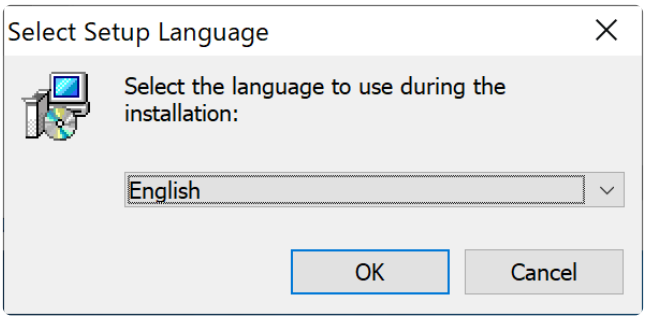
Prepare a Windows 10 PC and Chrome browser to install BAP.
You will also need to be connected to the internet as you will be registering a license.

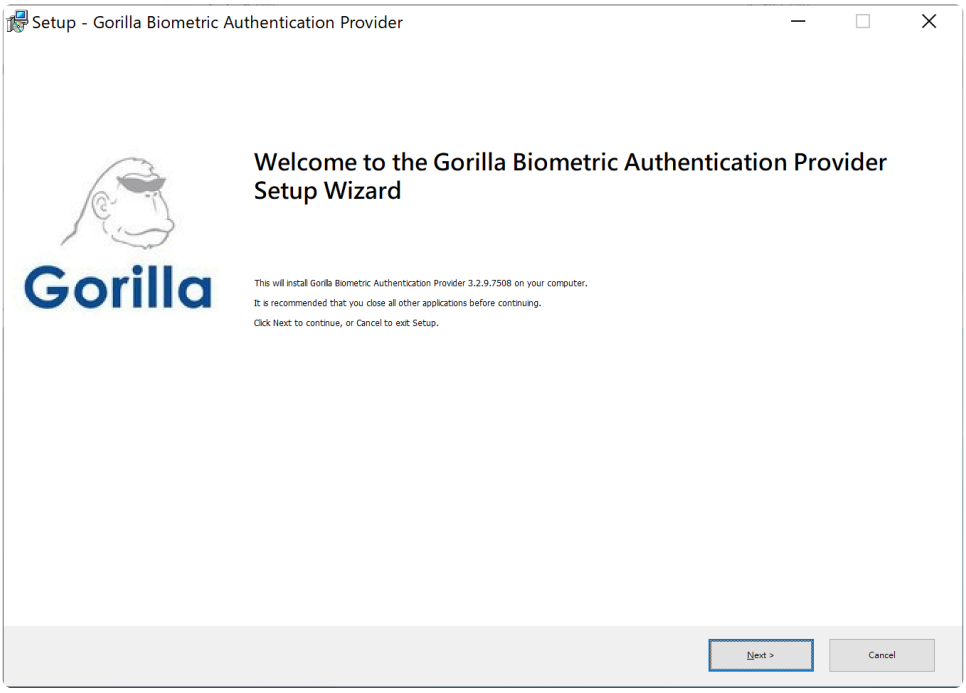
Notes before installing

BAP uses PostgreSQL. If PostgreSQL is already installed, it may not be installed correctly due to conflicts such as version differences, therefore please uninstall any previous PostgreSQL installation you may have. Also delete the data folder.

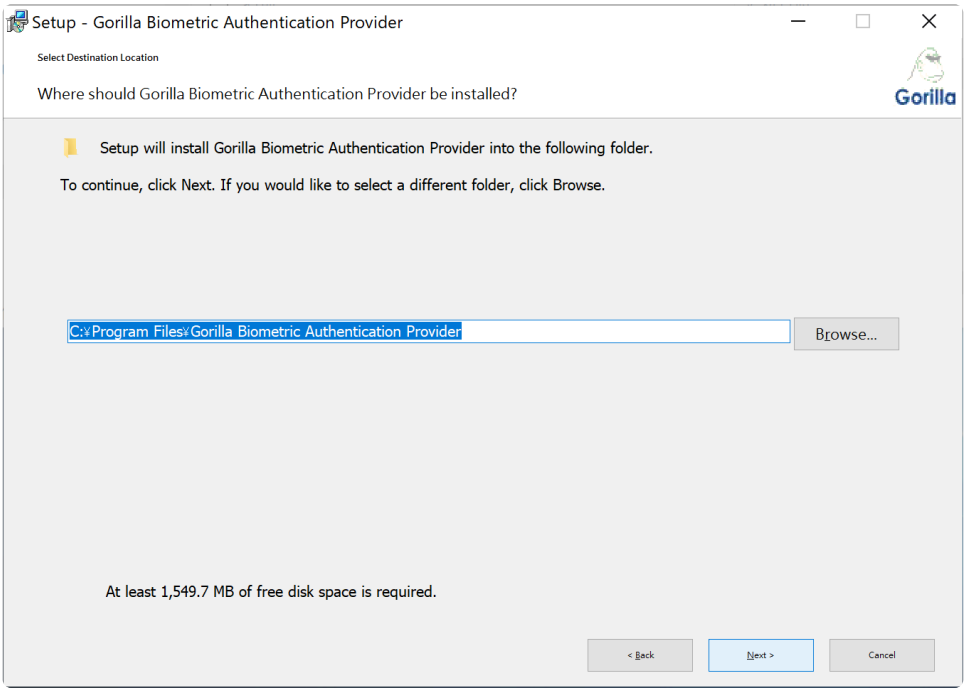
Installation procedure

Unzip the downloaded zip. When expanded, a BAP folder will be created. Inside you will find the `Gorilla Biometric Authentication Providor_<version number>.Setup` application. Please run it.

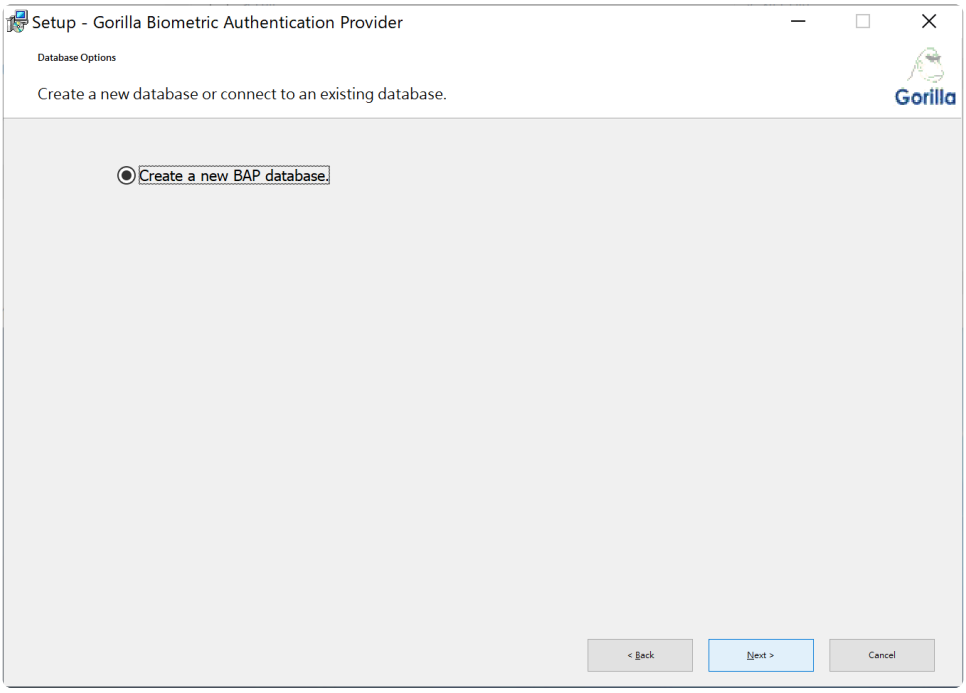




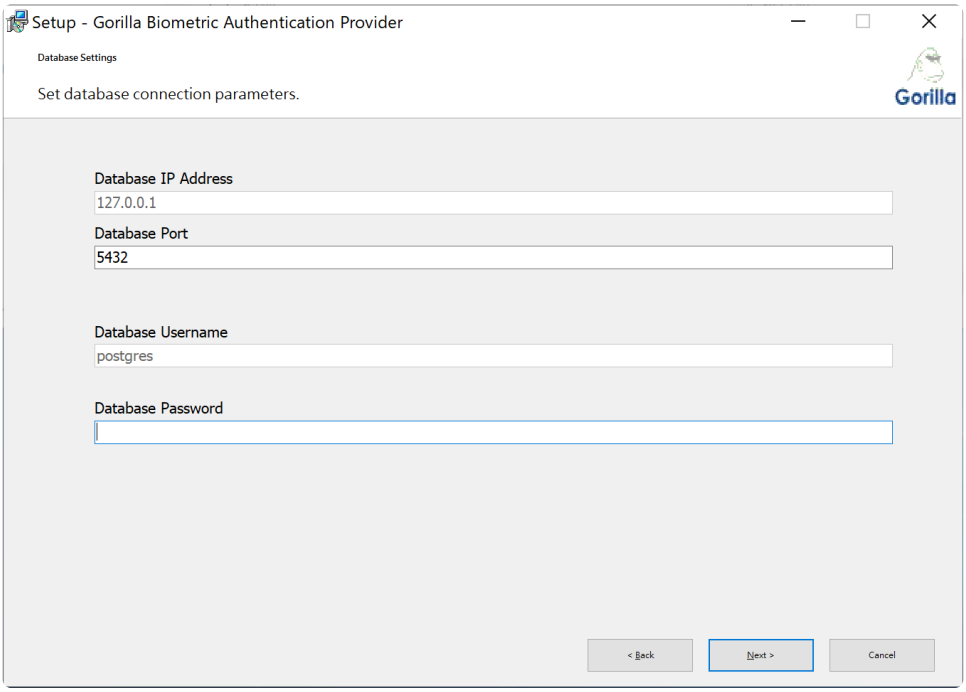
Click the Next button to proceed with the installation.



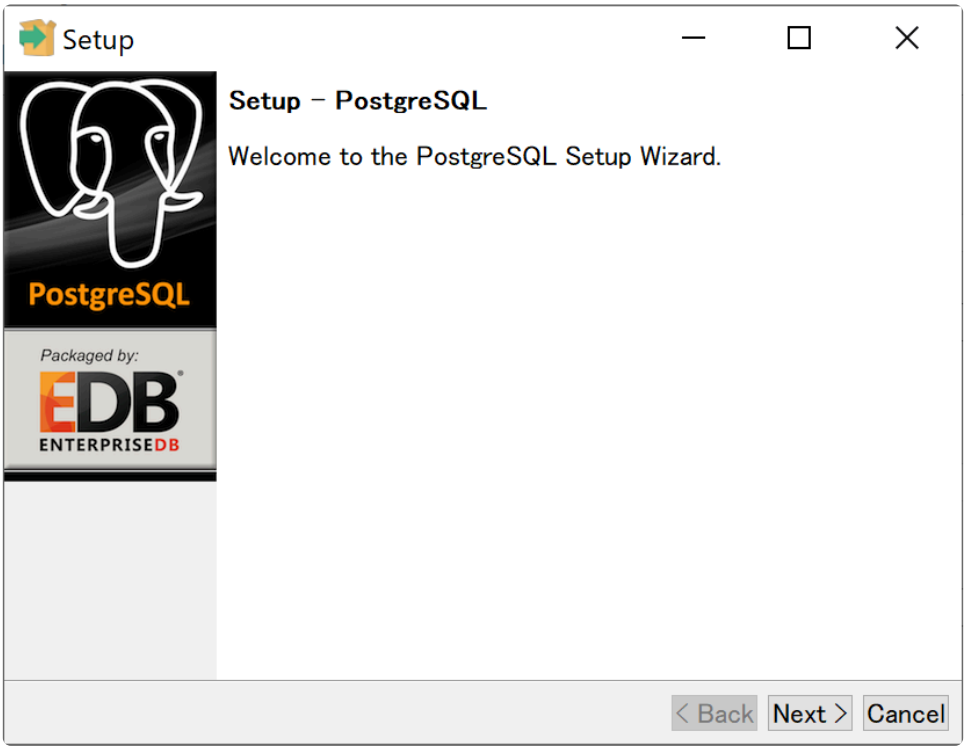
Please click Next button to install in the standard folder.



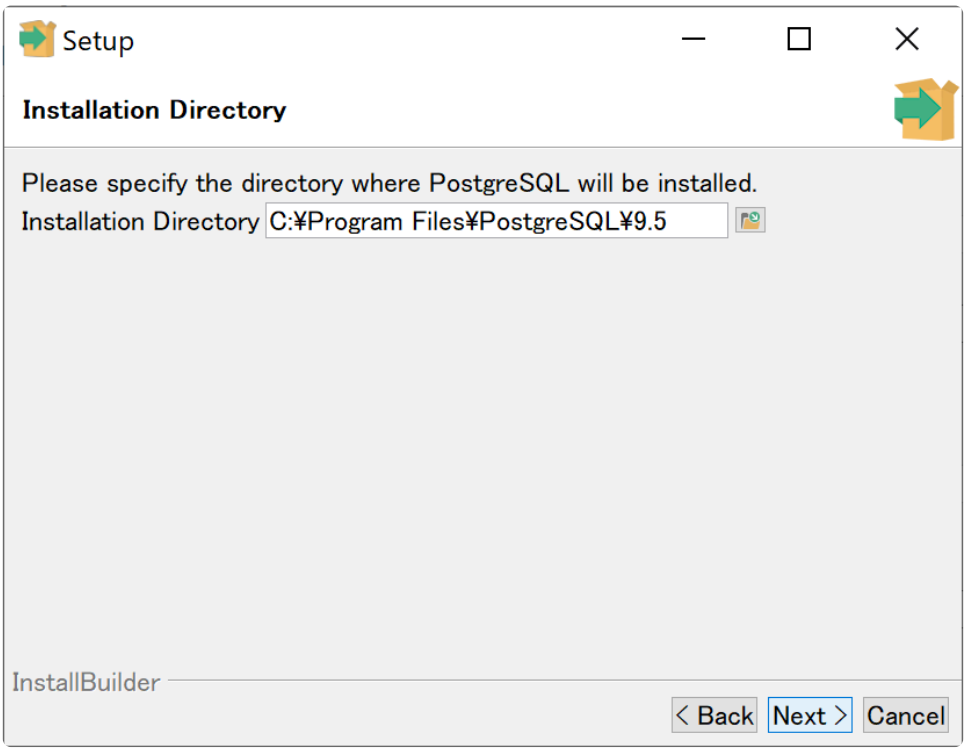
Press the Next button.



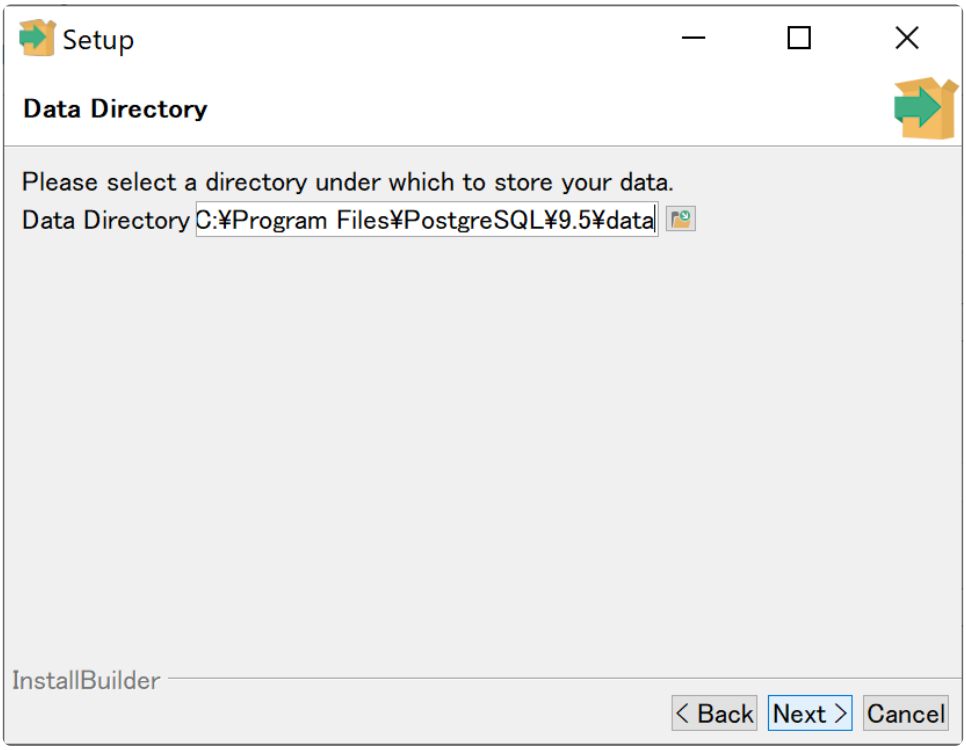
For the Database Password, set the PostgreSQL password that will be set later. Click the Next button to start the PostgreSQL installer.



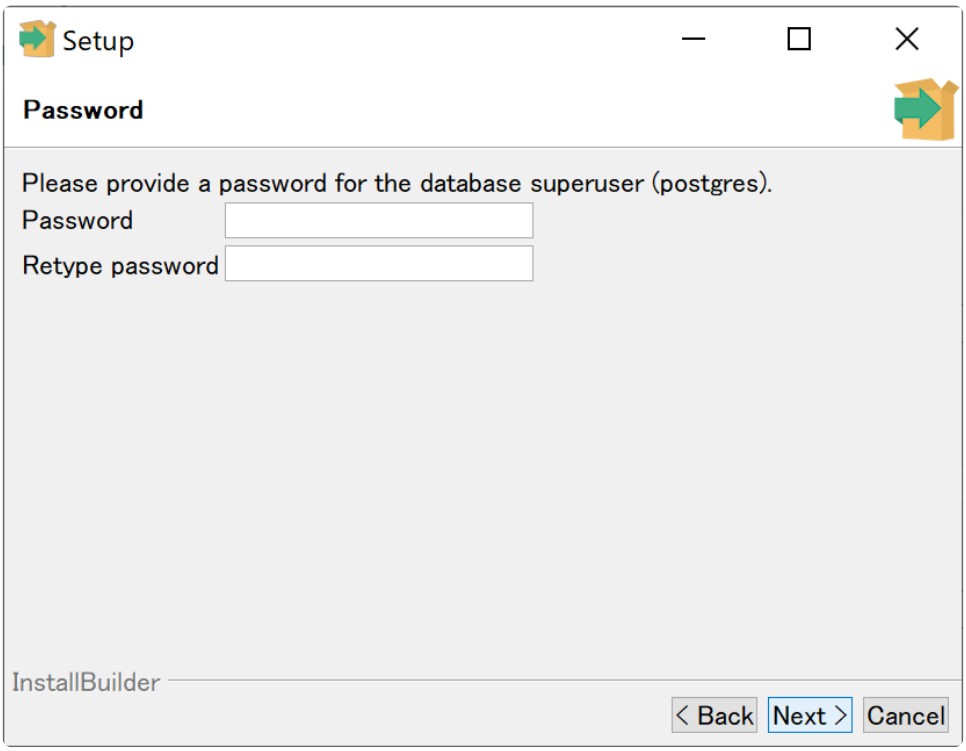
Press the Next button.



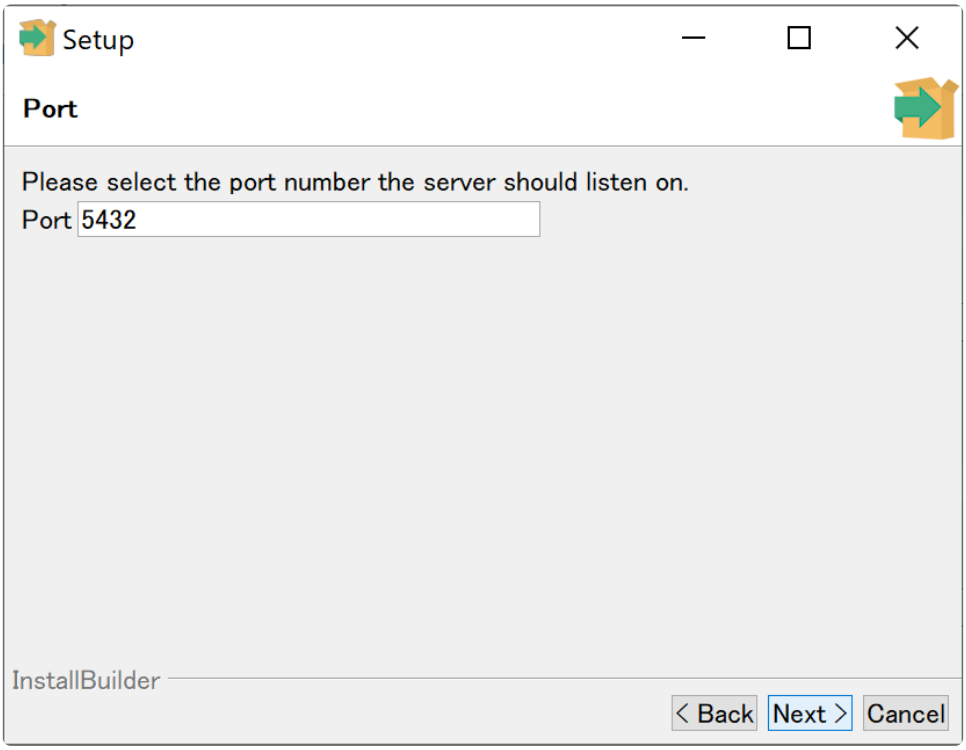
Please click the Next button to install in the standard folder.



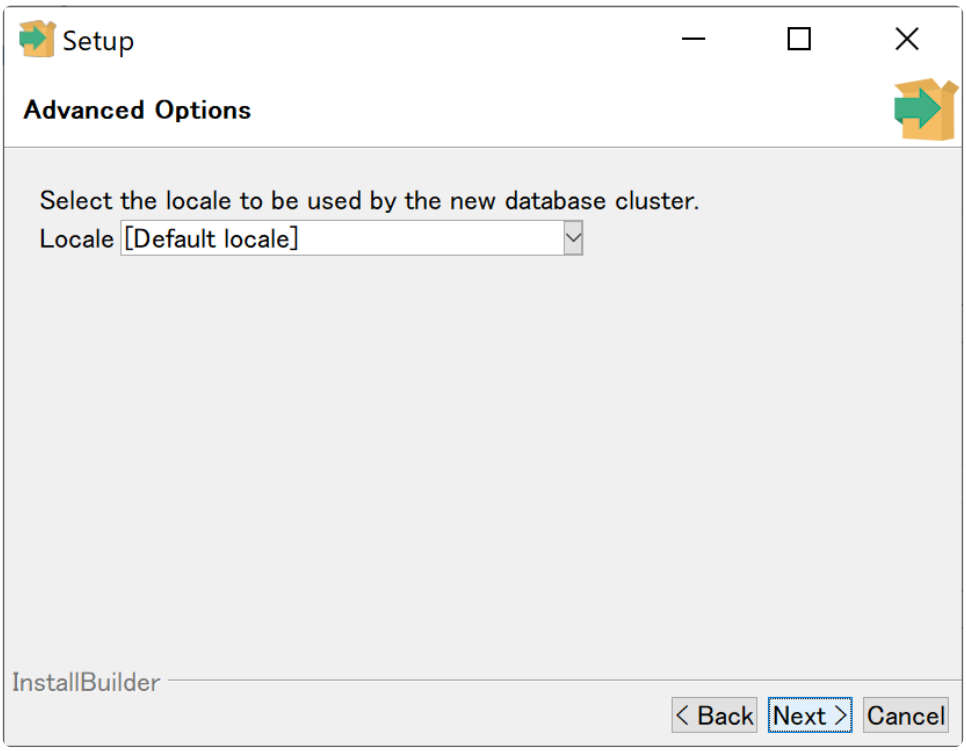
Please click the Next button to install in the standard folder.



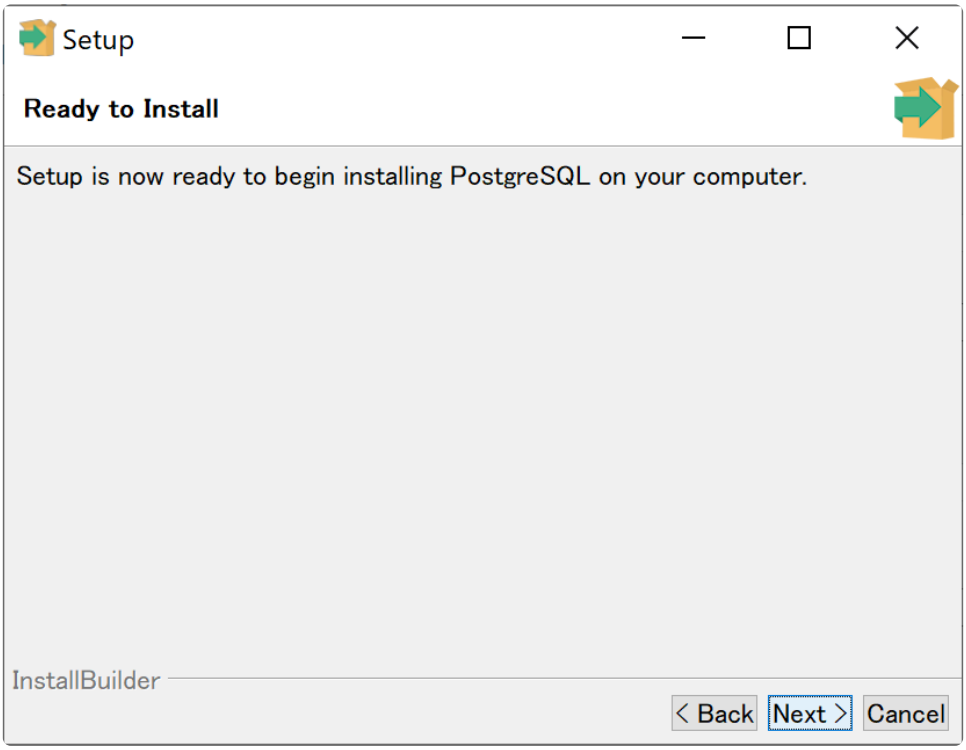
Register the PostgreSQL password you registered earlier and click the Next button.



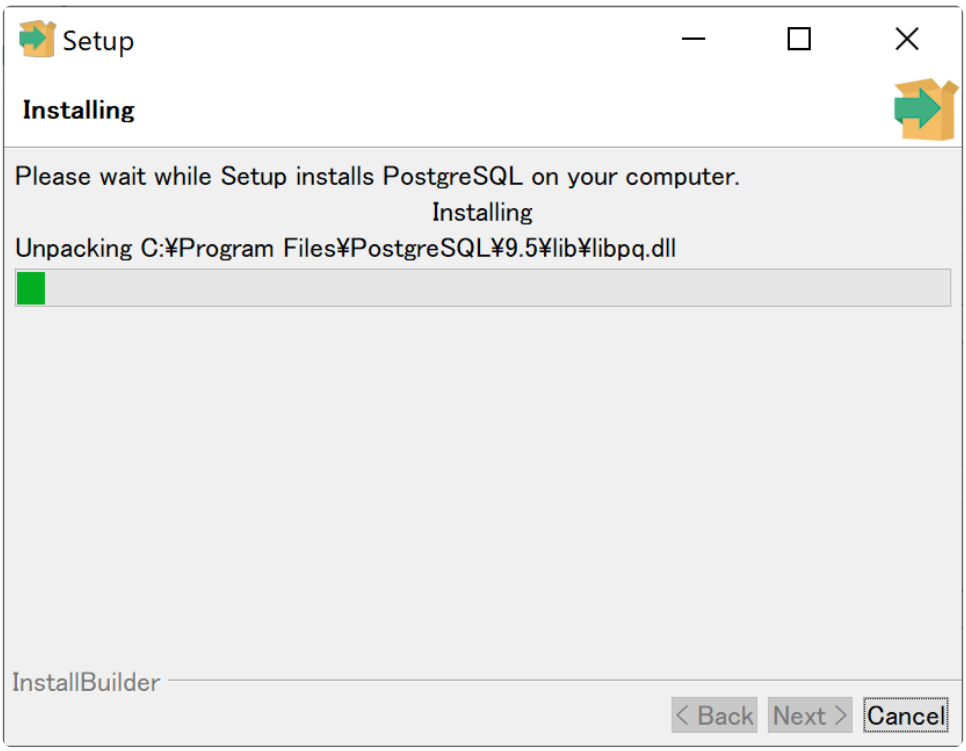
Since the standard port is used, just press the Next button.



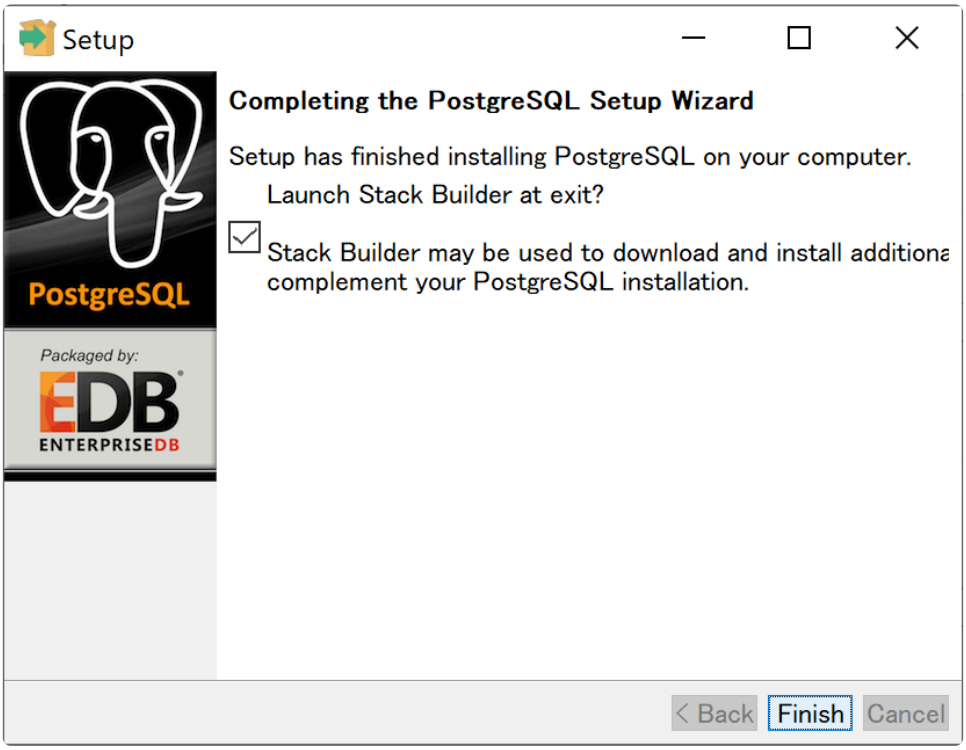
Press the Next button without making any changes.



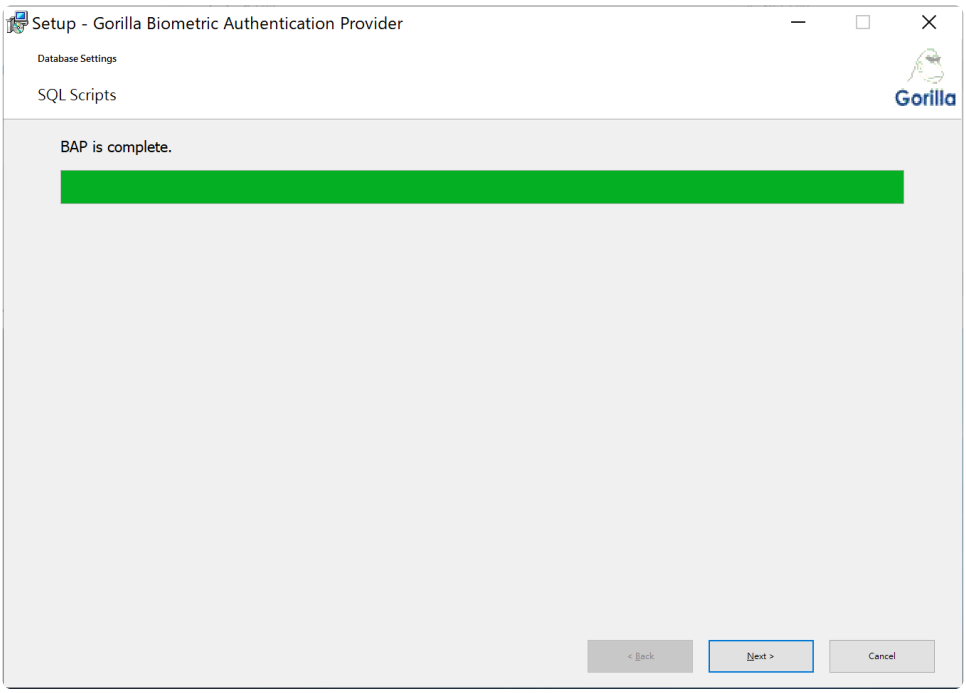
Press the Next button.



The file copy will start.

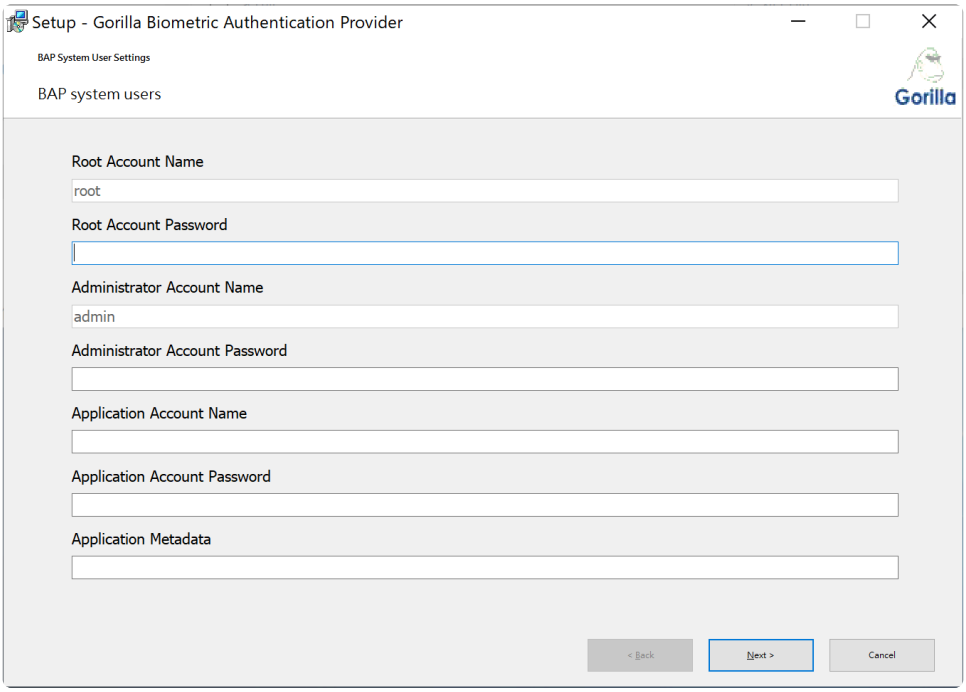


When the installation is complete, uncheck “Stack Builder may be used …” and press the Finish button.



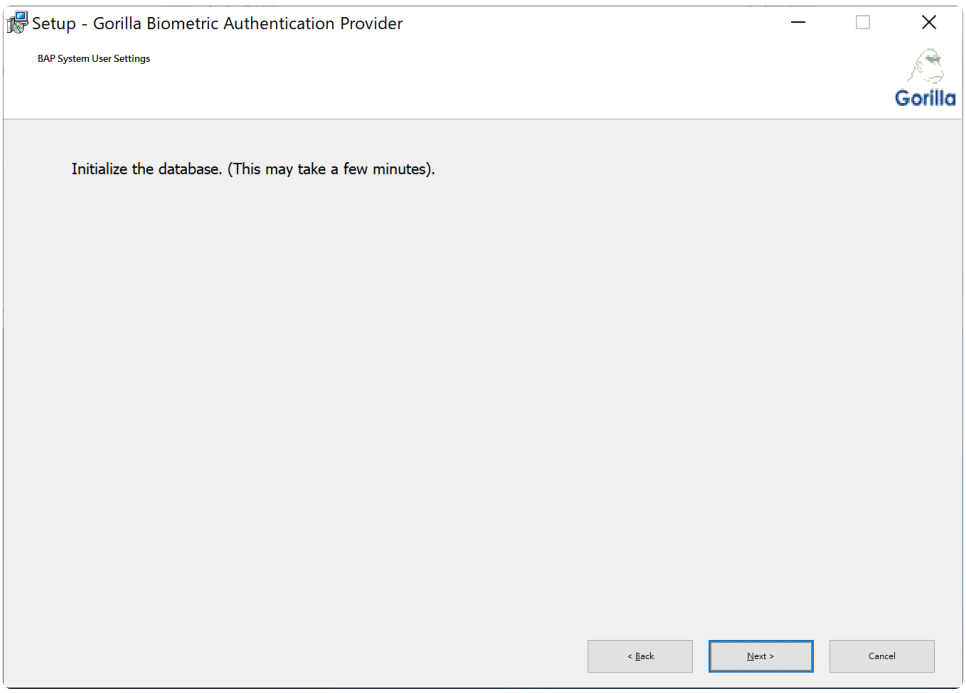
After the installation of BAP is completed, the initial settings will be continued. Click the Next button.

Initial settings

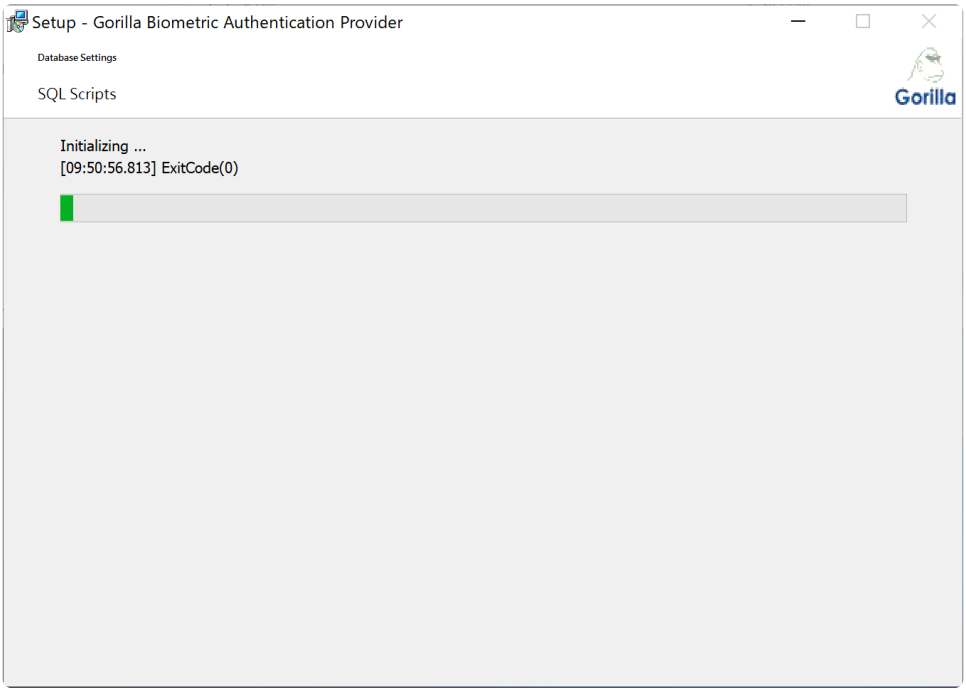


Create three users in BAP: Root, Administrator, and Application.
The Admin account is used for the operation.

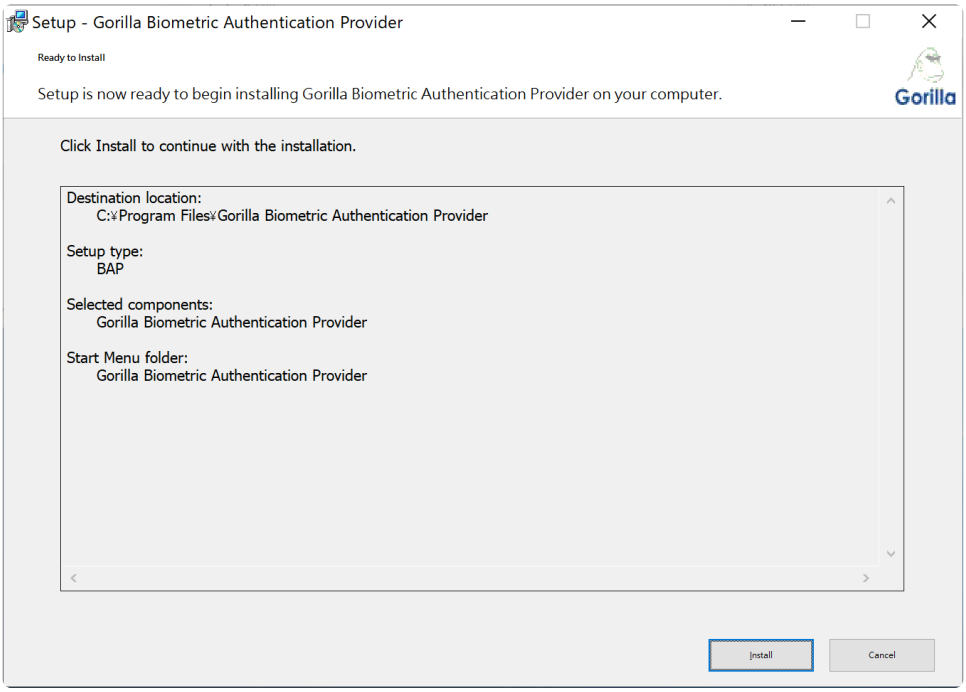
Register for Root Account Password, Administrator Account Password, and Application Account Password respectively, and click the Next button.



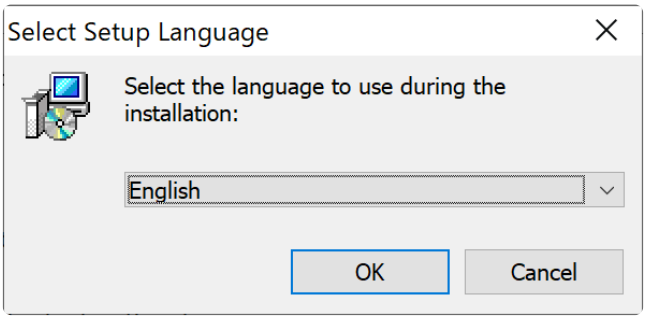
Press the Next button.



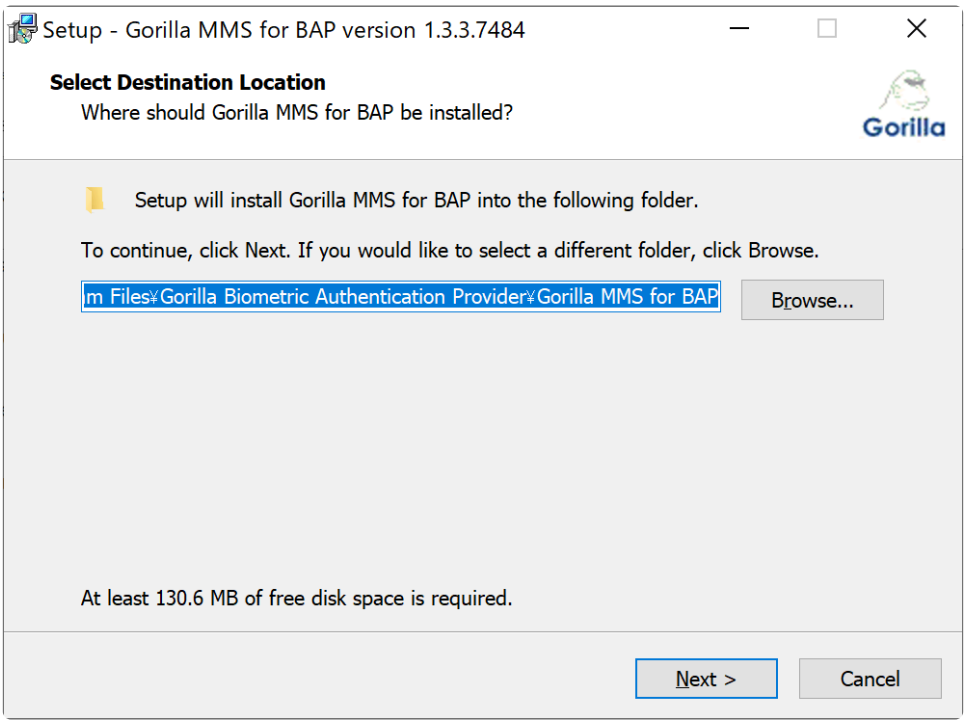
The file copy will start.



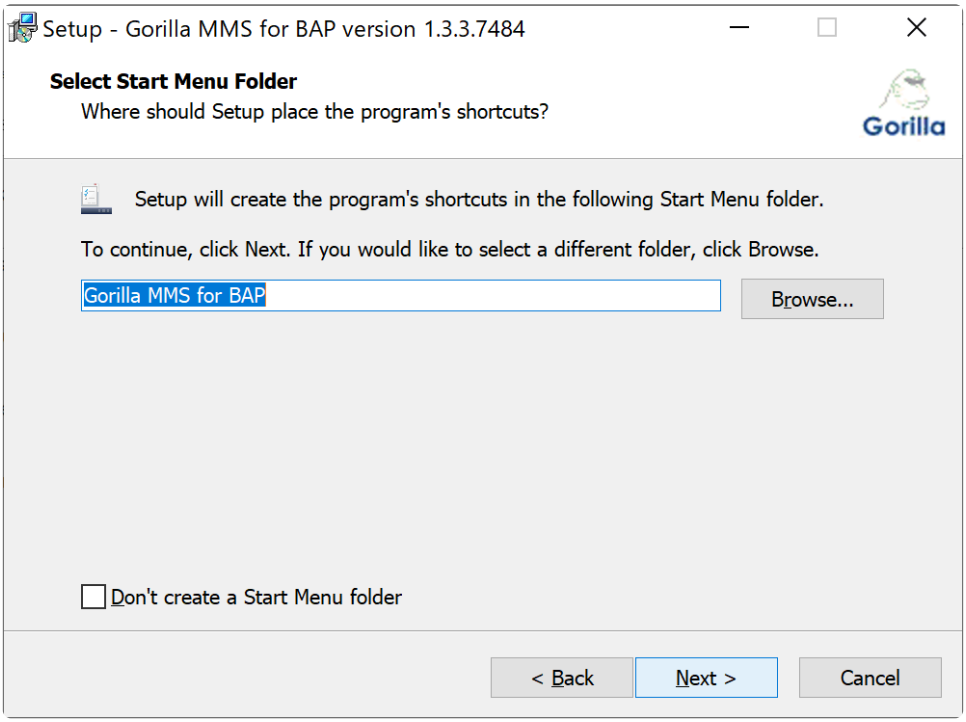
Press the Install button. The MMS installer will start.



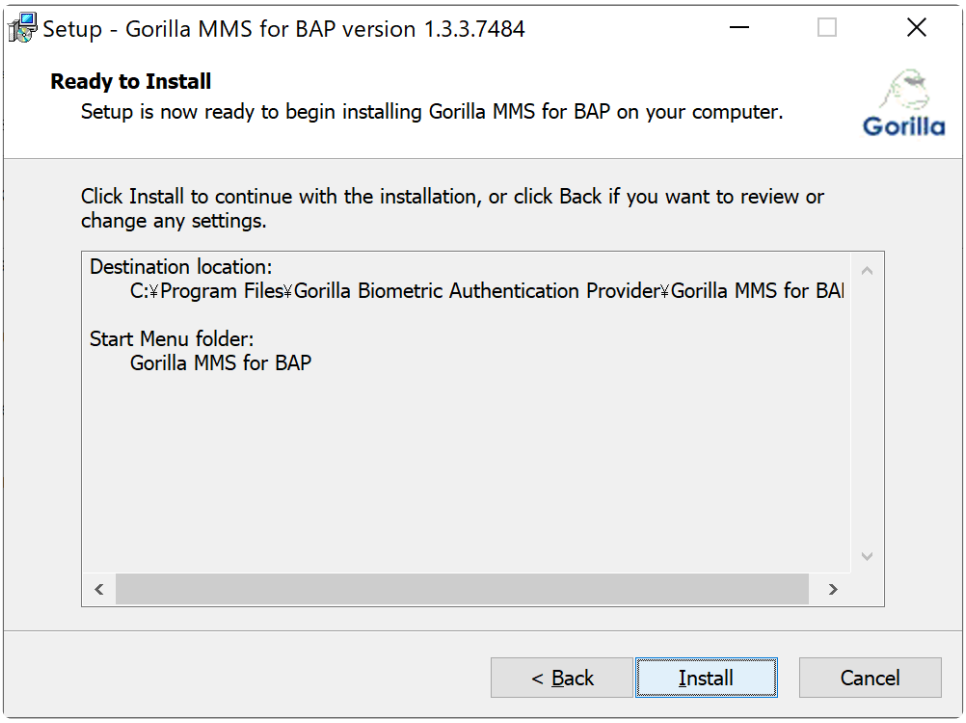
Press the OK button.



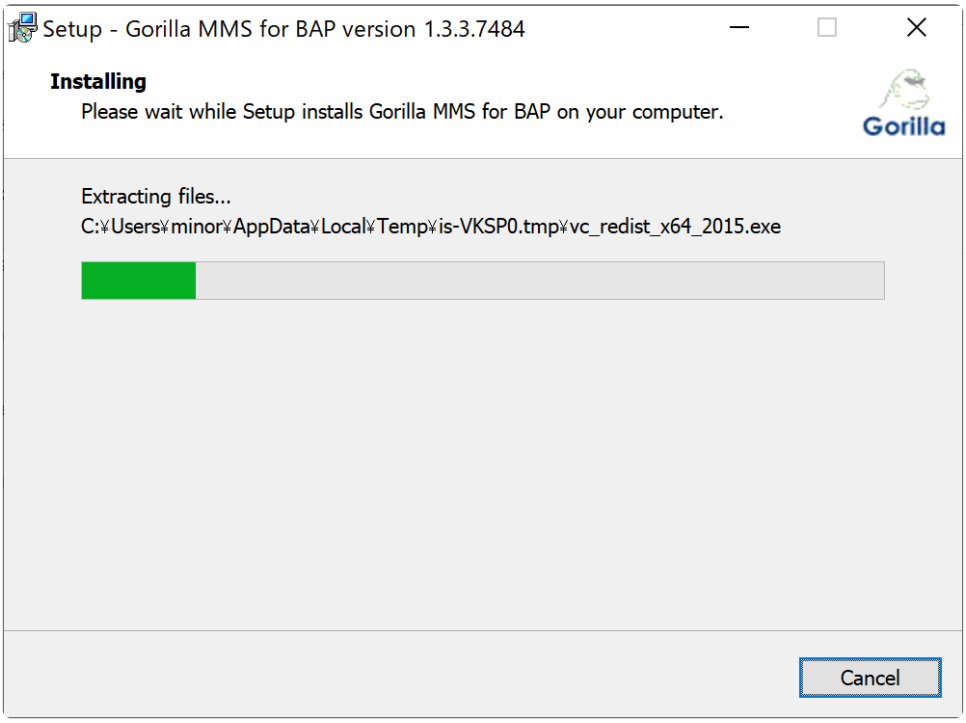
Please click the Next button to install in the standard folder.



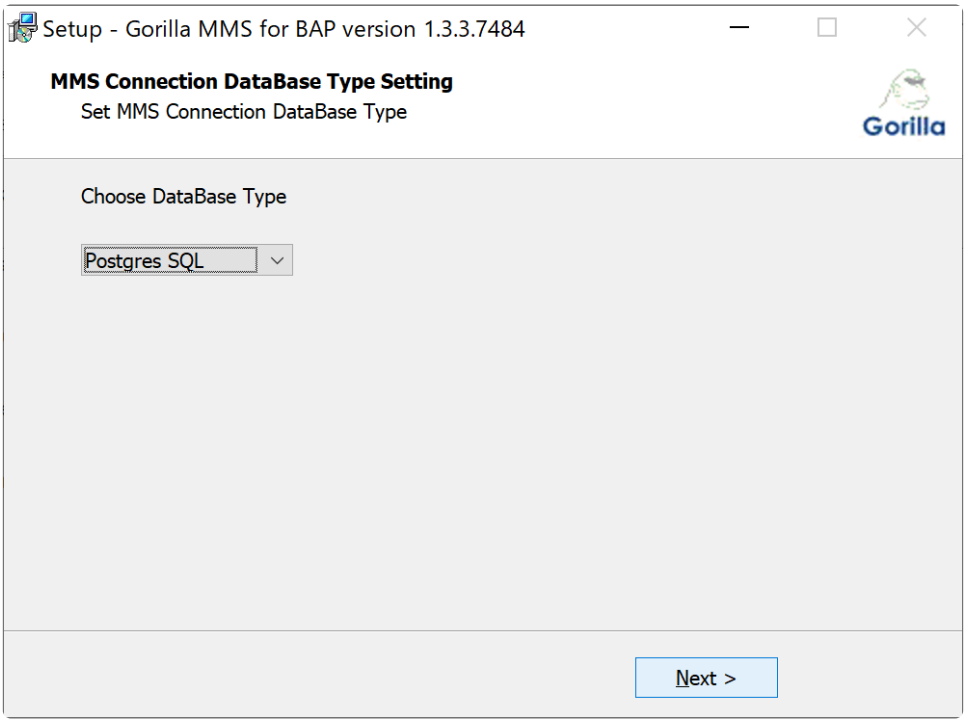
Just press the Next button.



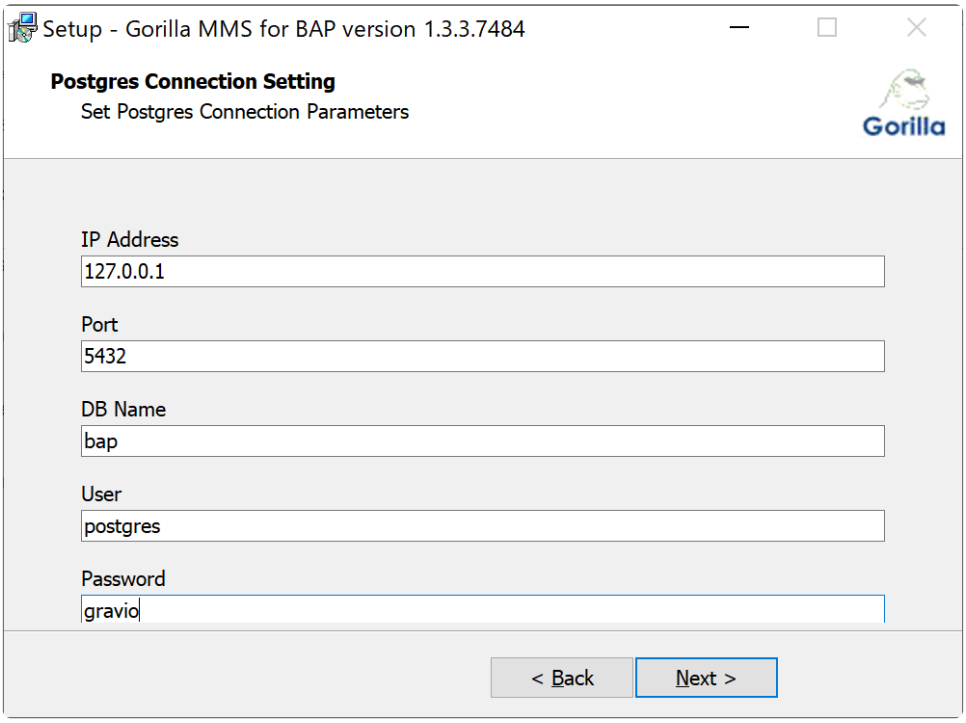
Press the Install button.



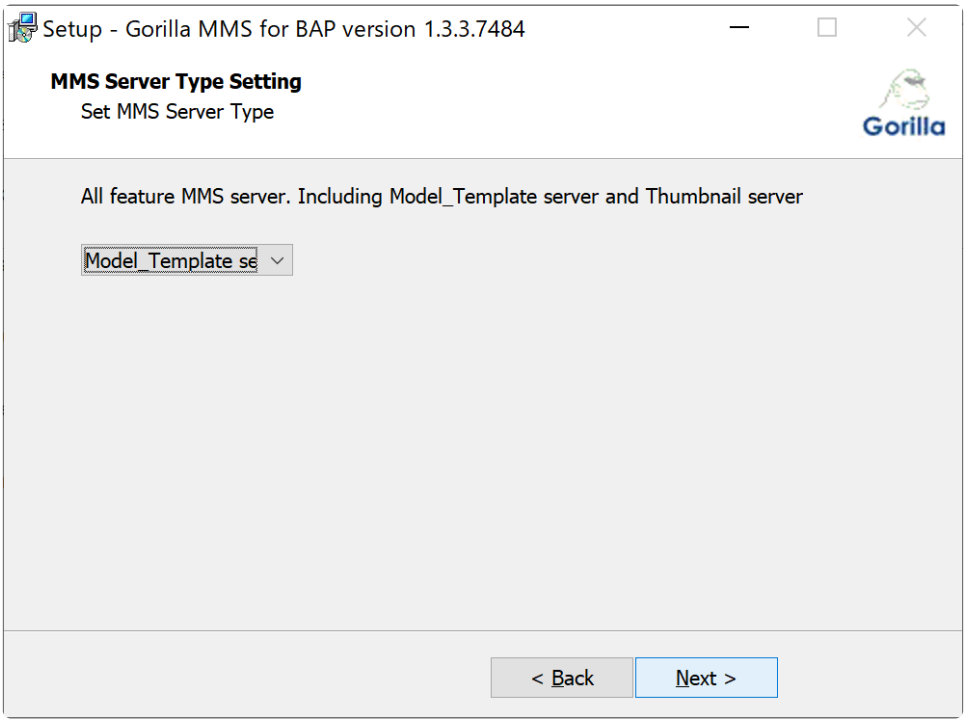
The file copy will start.



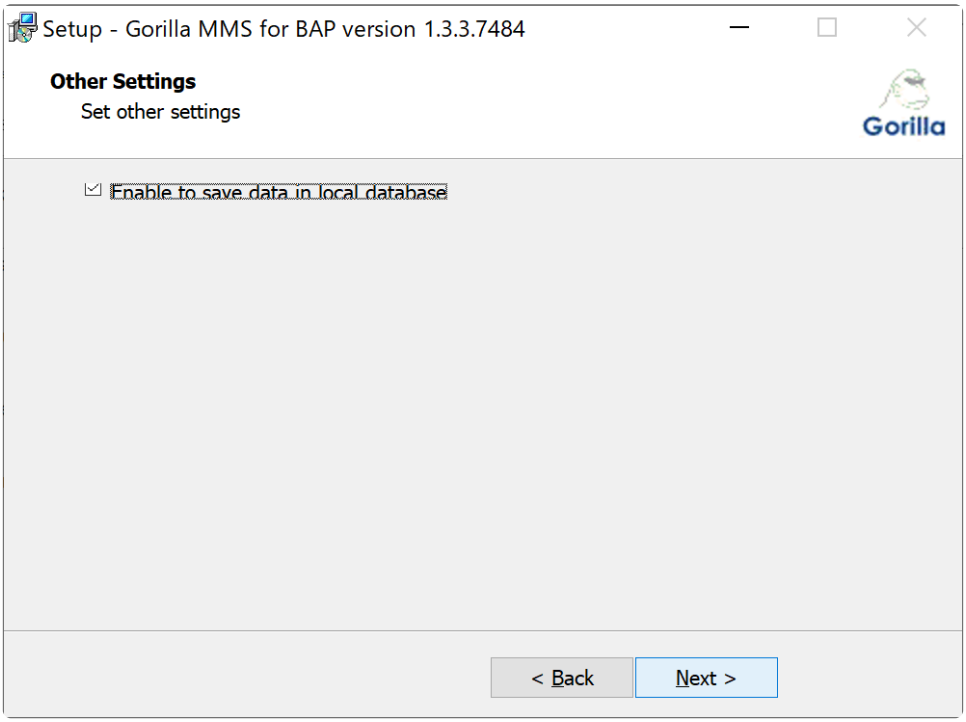
Just press the Next button.



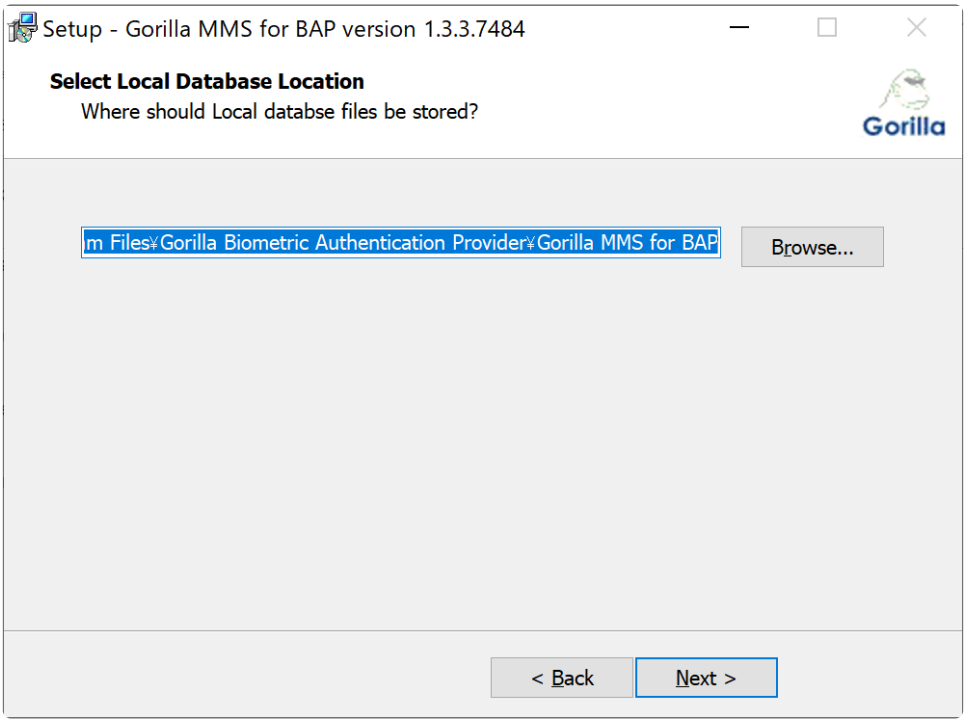
Make sure that the value set in the PostgreSQL installation is displayed, and click the Next button.



Just press the Next button.

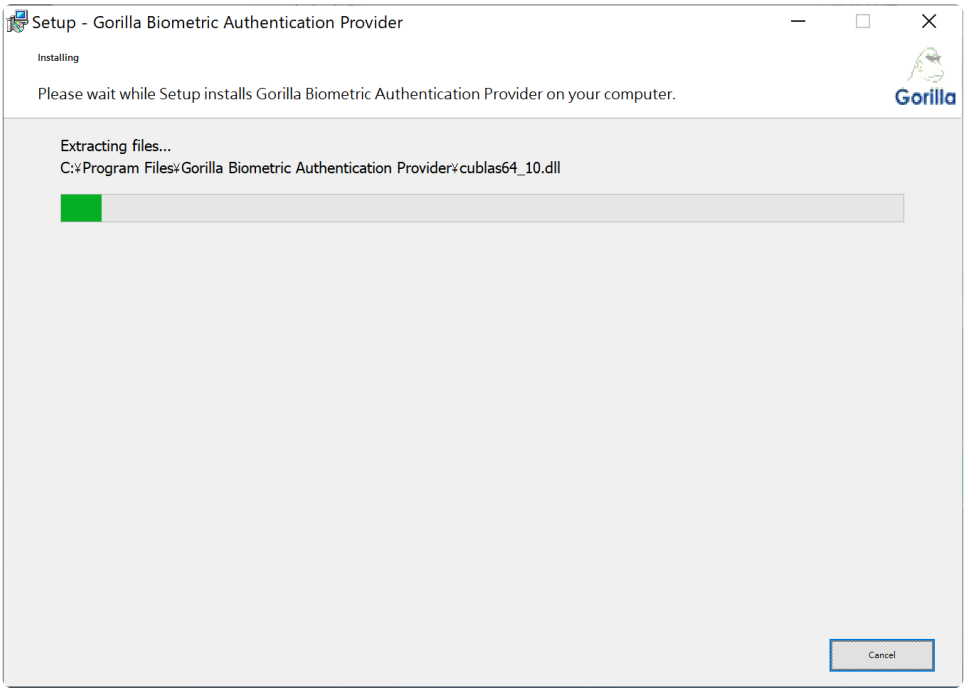


Just press the Next button.

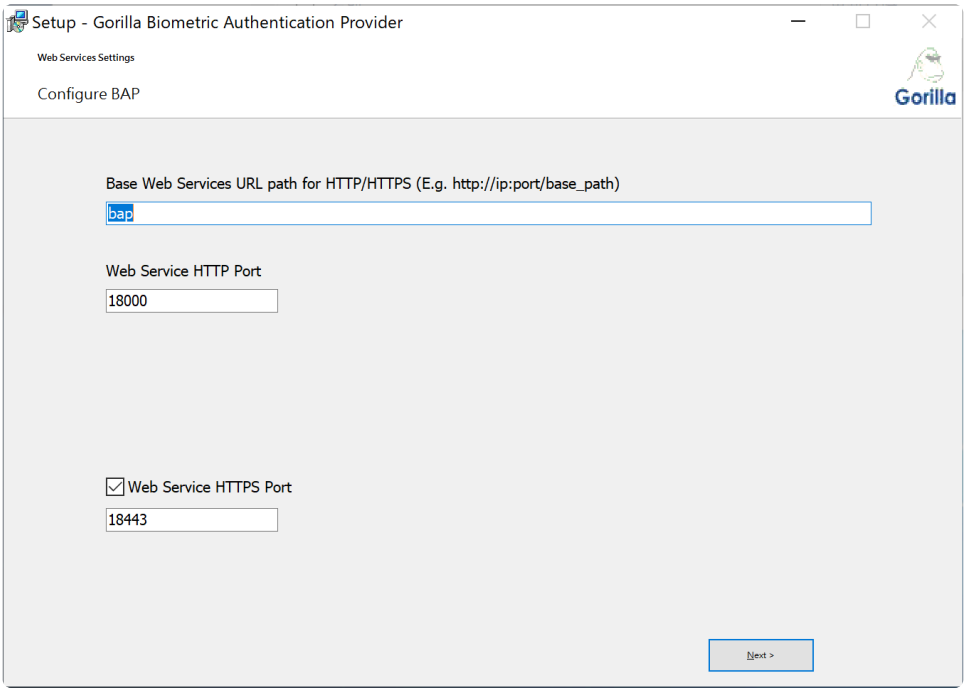


Please click the Next button to install in the standard folder.

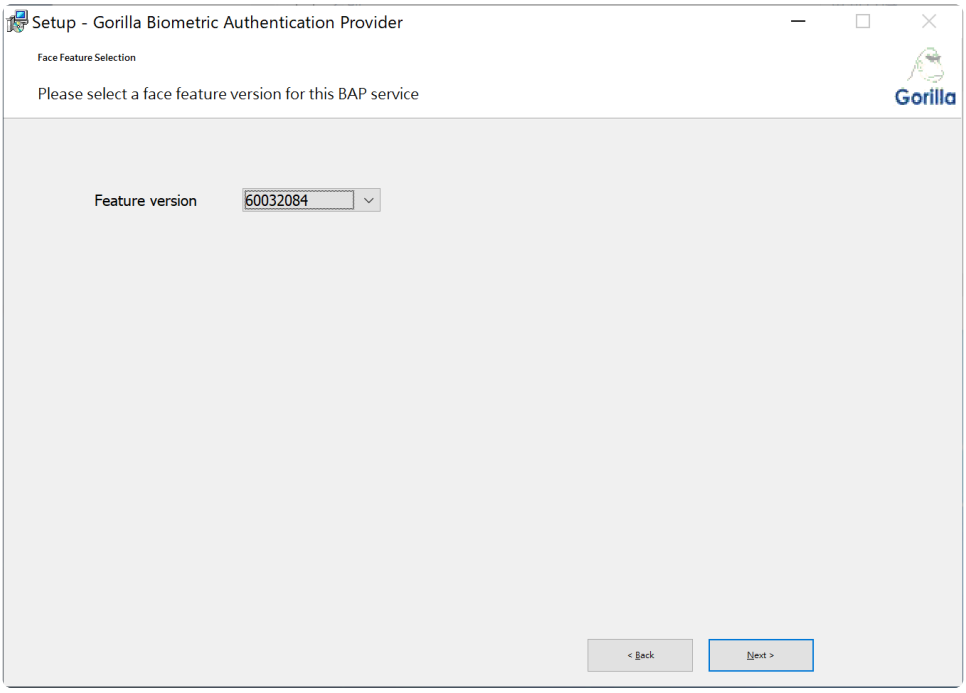
When the MMS installation is complete, click the Finish button.



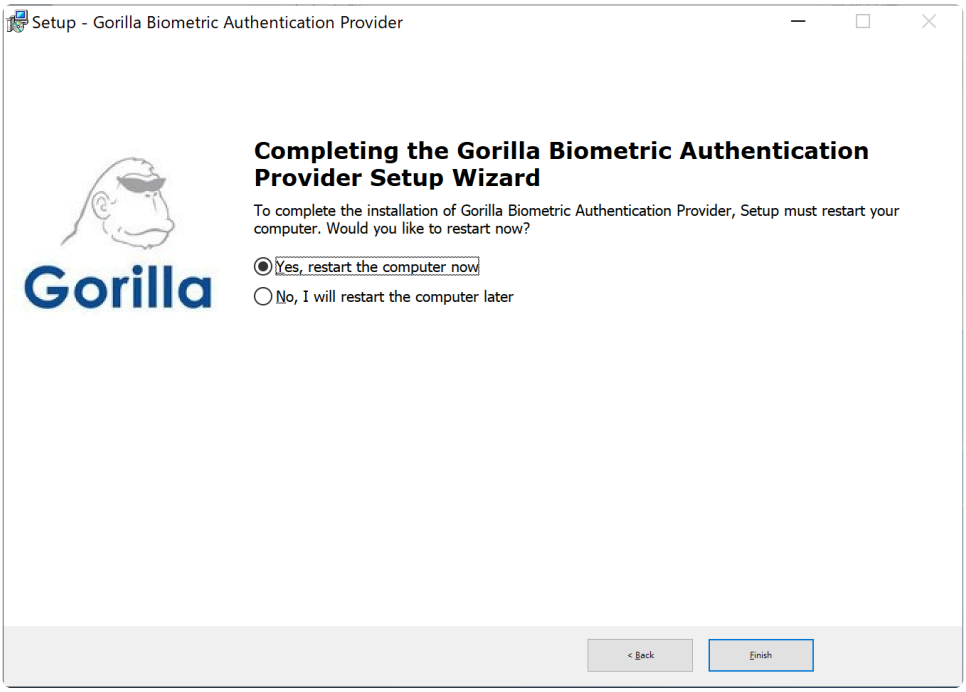
Then the file copy will proceed.



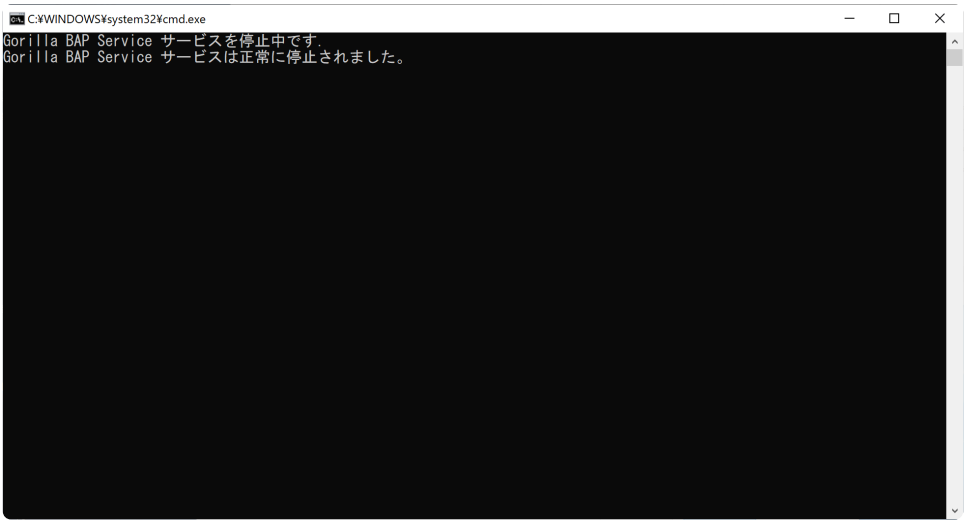
Since it is a standard setting, just press the Next button.



%(color-red) Be sure to select 60032084 as the Feature version, as it cannot be used with the default version in the current BAP version. %



When the installation is complete, restart the PC, by clicking the Finish button.



Installed files will be displayed and terminal windows may appear irregularly. They can safely be ignored.

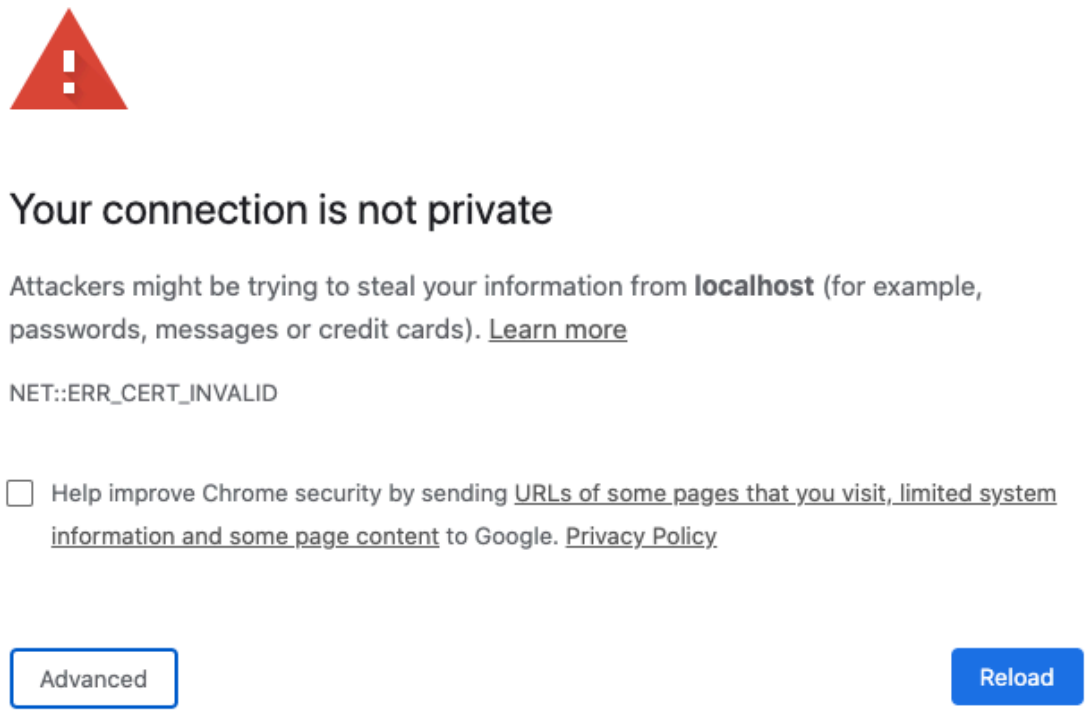
7.2.1. Gorilla BAP License Registration

License registration

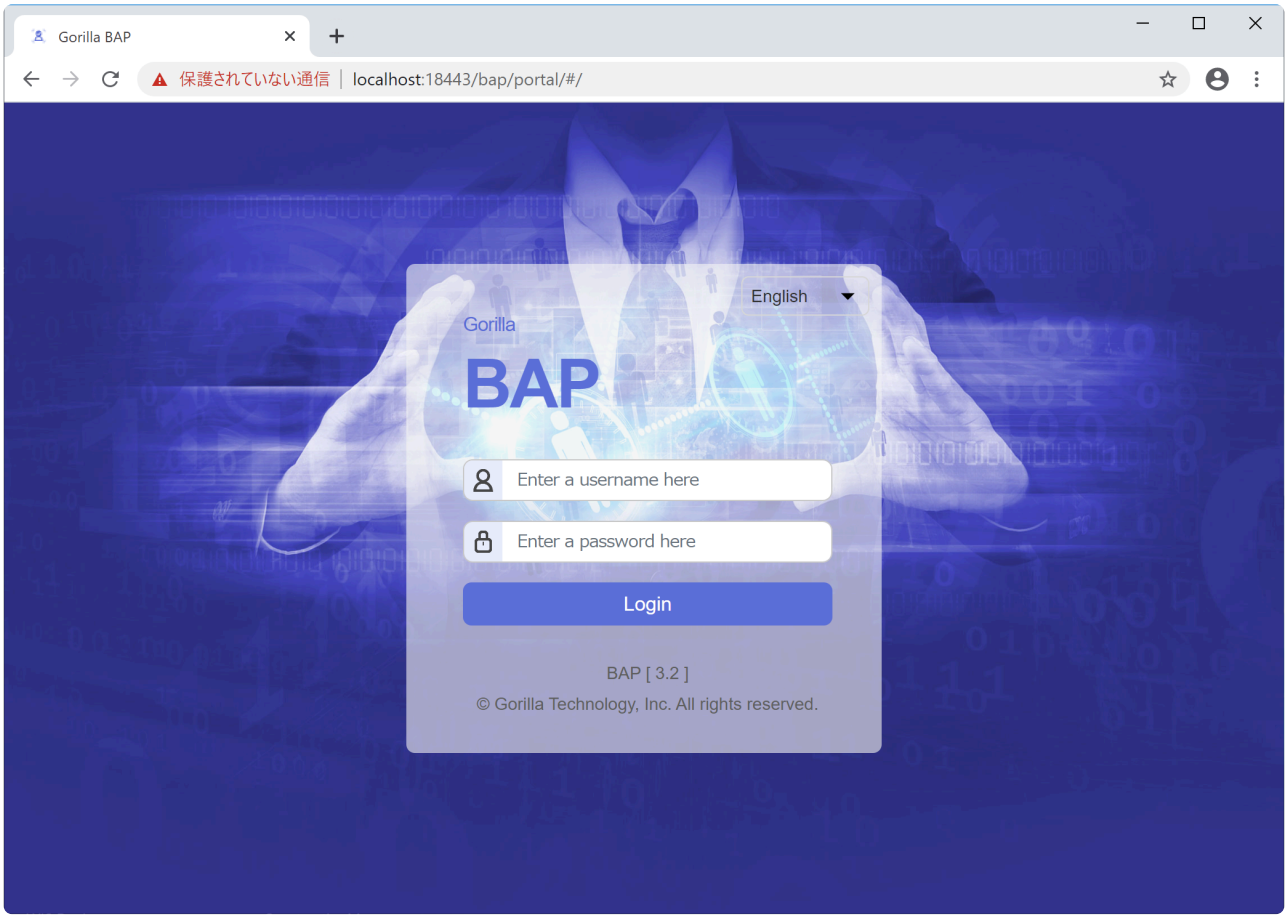
After installing BAP, use the Chrome browser to open the following address:

https://localhost:18443/bap/portal

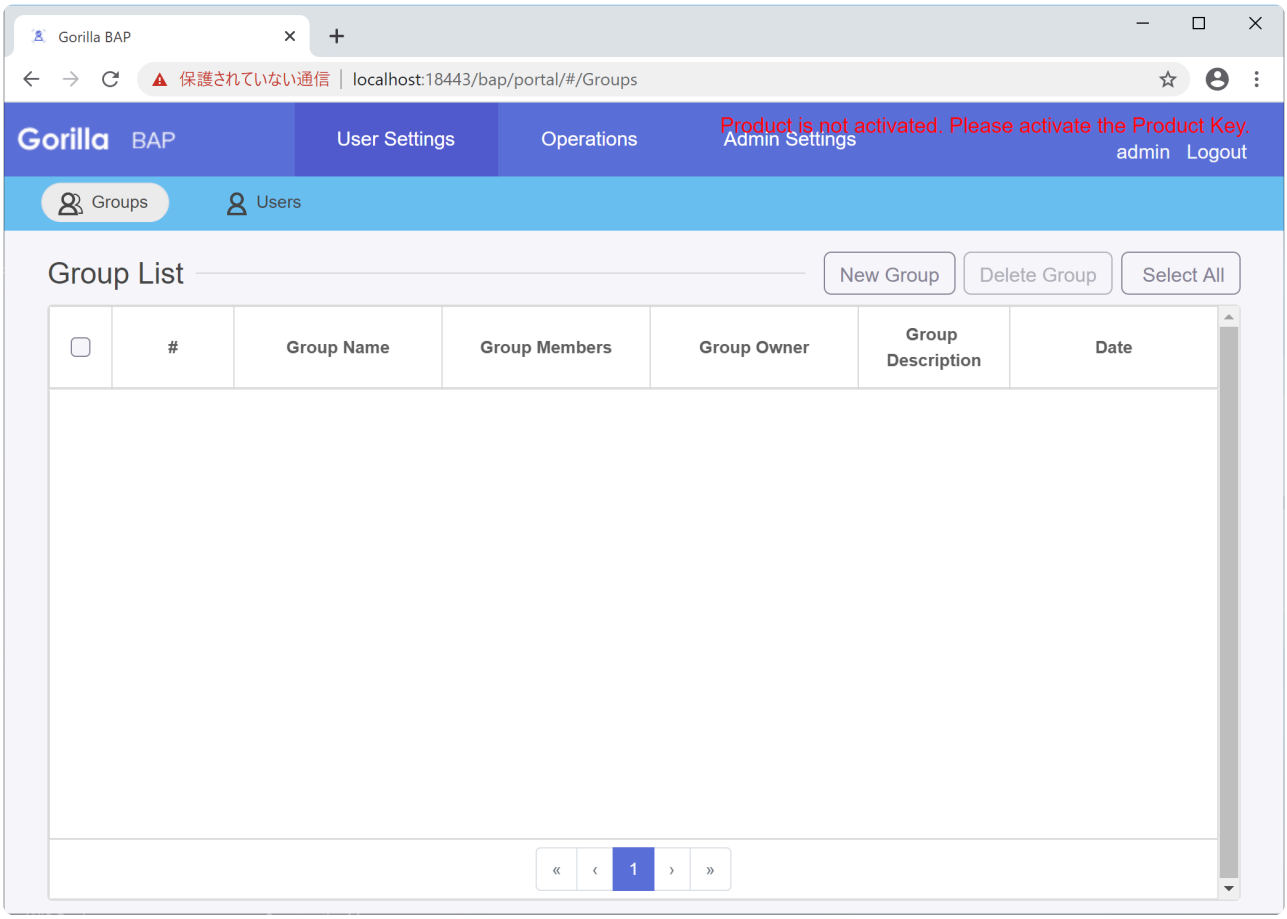
Note: This assumes that you operate Chrome on the same PC as the BAP server.
When operating on a different PC, change “localhost” accordingly



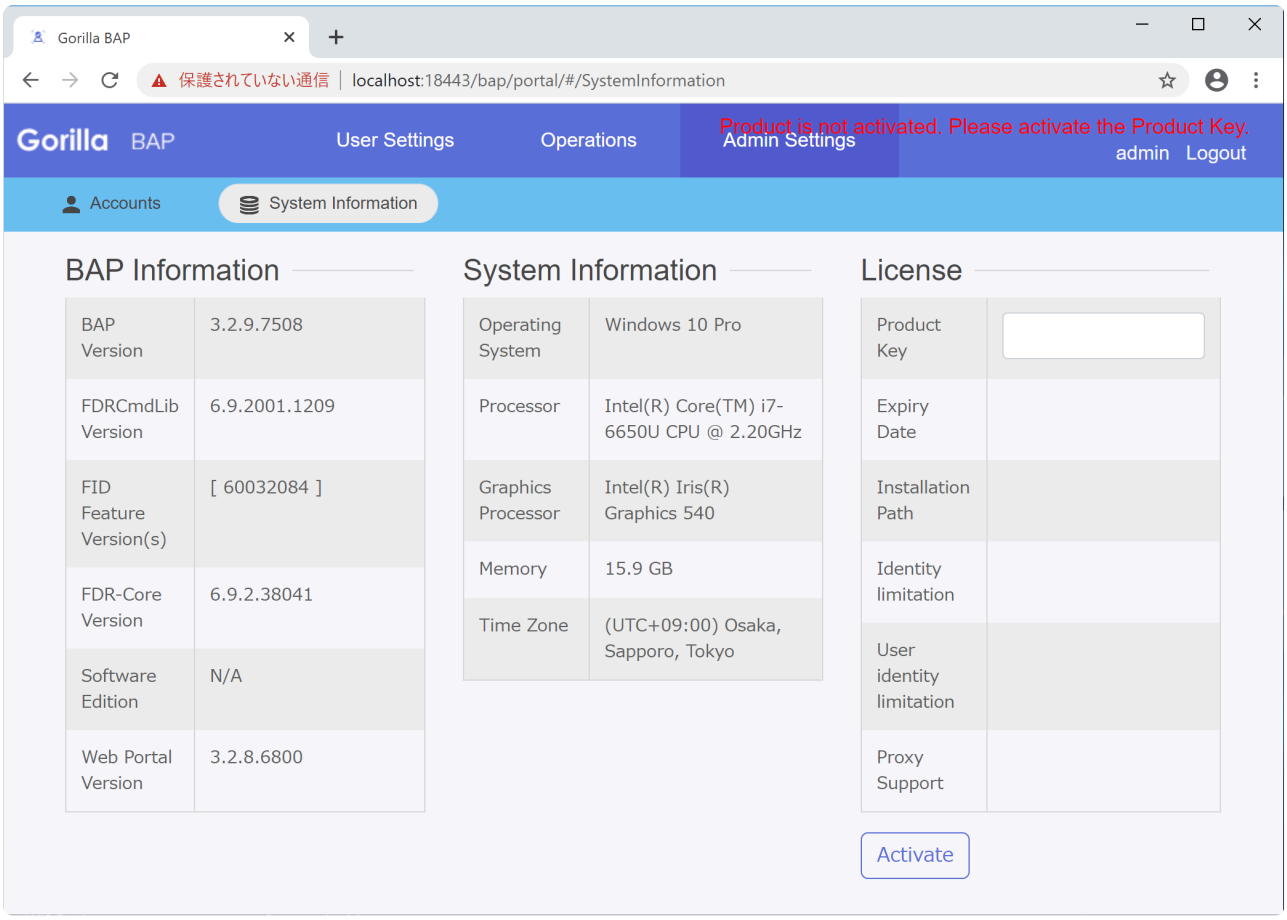
Note: A warning is displayed first because you are using a self-signed certificate with https.
If you see this warning, press the Advanced button to access the localhost (insecure) link.



Please register the admin and password of BAP Administrator and press the Login button.



After logging in, the User Setting screen will be displayed.



Next, register the acquired BAP license key and press the Activate button to activate it. Your PC must be connected to the internet to authenticate.

7.2.2. Unregistering the BAP License

Unregister the license

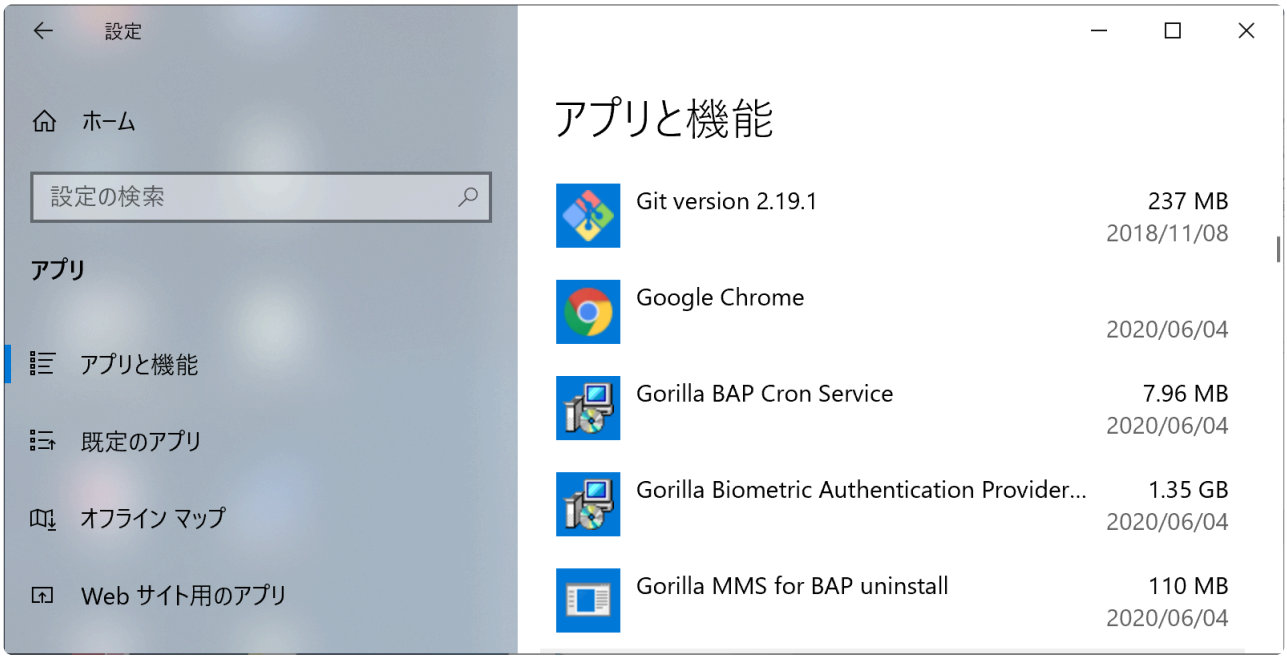
If you want to move your BAP installation to another PC, be sure to deregister the installed license beforehand.

Click the Deactivate button in the Admin Setting -> System Information -> License and confirm that it has been deactivated.

Note: If BAP is uninstalled without deregistering the license beforehand, the license will remain in the registered state. Consequently, this license can not be used on another PC. Your PC must be connected to the internet to unregister.

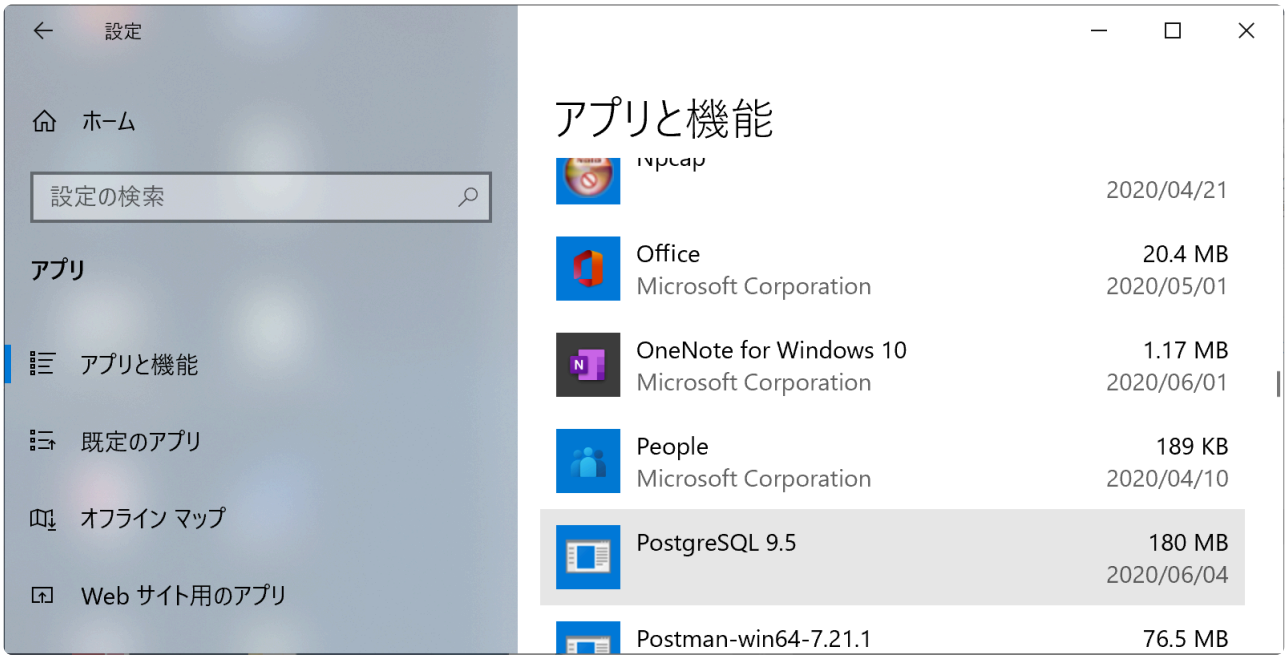
7.2.3. Uninstalling of Gorilla BAP

Open Settings from the Windows Start menu to open apps and features.



Uninstall Gorilla Bap Cron Service

Then uninstall Gorilla Biometric Authentication Provider and Gorilla MMS for BAP uninstall respectively.



Then uninstall PostgreSQL.

Delete the data folder

Don’t forget to delete the data folder used by BAP.

PostgreSQL	c:\Program Files\PostgreSQL
Data	c:\Program Data\Gorilla Biometric Authentication Provider
Gorilla data	c:\Program Files\Gorilla Biometric Authentication Provider, c:\Program Files\Gorilla Technology Group, Inc.

7.3. Installation of Gorilla IVAR

Things to prepare in advance

- Password for the IVAR operational account Administrator
- IVAR license key

Installation environment

Prepare a Windows 10 PC and Chrome browser to install IVAR.
You will also need to be connected to the internet as you will be registering a license.

Notes before installing

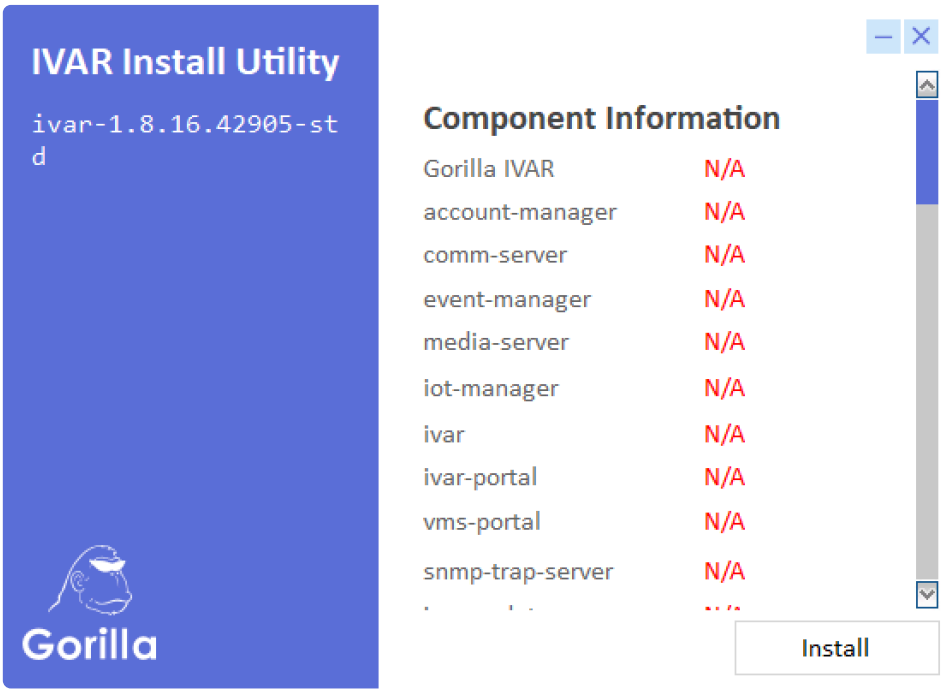
IVAR uses MongoDB. If MongoDB is already installed, IVAR may not be installed correctly due to conflicts such as version differences, so please uninstall MongoDB before proceeding. Also, delete the data folder.

Installation procedure

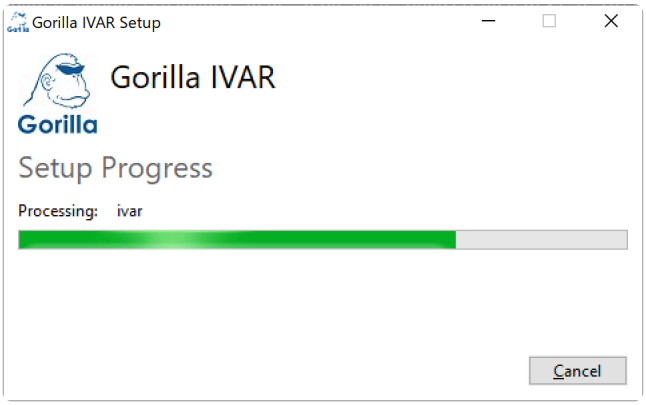
Unzip the downloaded zip. When expanded, an `IVAR` folder will be created, please open its content.

Within the zip file, there is another a zip file @ivar-x.x.x-std-win-x86_64.zip containing the actual server. Please unzip it and open it.

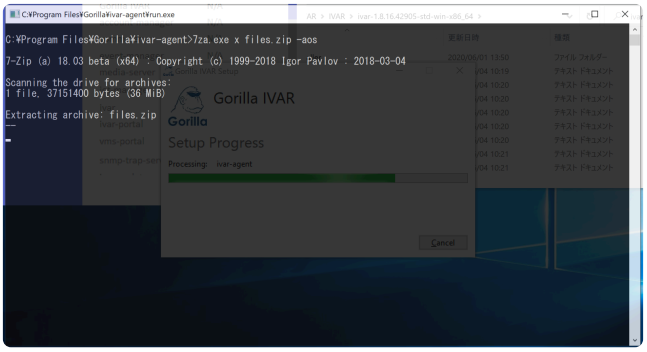
It contains the `IVARInstallUtility` application, please open this application:

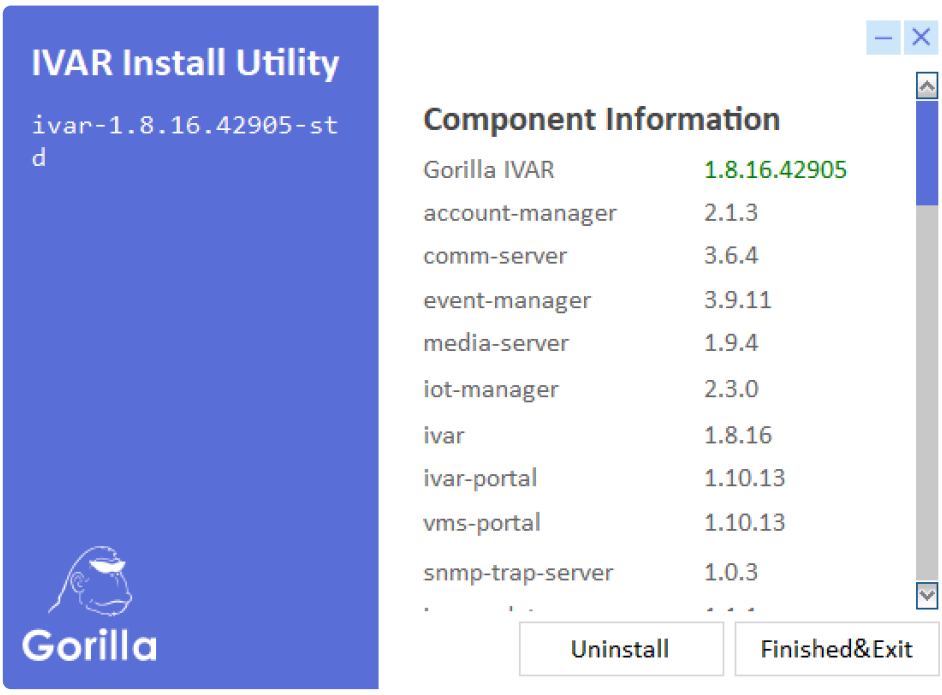


When started up, the module required for IVAR is displayed in Component Information and it is N/A. Click the Install button here to start the installation.



Installed files will be displayed and prompts will appear irregularly. Please wait without operating the screen during installation.





When the installation is complete, the version number of each module will be displayed.

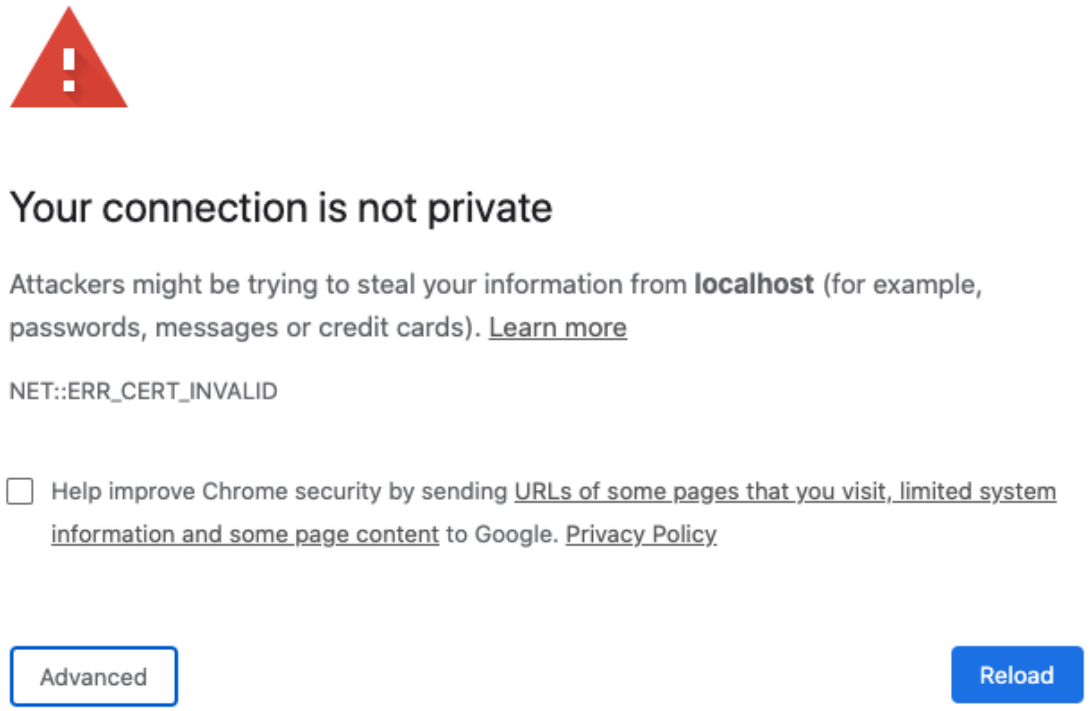
7.3.1. Initial setup and license registration of Gorilla IVAR

Initial settings

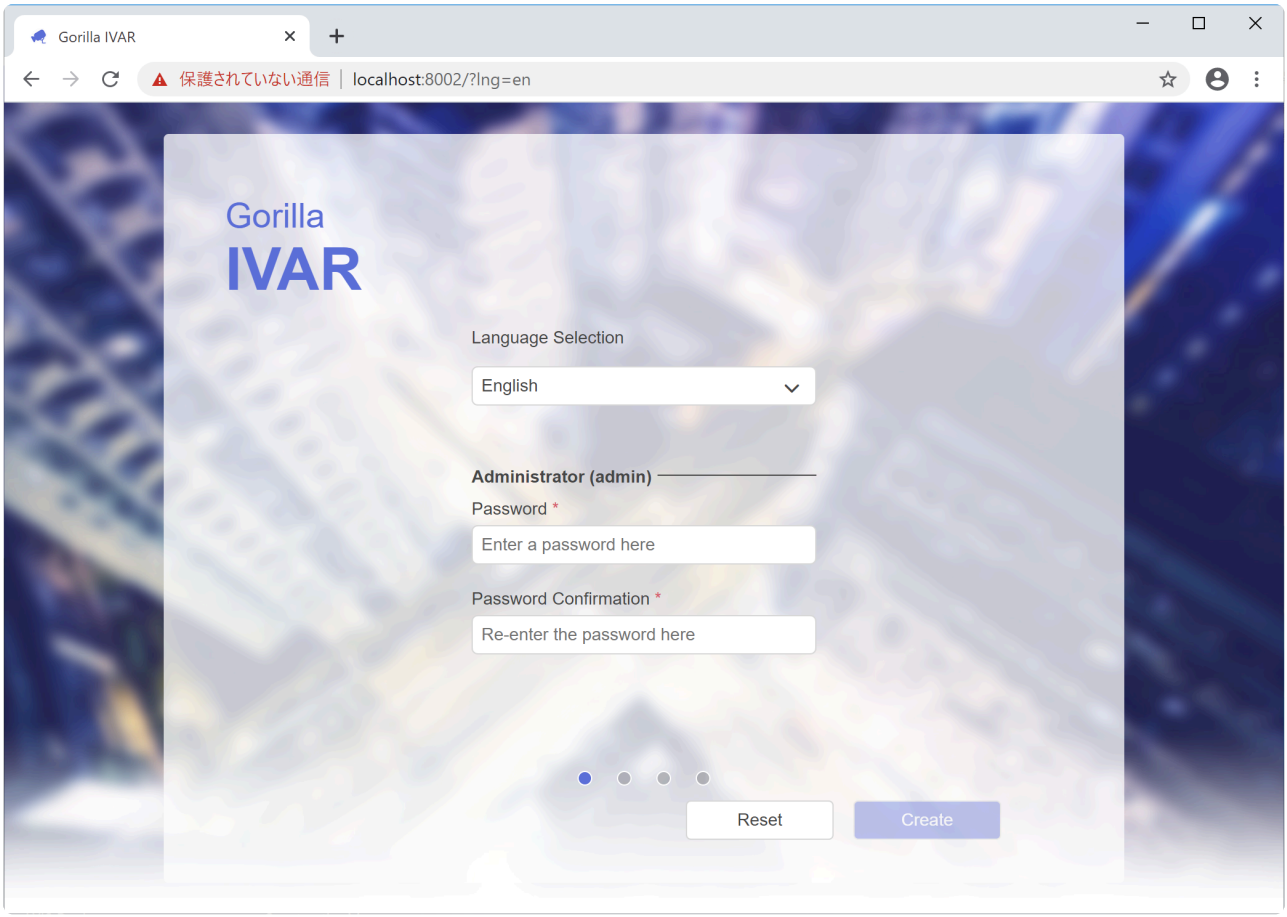
After installing IVAR, use Chrome to open the following address:

https://localhost:8002/

Note: It is assumed that you will operate Chrome on the same PC where IVAR is being installed. When operating on a different PC, change “localhost” according to the server’s IP address.

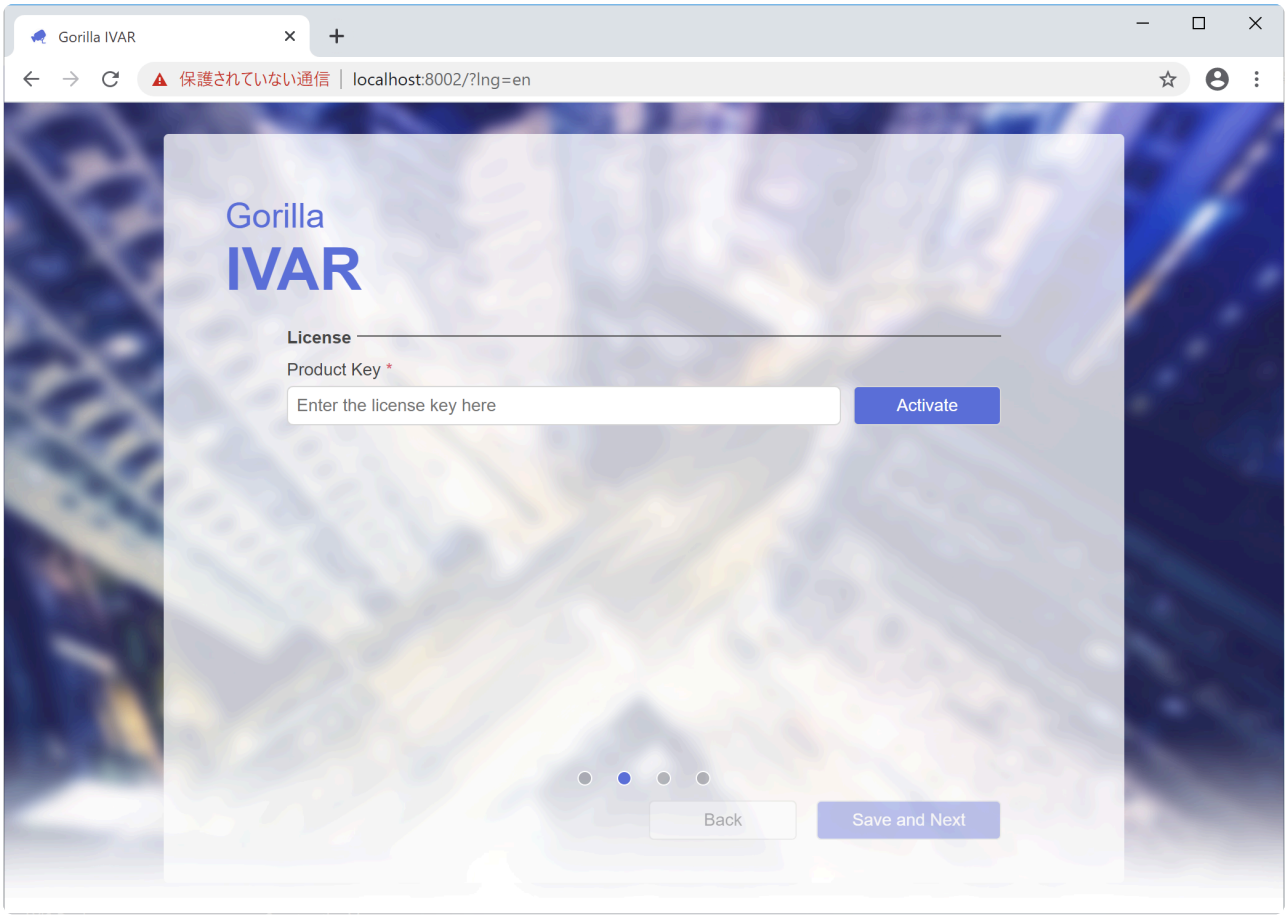


Note: A warning is displayed first because you are using a self-signed certificate with https. If you see this warning, press the Advanced button to access the localhost (insecure) link.



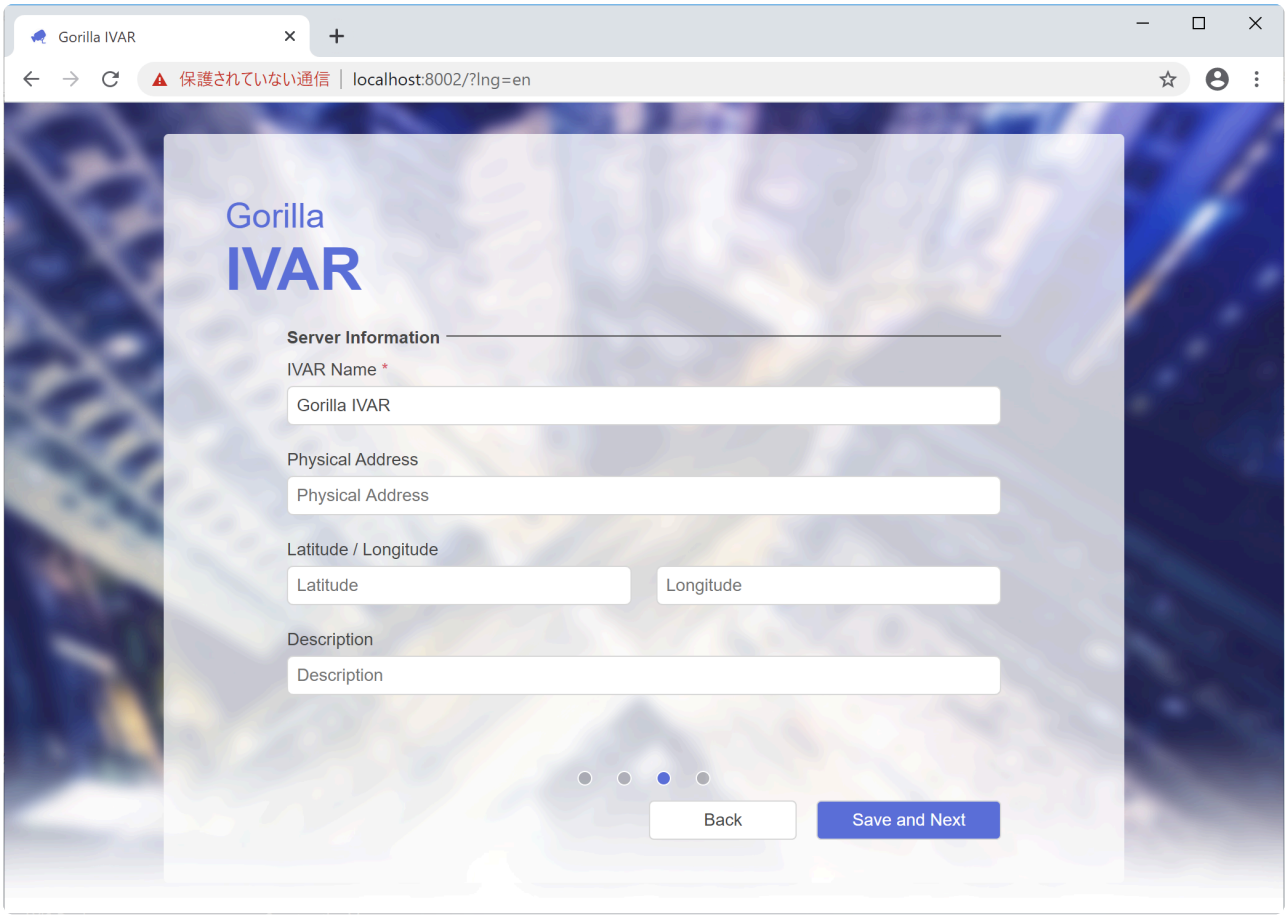
Register the IVAR Administrator password and press the Create button.

License registration



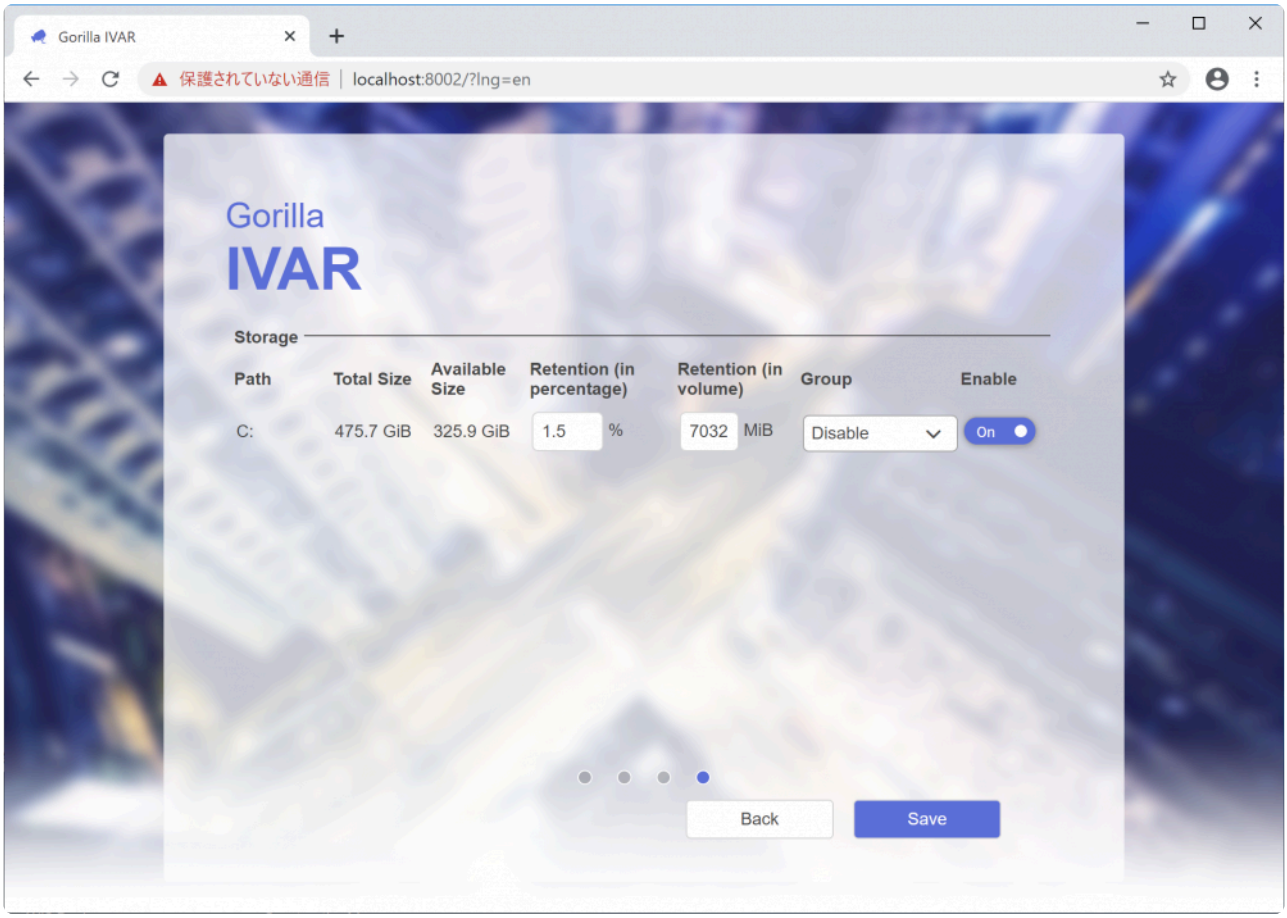
Then enter the obtained IVAR license key and press the Activate button to activate the license. Your PC must be connected to the internet to authenticate.

When authentication is complete, click the Save and Next button.



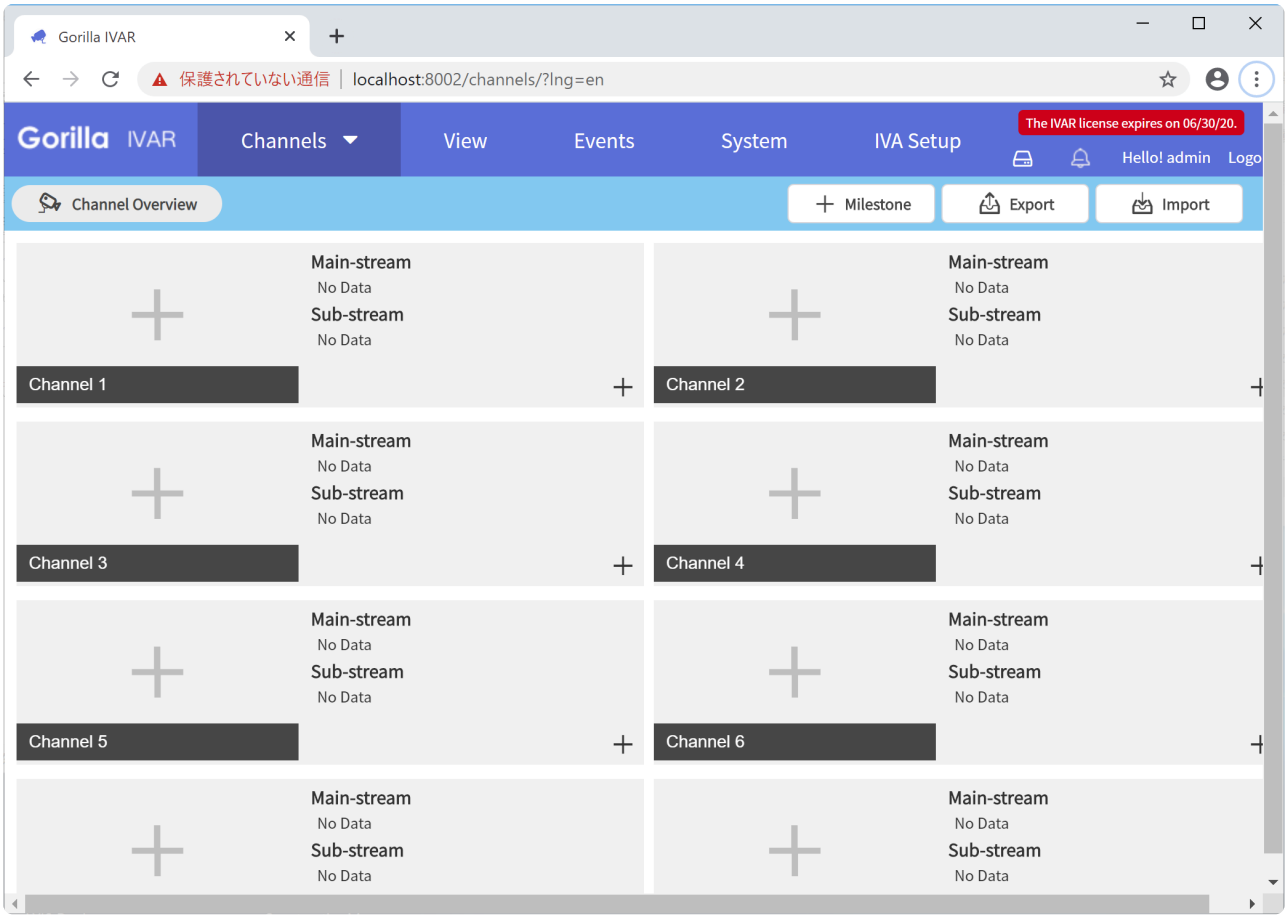
Next, the screen for registering Server Information appears. Adjust the settings to your requirements and press the Save and Next button.

Note: You can also register the information on the settings screen after installation.



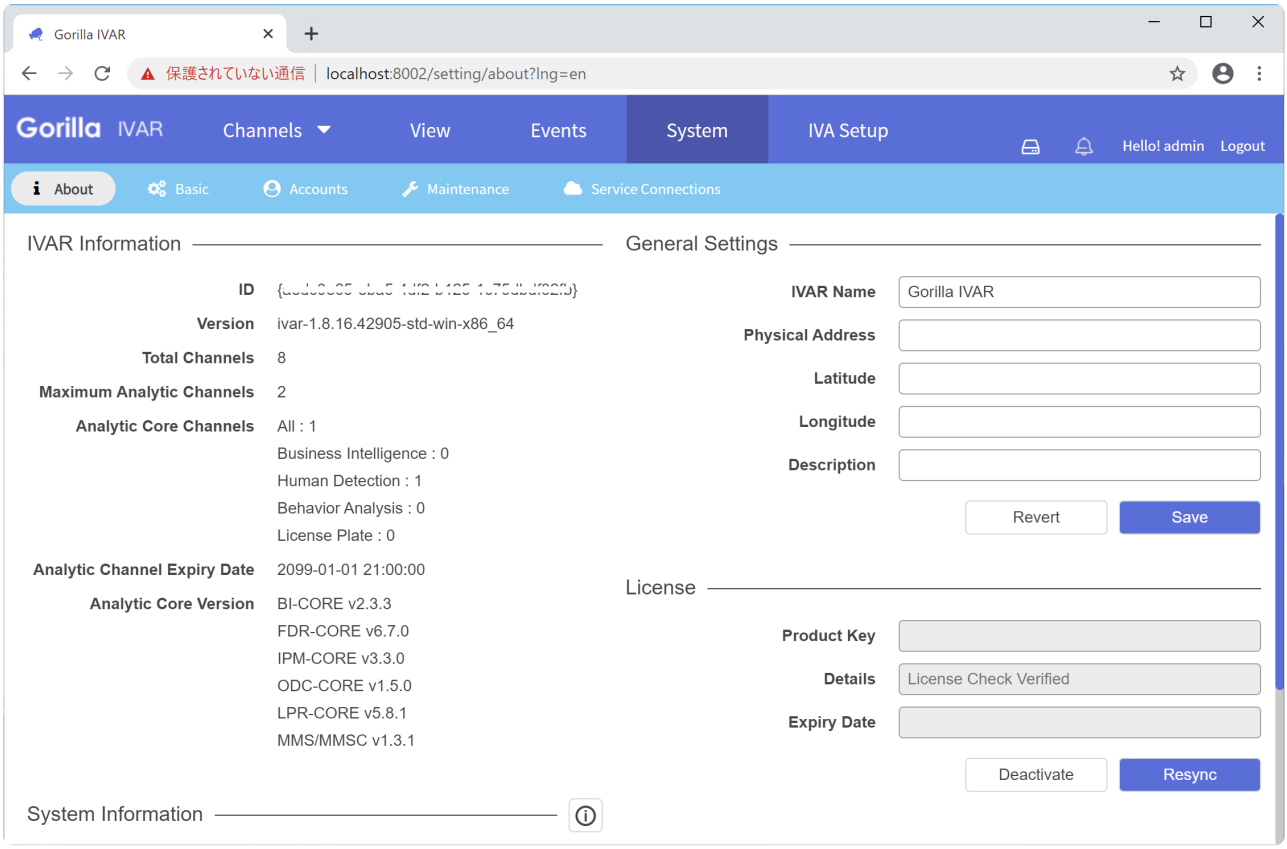
Then the storage registration screen appears. Specify the drive to save the video data. When this setting is completed, press the Save button.

This concludes the initial settings process and opens the channel screen.



License check

To see your registered license, open About from the System menu.



The license registered in License is displayed.

7.3.2. Unregistering the Gorilla IVAR License

Unregistering license

If you want to move your IVAR installation to another PC, be sure to deregister the installed license beforehand.

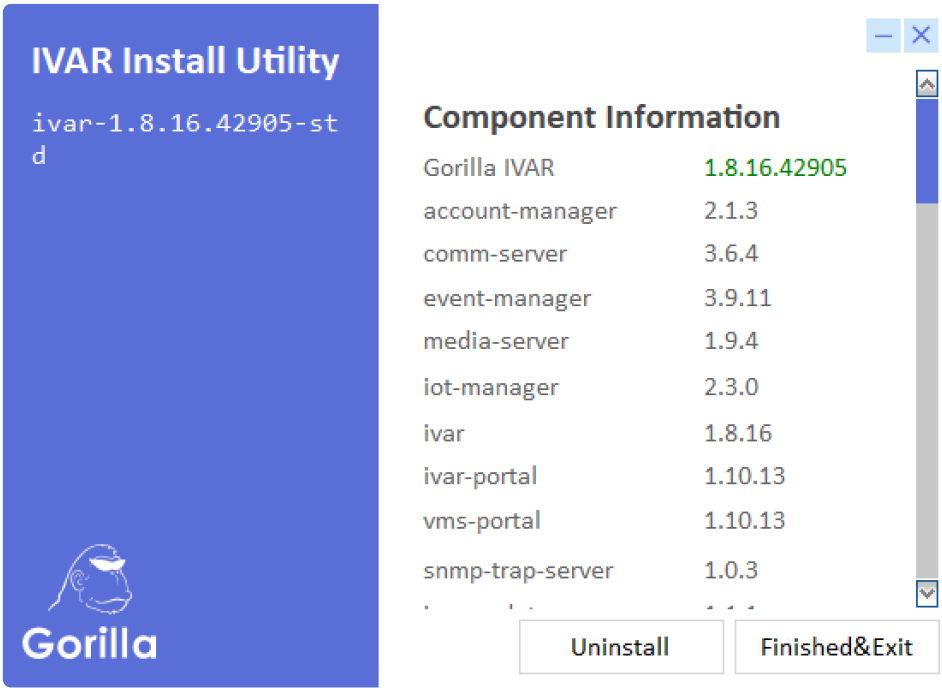
Make sure it is deactivated by pressing the Deactivate button in System -> About -> License.

Note: If IVAR is uninstalled without deregistering the license beforehand, the license will remain in the registered state. Consequently, this license can not be used on another PC. Your PC must be connected to the internet to unregister.

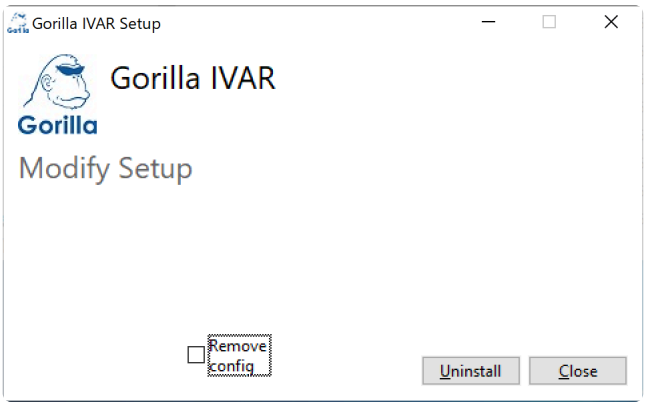
7.3.3. Uninstalling of Gorilla IVAR

Uninstall

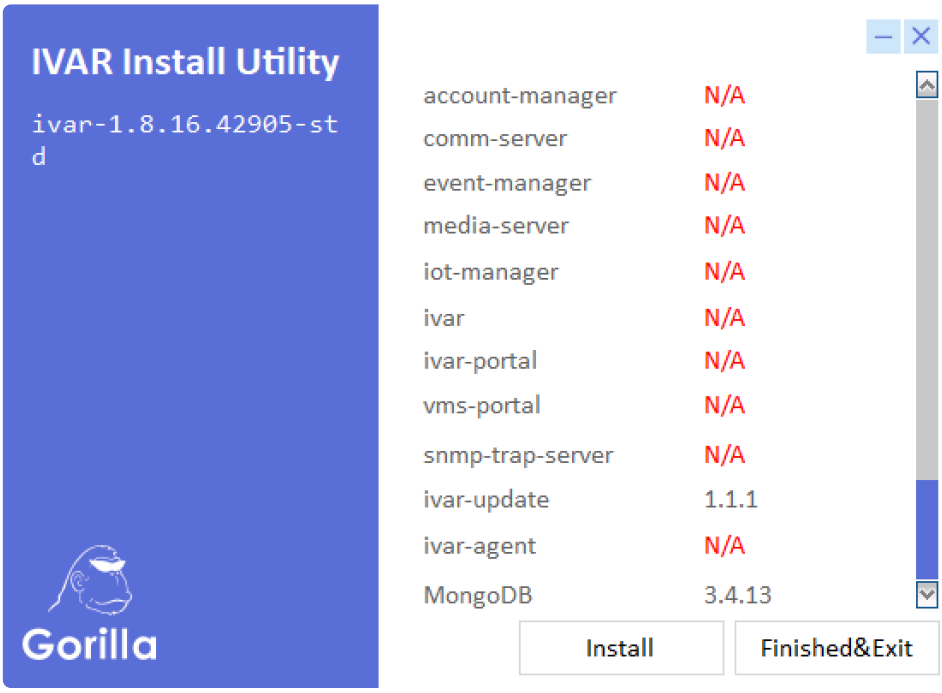
Unzip the zip file that you used to install IVAR. It contains the IVARInstallUtility application.



When started, the Component Information shows the modules installed by IVAR. Press the Uninstall button to remove them.

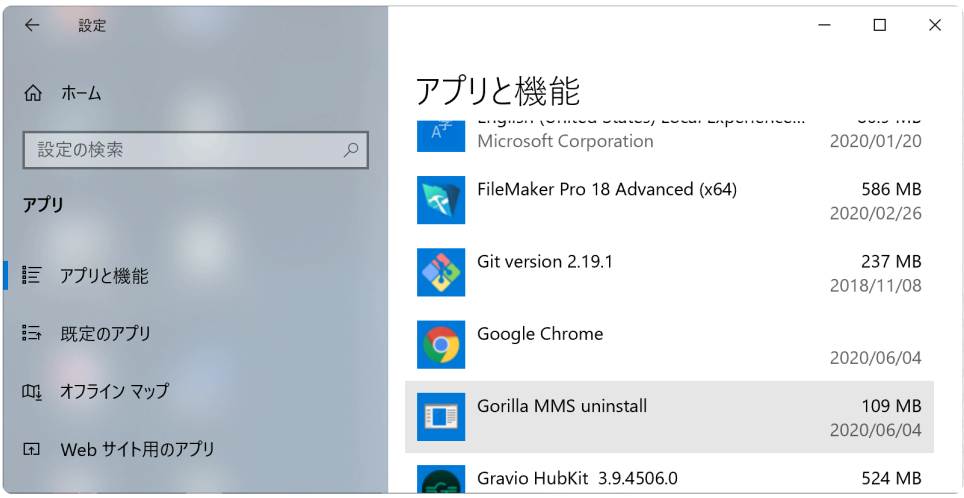


Then press the Uninstall button to start uninstalling.



As soon as the modules are removed, they show up as N/A.
Note: The ivar-update and MongoDB are not uninstalled automatically with this process. You will need to uninstall the Gorilla MMS separately via the Windows Settings.

Open Settings from the Windows Start menu to open apps and features.



Uninstall Gorilla MMS.

This will remove the ivar-update, MongoDB and mongodb_config respectively.

Delete the data folder

Also delete the data writing folder used by IVAR.

MongoDB	c:\data, c:\Program Files\MongoDB

Video data	c:\records
Gorilla data	c:\Program Files\Gorilla MMS

7.4. Testing the Gravio Gorilla Connection

In this chapter, we test the connection between Gravio and Gorilla by counting the number of people in the image recognition process of Gorilla's camera image. The data gathered is then used as a trigger to send the processed data to Gravio.

In Gravio, the an IVAR is set to start an arbitrary action.

Preparation

Prepare a PC with Gorilla IVAR/BAP installed, one ONVIF camera, a PC with Gravio GS/HubKit installed and a Chrome browser. Gravio requires an Enterprise license and the **HubKit requires a registration in advance with a Coordinator.** In this example we use the ONVIF camera TS-NA220W from IO Data.

Setting flow

IVAR camera + image recognition settings

1. Register the ONVIF camera with IVAR
2. Set the image recognition
3. Confirm image recognition on the Live system

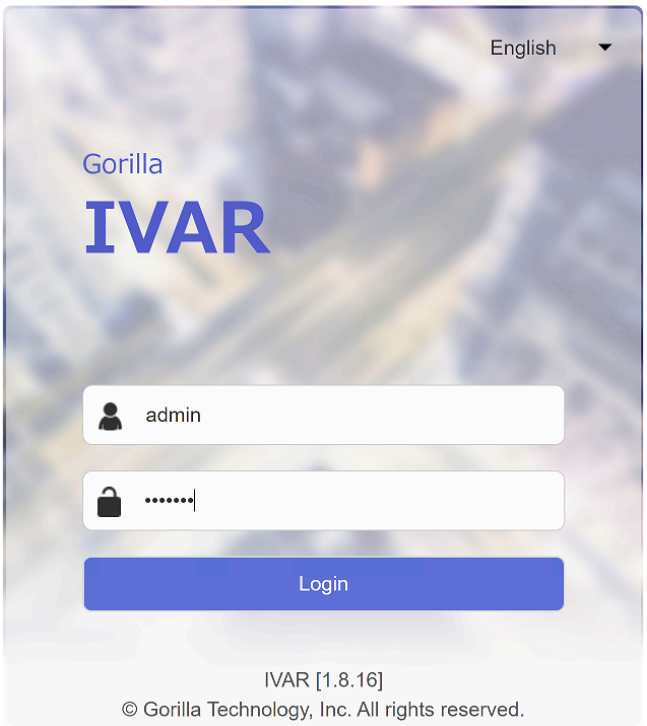
Setting up Gravio and IVAR integration

4. Set up IVAR in Gravio Studio
5. Register the event
6. Confirm event reception in the Data Viewer of Gravio Studio

Gravio IVAR Trigger Settings

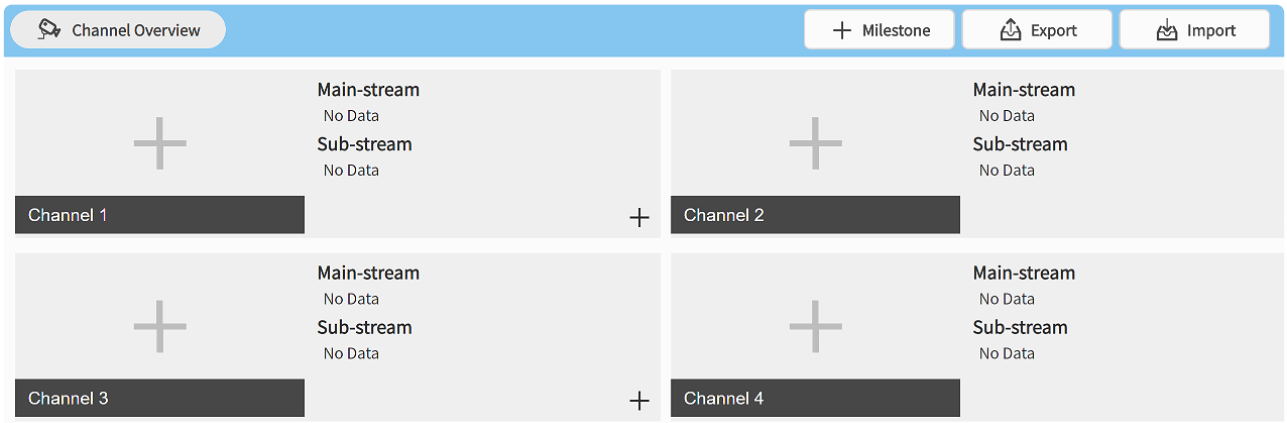
7. Creating an action for invocation
8. Set an IVAR trigger
9. Confirm execution of the action by triggering a noticeable activity

7.4.1. IVAR Camera and Image Recognition Setup

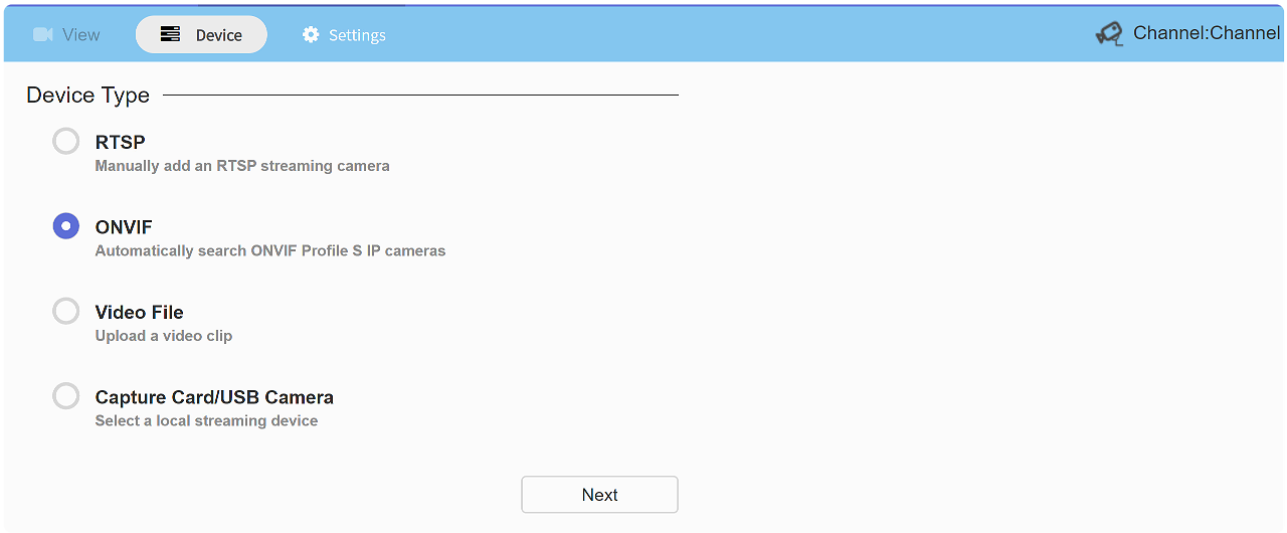


Log in to IVAR using the admin account.

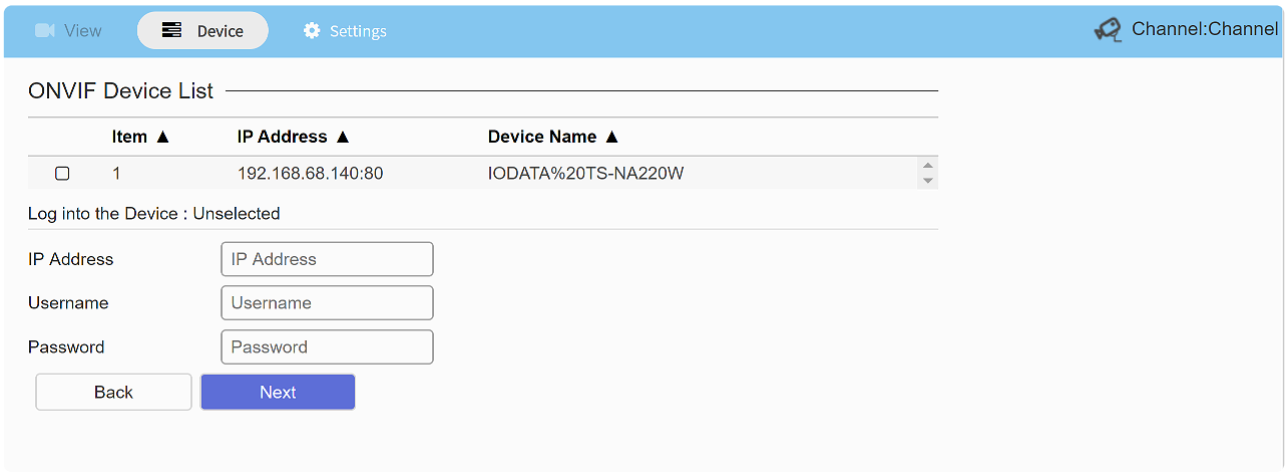
1. Register the ONVIF camera with IVAR



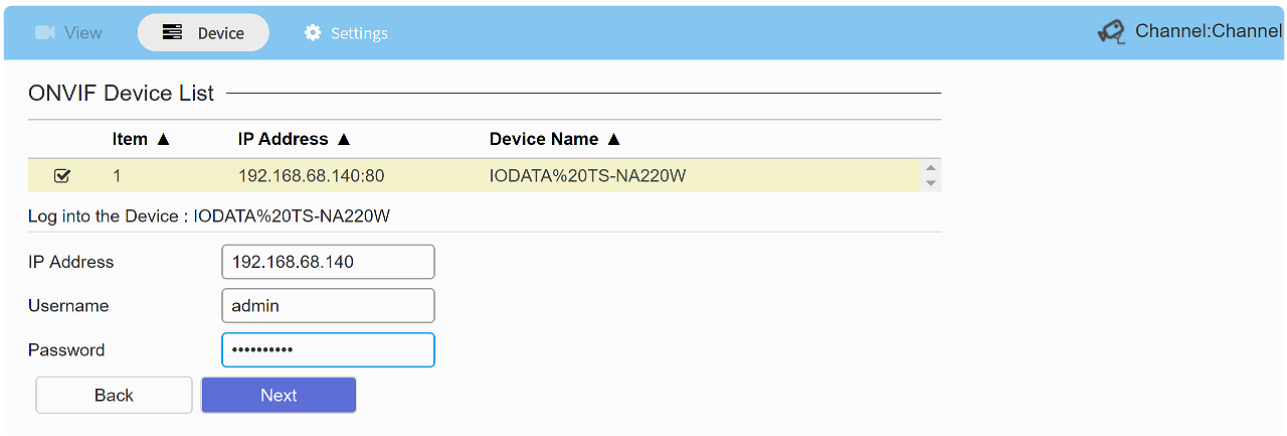
After logging in, the channel screen will be displayed.
Press the + button to register channel 1.



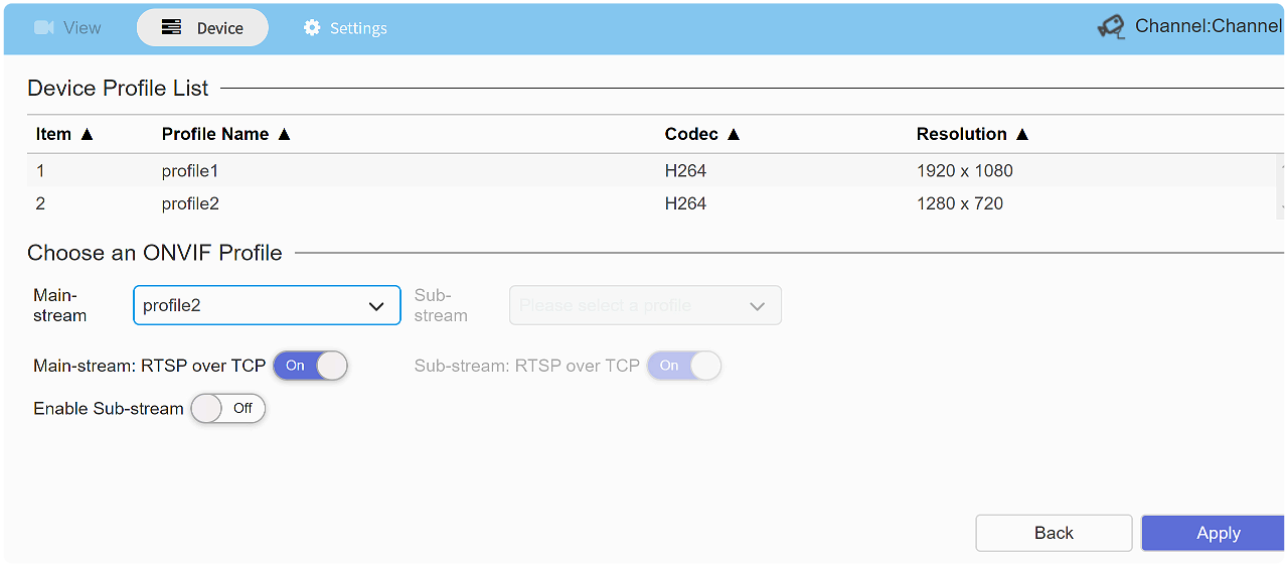
The camera used is an ONVIF camera, so select ONVIF and press the Next button.



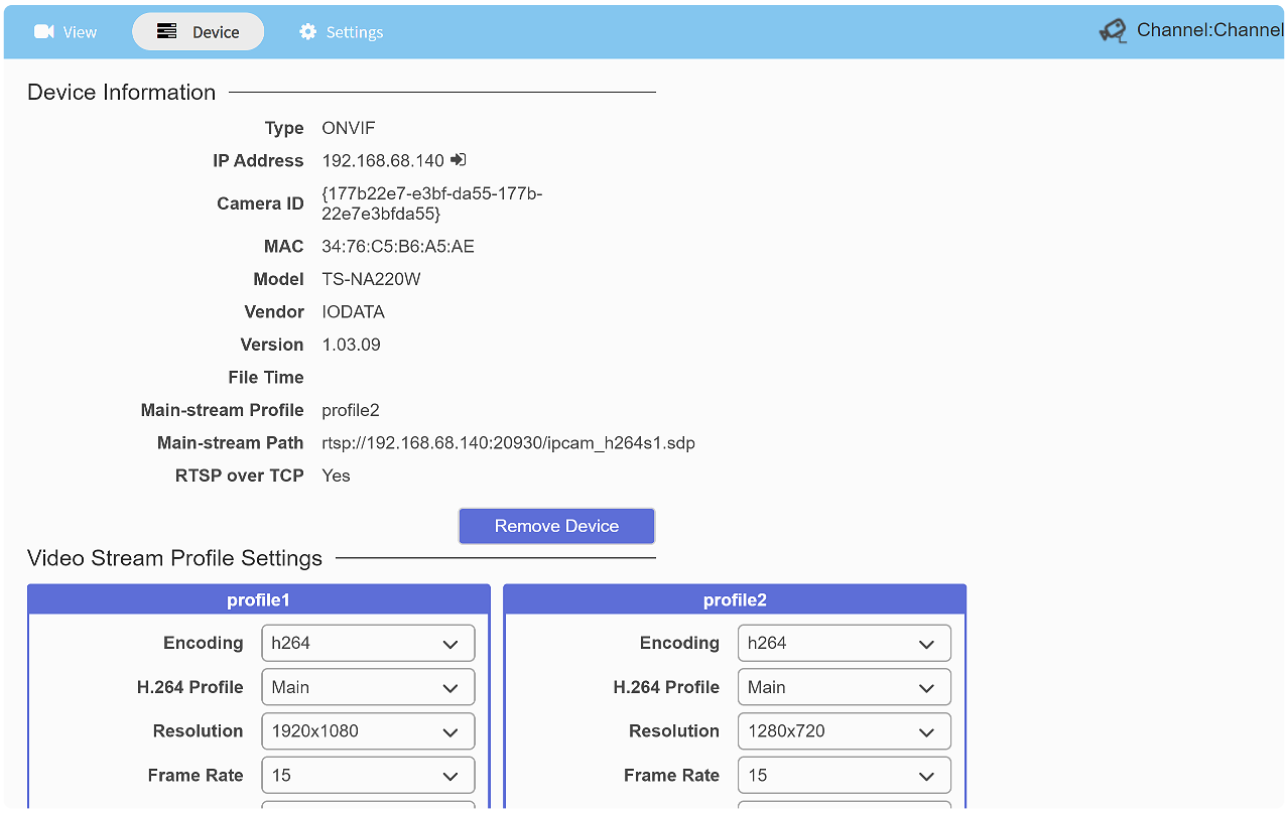
The cameras installed on the same network are displayed in the list.



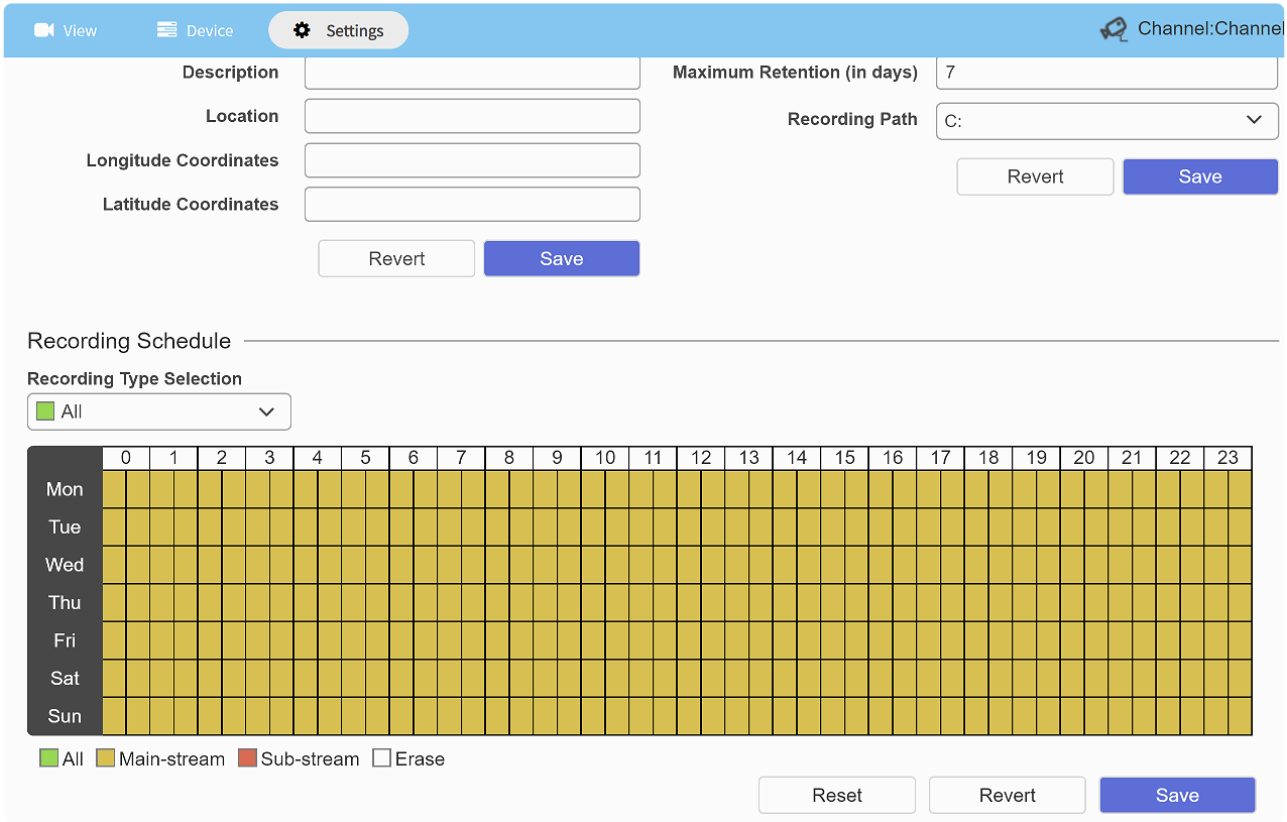
Since we will use IO DATA TS-NA220W this time, select the checkbox, enter the Username and Password and press the Next button.



The profiles that can be used with the camera are displayed in the list, so select Main-stream. Since this is a demo, select profile2 and press the Apply button.

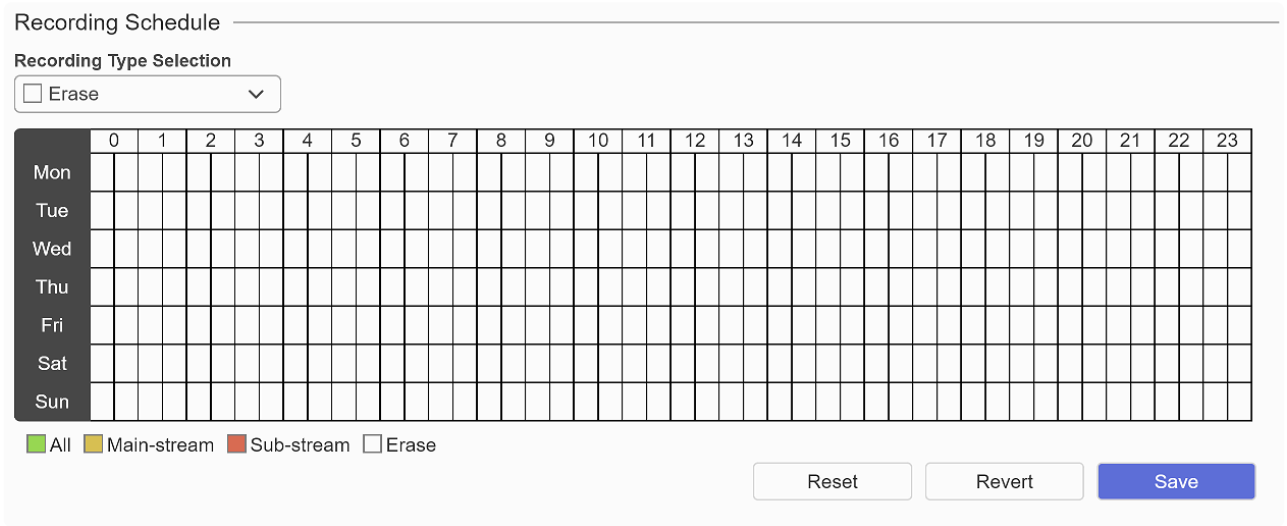


The camera has been registered. To remove it, press the Remove button.

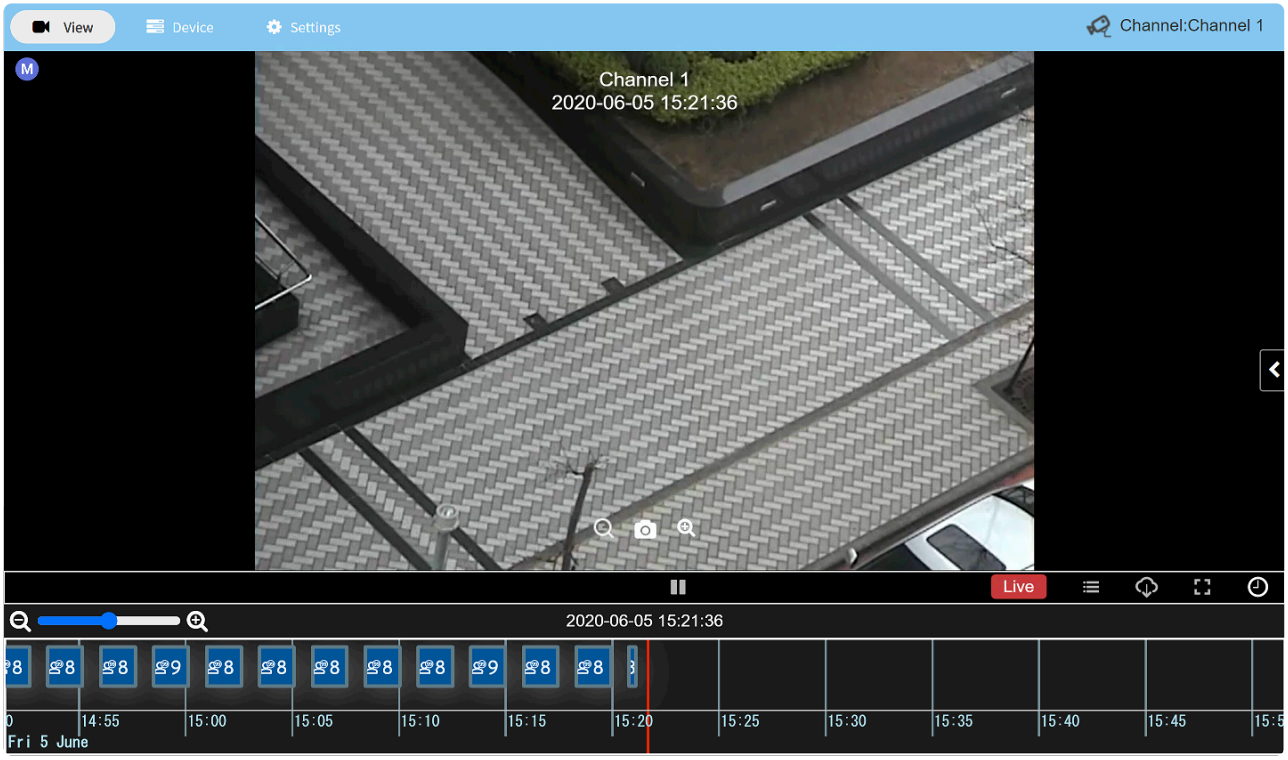


The image is automatically saved when the camera settings are completed, so change the settings so that the image is not saved in the demo so that the hard disk is not compressed.

Press the Setting button to open the Recording Schedule below.

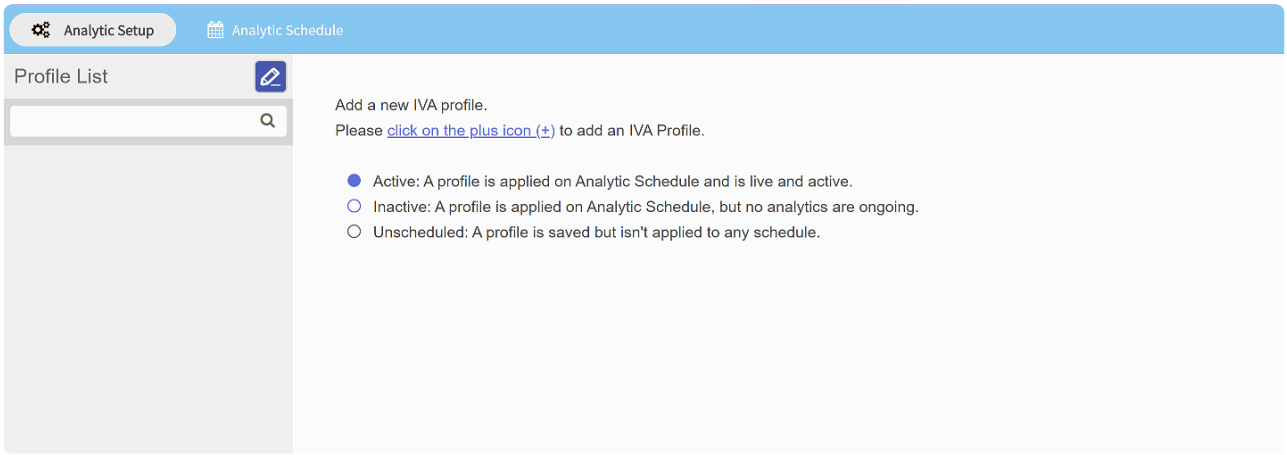


Select Erase in Recording Type Selection, and then clear the recorded Main-stream (yellow) by specifying a range (white).
Press the Save button to change the settings.



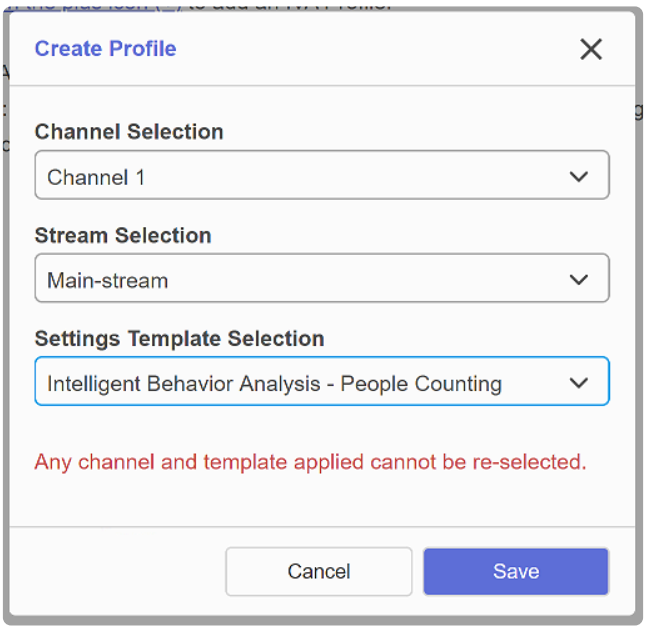
Press the View button to switch to the live image.

2. Image recognition settings

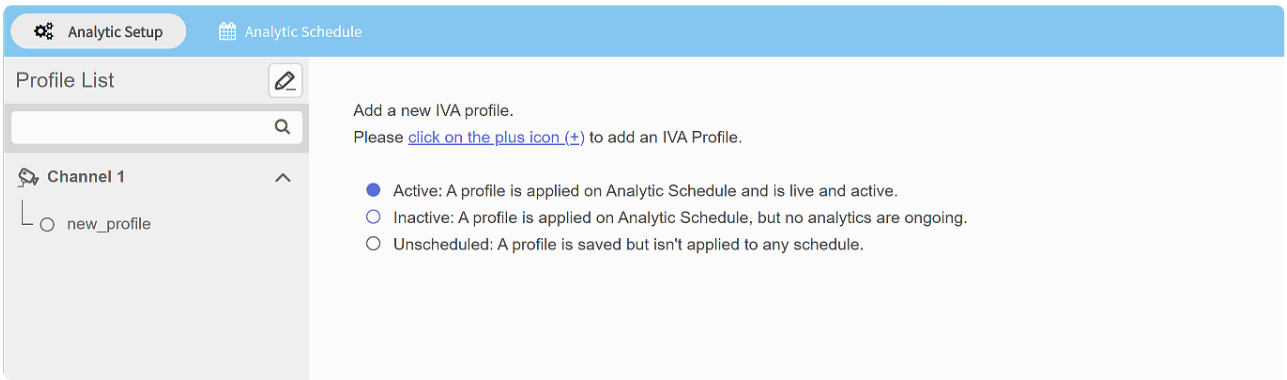


Next, set the image recognition. This time, as a demo, we will set People Count, which counts the number of people captured in the camera.
In addition, there is also face recognition that registers users in advance and a function that recognizes user behaviour transitions.

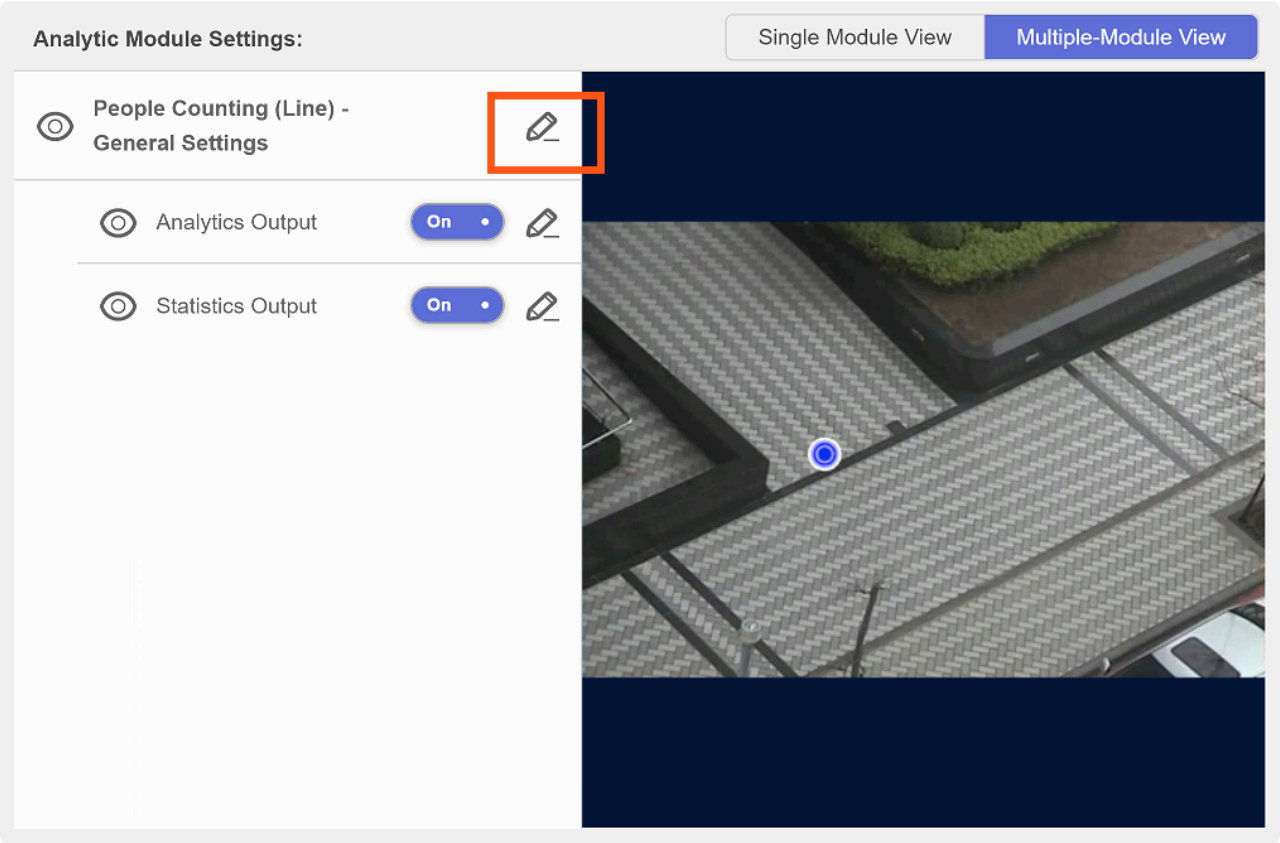
Open Analytic Setup from the IVA Setup menu.
Click the + button on the bottom left or the click on the plus icon (+) link.



In Create Profile, select Channel 1 for Channel Selection, select Intelligent Behavior Analysis-People Count for Settings Template Selection and click the Save button to save.

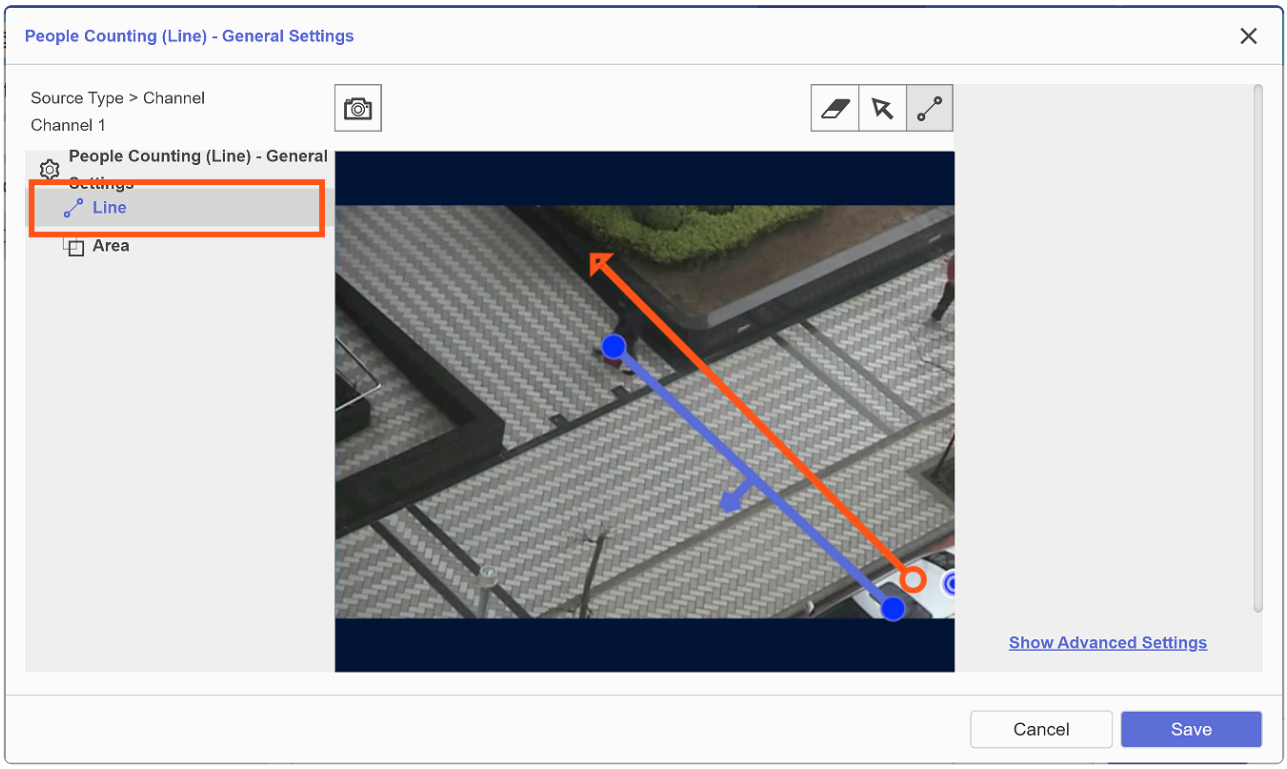


When saved, new_profile will be displayed on Channel 1 on the left, so select it.



Set the conditions under which the People Count is valid. The demo sets a line to recognize and counts the number of people moving across that line.

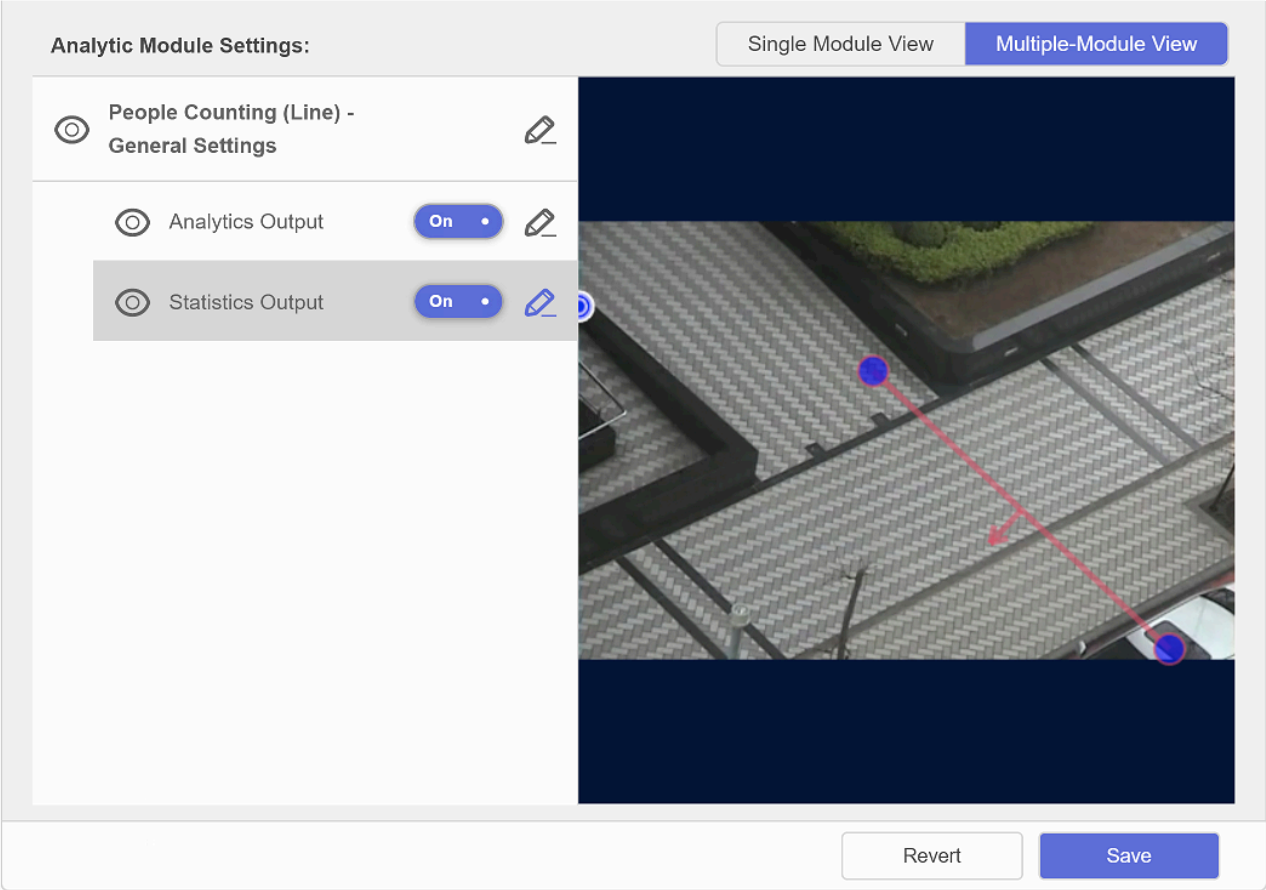
Click the pencil icon.



Select LINE and draw a line with the mouse at the point where the image is recognized. Double-click to confirm.

You can specify whether to cross the line from left to right or from right to left depending on the direction of drawing the line.
In the demo, this is the setting when the line is crossed from the right to the left.

Press the Save button to save.



Confirm the recognized setting conditions and press the Save button to save.

To enable the settings, select Always On for Schedule Settings and press the Save button.

3. Confirm image recognition in Live

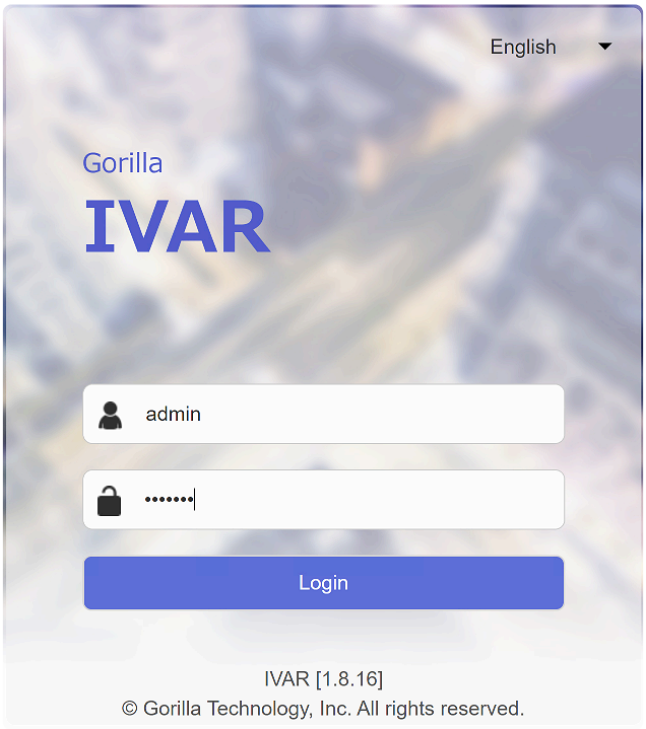
Open Channel 1 from the Channels menu and select View to see the live video.

It is displayed on the right side when the image of the camera is recognized and the event of People

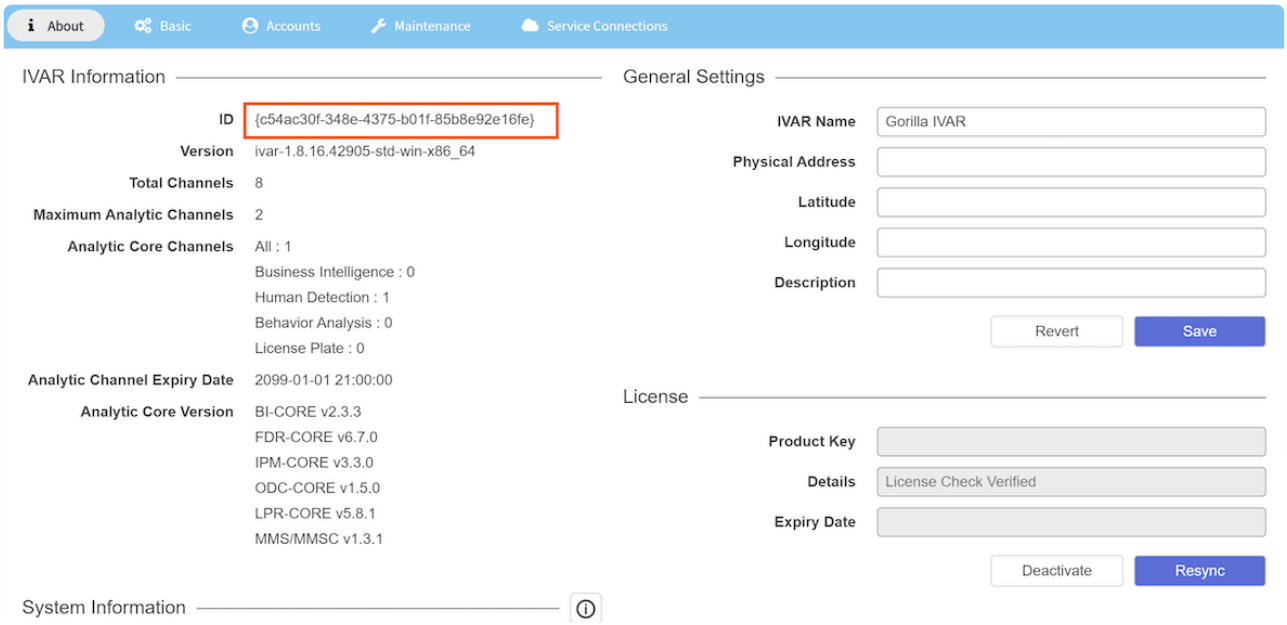
Count occurs.

7.4.2. Connecting Gravio and IVAR

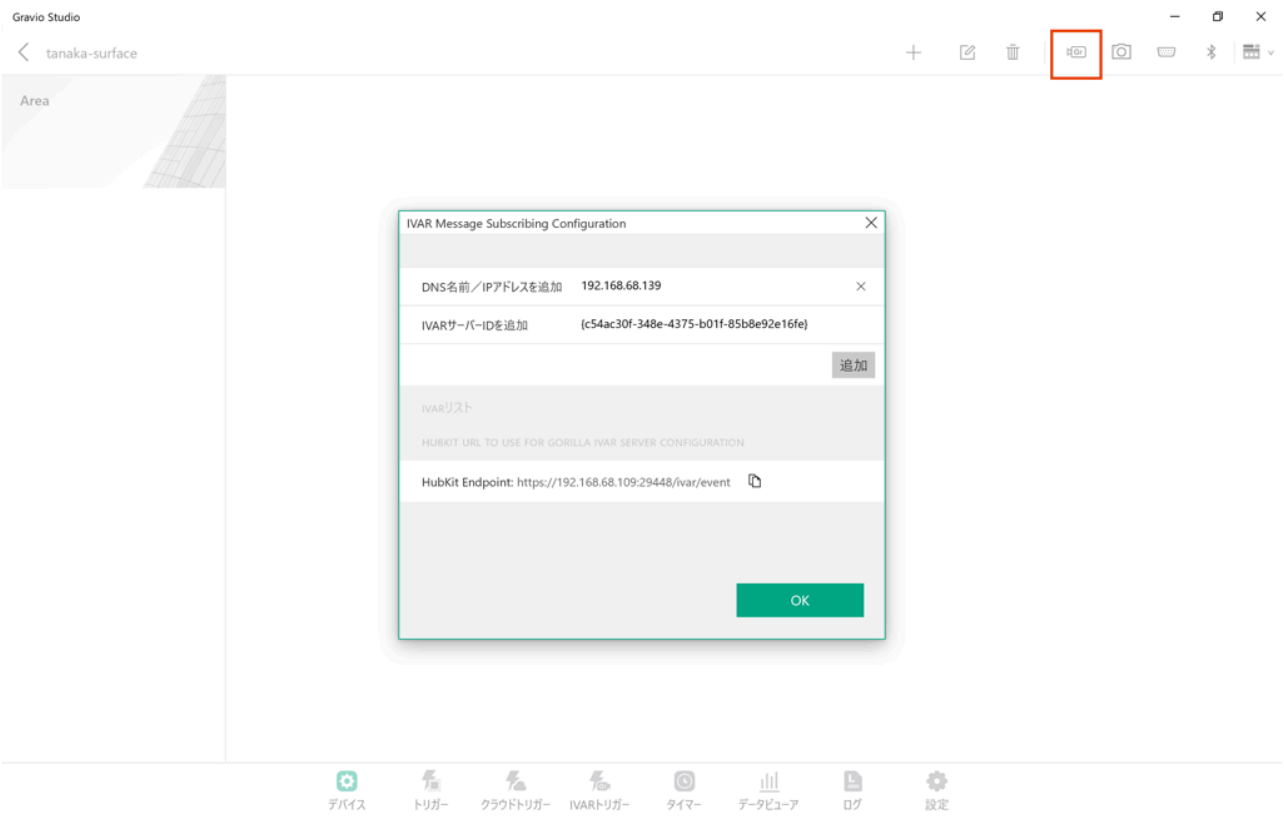
In order to link Gravio and IVAR, Log into IVAR using your admin account:



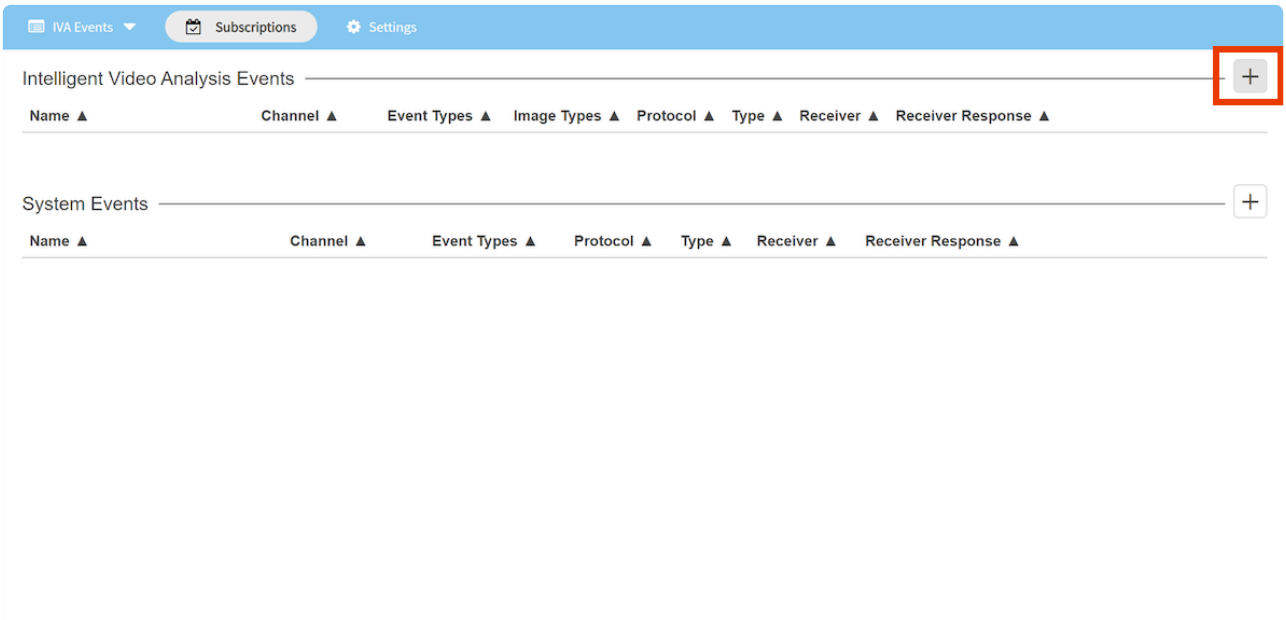
4. Setting IVAR in Gravio Studio



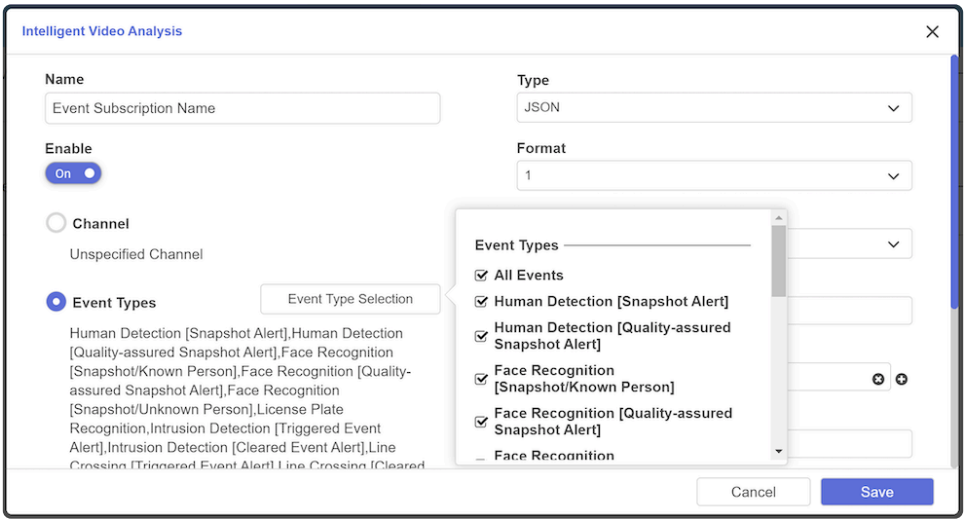
Select the System menu to open the About menu and copy the IVAR Information ID including {}.



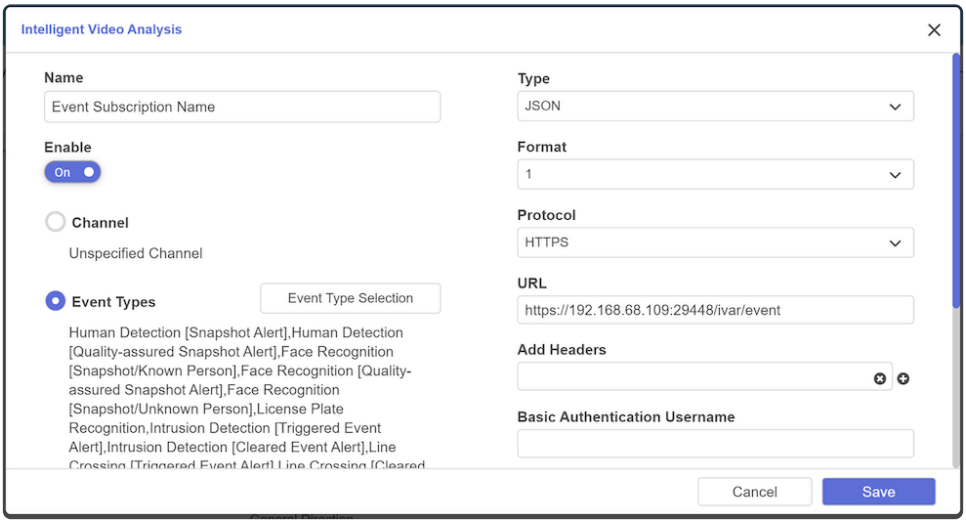
Select HubKit from Gravio Studio and open IVAR Message Subscribing Configuration from the device panel.
To add DNS name/IP address, enter the address of the PC where IVAR/BAP is installed.
To add the IVAR server ID, paste the ID you just copied.
Click the Add button to register.
Then copy the HubKit Endpoint.



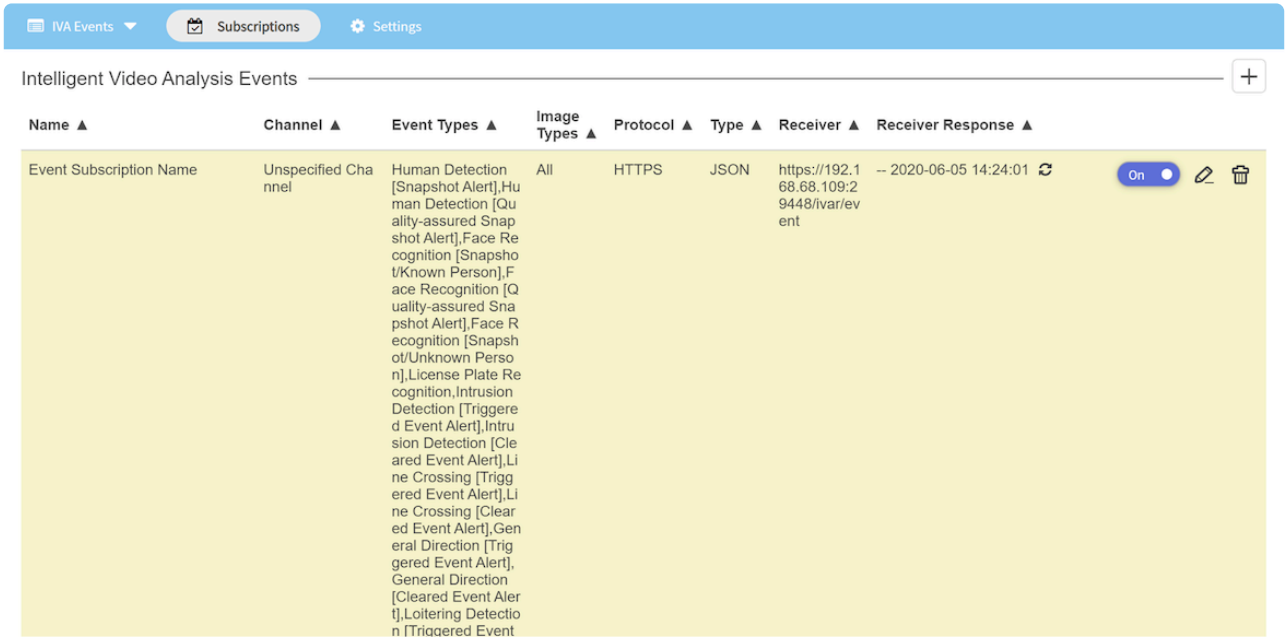
Select Events in IVAR to open Subscriptions
Press the + button.



Enable and open Event Types Selection and select All Events.

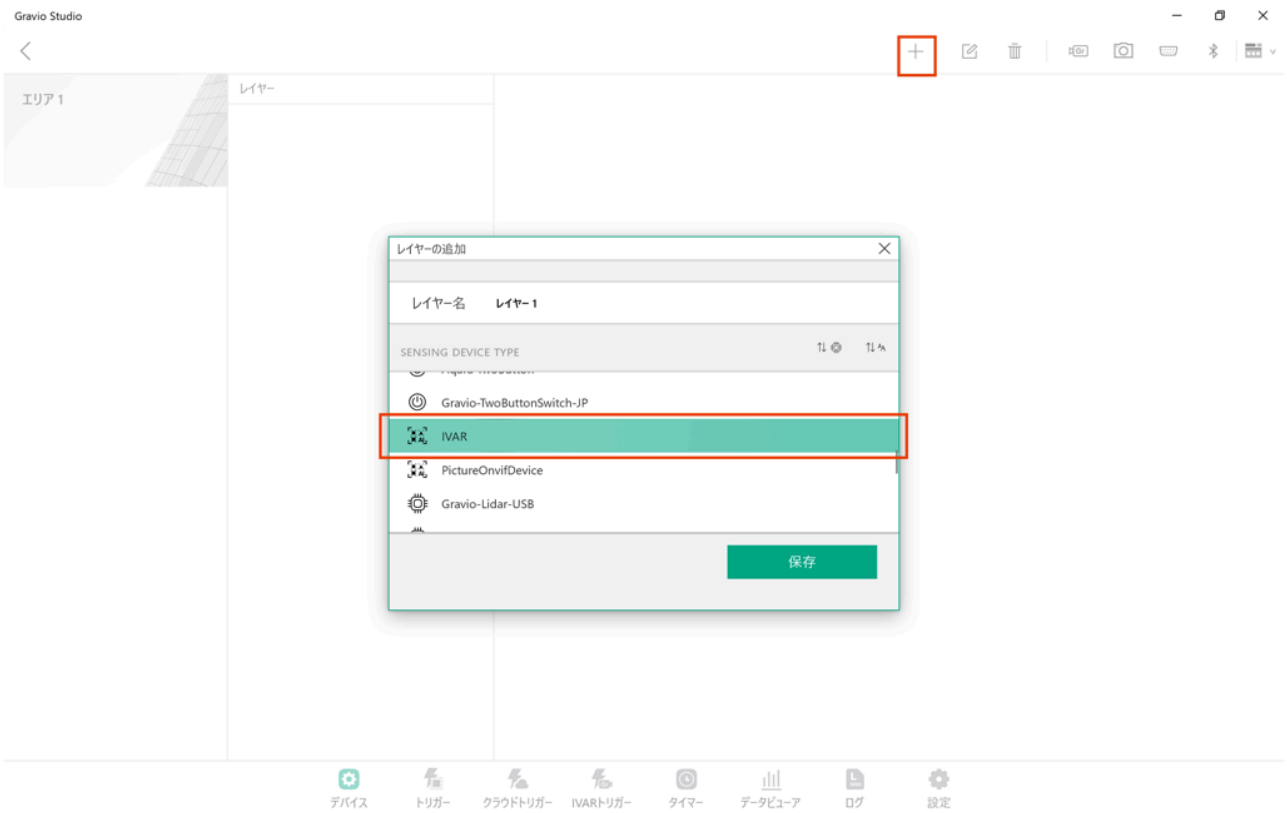


Change the Protocol to HTTPS and pace the HubKit Endpoint you just copied to the URL. Press the Save button to save.

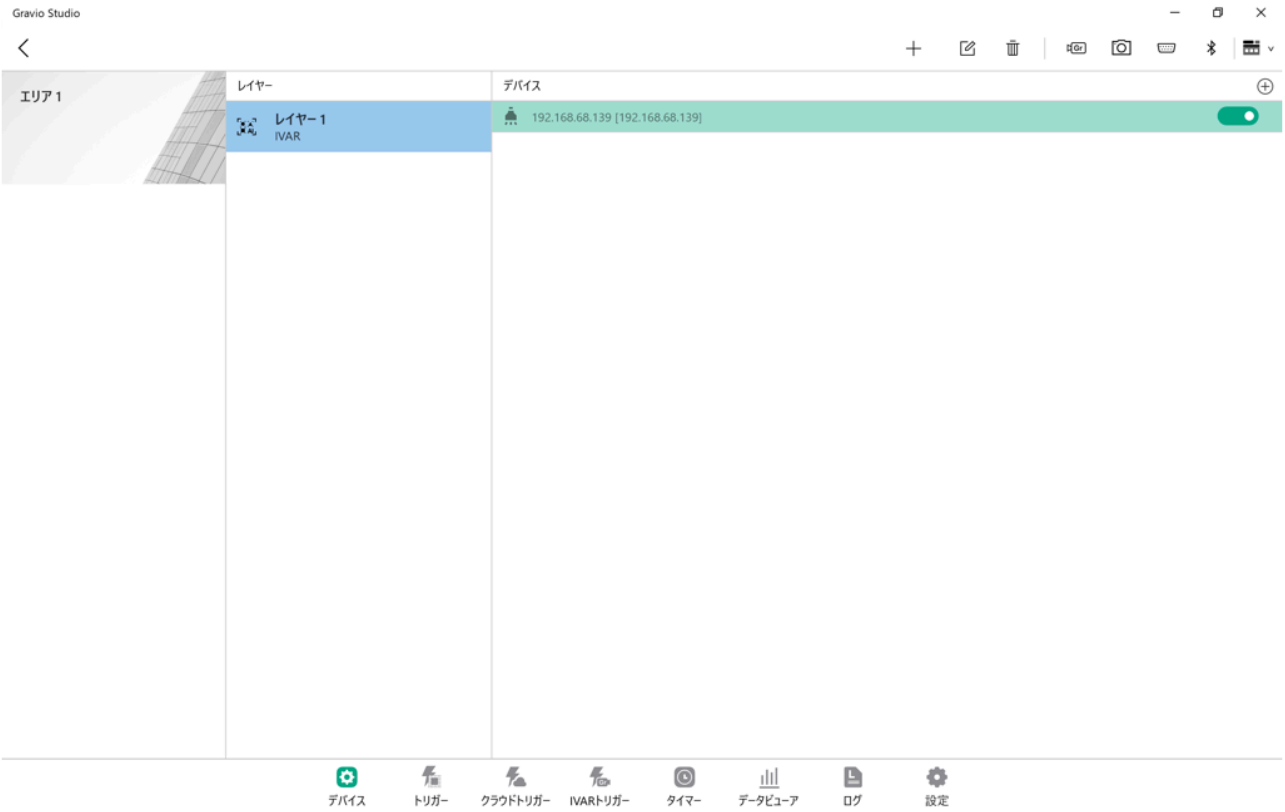


Once saved, it will be displayed in the event list.
Click the trash can icon to delete.

5. Event registration

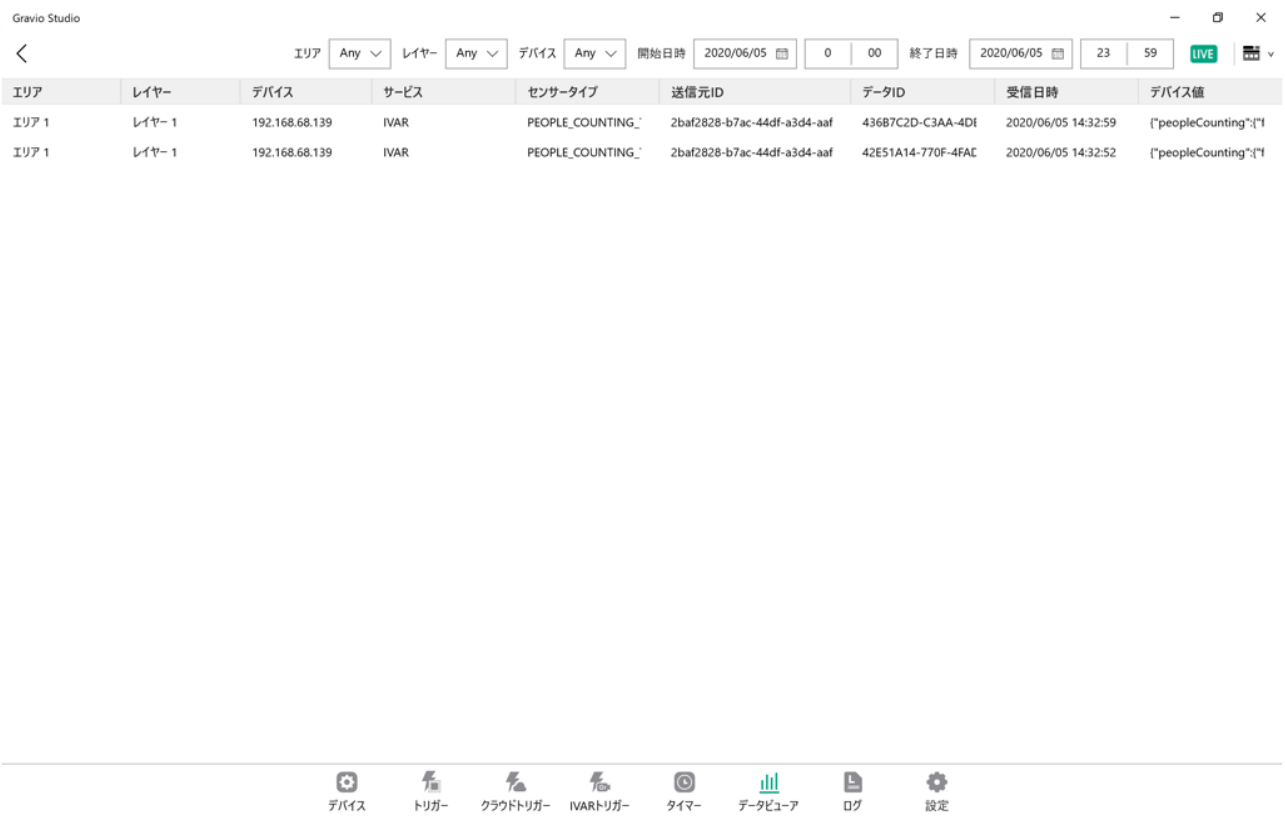


Select a device in Gravio Studio and add areas and layers.
Select IVAR as the layer and press the save button.



Activates the saved IVAR layer.

6. Confirm event reception in Data Viewer in Gravio Studio



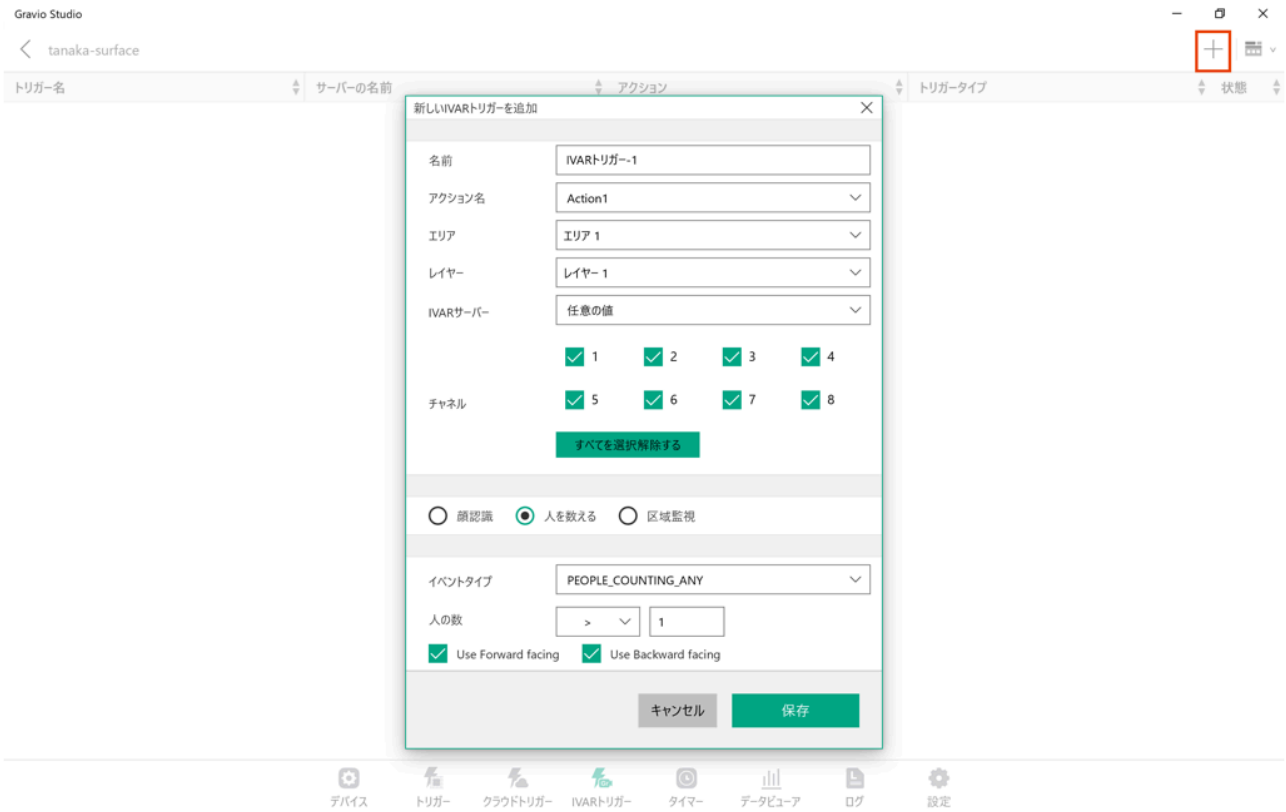
Open the data viewer and verify that IVAR is sending the detected events.

7.4.3. Setting up IVAR Triggers

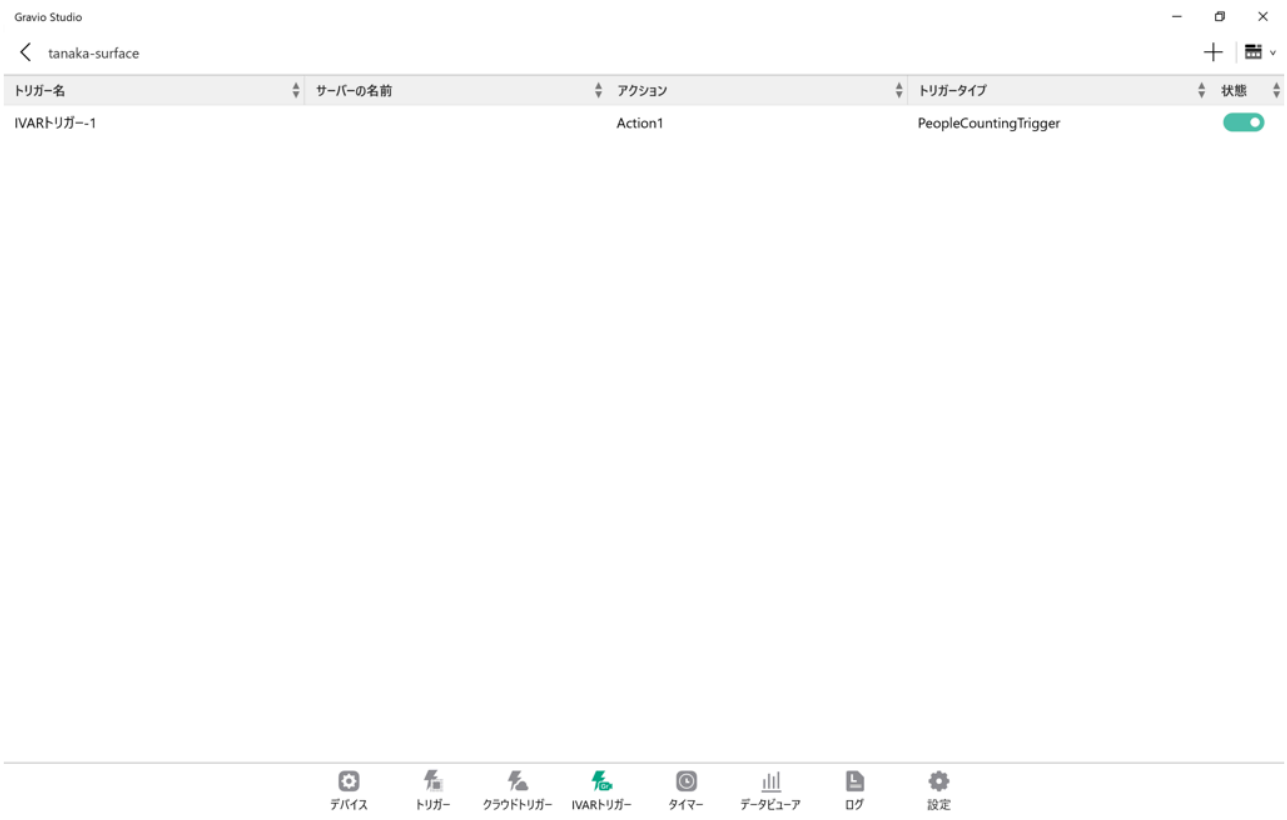
7. Creating an action to be called by IVAR

Create an action to call on the IVAR trigger. In this example, we will create an action that triggers a Slack message. If you have Slack integrated, this is an easy way to confirm that the action has been triggered.

8. Setting IVAR Trigger



Select IVAR Trigger and press + button to open Add New IVAR Trigger.
Select the action, area, and layer according to the setting value, and when counting the number of people, select PEOPLE_COUNTING_ANY as the event type.
Enter conditions such as >1 and >0 for the number of people.
Press the save button to save.



Activate the saved IVAR trigger.

9. Confirm execution of action in IVAR trigger

The specified action is executed when an event that is enabled and matches the IVAR trigger condition is sent from IVAR.

Since we are triggering a Slack component in this example, a Slack notification will should be triggered.

8. Advanced

8.1. How to create a machine learning model for computer vision object recognition

One of the big new features of Gravio 4 is the way you can create inference files to use Artificial Intelligence and Machine Learning to detect objects in visual feeds, such as from CCTV security camera systems based on ONVIF.

Please refer to the ONVIF section of this documentation to learn how to connect your network camera to Gravio 4.

Please note that you need a Gravio Enterprise Software License to create and deploy your own image inferences within your Gravio Edge IoT infrastructure. It support object detection and image classification for images. Video support is coming soon.

Once you have the possibility to create your own image inference files and teach Gravio to detect anything you want it to recognise in a picture, the possibilities are only limited by the quality of the picture of your camera and your learning model accuracy.

In essence, Gravio 4 leverages the power of Google's TensorFlow framework.

Our AI expert team will be able to help you creating these bespoke models, please get in touch with us on sales@gravio.com.

Once your model is created, you can upload it to your Gravio Coordinator, from which your Gravio Edge Nodes can deploy the inference files locally and therefore interact with the local camera systems. To deploy such a model, please consult your Gravio Coordinator documentation.

This section describes how to create a machine learning model for object recognition (Darknet YOLO v2 format) and convert it to a format for use with TensorFlow. Currently, that's what the Edge IoT platform requires to function at its best.

(Please note, TensorFlow model formats work with Gravio HubKit Version 4.0 later only.)

These are the steps required:

1. Collect images for training. The more the better the model will be later.
2. Create the training data from the collected images.
3. Create a machine learning model from the training data.
4. Convert the machine learning models to a format TensorFlow can understand.

Supplemental Material: Annotation Data Creation with LabelImg

1. Collect the Data Images

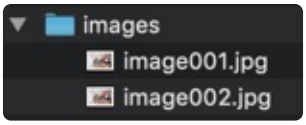
Prepare the image data to be used for the machine learning data of object recognition:

1. Create a new folder for storing images in the working folder on your computer and name the folder `images`.

2. Prepare the image data that contains the object to be recognized and place it in that newly created `images` folder.

Please note the following points.

- The image file format should be JPEG, and the file extension should be ".jpg". (Mixing different file extensions, such as .jpeg or .png, will result in errors during learning.)
- The image size should be at least 416 × 416 pixels.
- The number of images should be several hundred to several thousand for each object to be recognized.
- Use alphanumeric notation and sequential numbers for file names to make management more convenient. for example: `image001.jpg`, `image002.jpg`, etc.



2. Create the Training Data

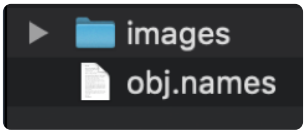
The training data consists of (1) prepared image data, (2) class information data. The latter is a list of names of objects to be recognized. Thirdly, we require annotation data (3) (annotation information indicating the rectangular area in the image of the object to be recognized) for each image. Here is how to add Class information data and annotation data:

1. Class information data is a list of names of objects to be recognized. Create a new file with a text editor and write one item to be recognized on each line. For example, if you want to build a learning model that recognizes three objects: `person`, `dog`, and `cat`, put the labels on a line each. The ID numbers of the objects to be recognized are assigned by sequential numbers starting at 0. (person=0, dog=1, cat=2)

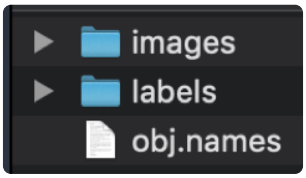
```
person
dog
cat
```

2. Complete the file editing of the class information data and save it. You can name the file as you wish, but we will name it `obj.names`, under the working folder on the same computer as the images folder.

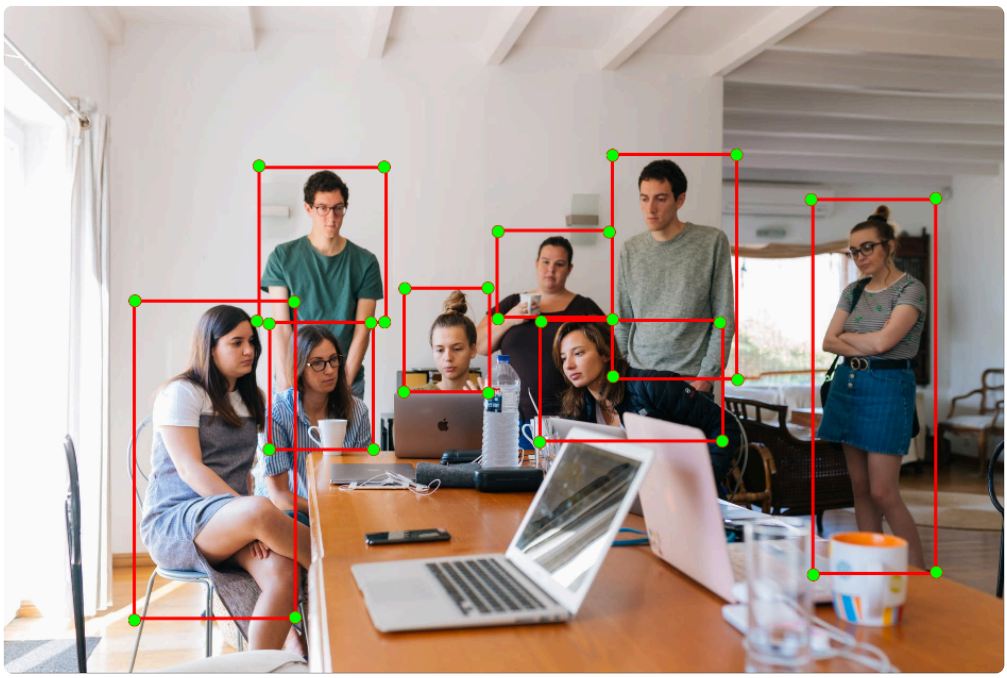
You can name the file anything you like.



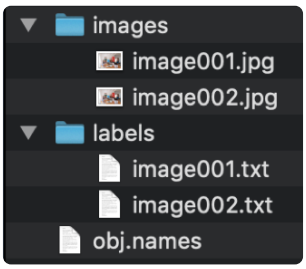
Next, in the working folder, create a new folder for storing the annotation data files and name it `labels`. The folder directly under the working folder will look like like this:



4. Annotate the objects in each image under the `images` folder. The annotation for each object is represented by a rectangular area. In the example photo below, the annotation of “person” is illustrated.



In practice, the annotation data is saved in the form of a text file within the `labels` folder. The filename extension `.jpg` of the corresponding image data must be saved with an identical name but with the extension converted to `.txt`.



In the annotation data file, each line should contain the following information as annotation data for one object:

`object ID` (number), `coordinates of the center point of the rectangular area` (x,y), and `size` (w,h).

The first line in the next screenshot represents the identifier `0` (=person), (x,y)=(`0.221543`, `0.712291`),

(w,h)=(`0.166489`, `0.532322`). The coordinates (x,y) and size (w,h) of the rectangular area of the object to be recognized are expressed as a ratio to the whole image, and the values range from 0 to 1.

```
0 0.221543 0.712291 0.166489 0.532322
0 0.314362 0.618915 0.112766 0.278532
0 0.318617 0.416600 0.125532 0.346369
0 0.542021 0.486433 0.140426 0.281724
0 0.644149 0.591381 0.189362 0.255387
```

The following tool is available for adding annotations to an image using GUI (in English). For more details, please refer to Annotation Data Creation with LabelImg at the end of this document.

LabelImg is available from https://github.com/tzutalin/labelImg

3. Creating a Machine Learning Model

Create a learning model in the following steps:

1. Prepare Darknet

In the terminal environment of the computer where you want to create the training model, create the darknet environment by following the steps below.

```
git clone https://github.com/pjreddie/darknet
cd darkent
make
```

If you are building the training environment on a GPU (CUDA) environment, edit the darknet/Makefile and set `GPU=1` as the first line before executing the last make command.

2. For the first training, get the trained weight data from the following location.

```
wget https://pjreddie.com/media/files/darknet19_448.conv.23
```

3. Set up your training data.

```
cd data
mkdir learning_data (folder of your choice)
```

Place the image data, annotation data, and class information data prepared in the steps 1 and 2 above in that folder:

```
darknet/data/learning_data/images/*.jpg
```

```
darknet/data/learning_data/labels/*.txt
darknet/data/learning_data/obj.names
```

4. Prepare the training data list and validation data list.

```
darknet/data/learning_data/train.txt
darknet/data/learning_data/test.txt
```

In `train.txt` and `test.txt`, enter the file paths from the `data` folder onwards as shown below to avoid duplication of file names. Only the image files listed in `train.txt` will be used for learning.

```
data/learning_data/images/img001.jpg
data/learning_data/images/img002.jpg
data/learning_data/images/img003.jpg
...
```

5. Prepare the learning configuration file.

Create a new file `darknet/cfg/obj.data` and enter the following:

```
classes = 3
train = data/learning_data/train.txt
valid = data/learning_data/test.txt
names = data/learning_data/obj.names
backup = backup
```

6. Prepare the model configuration file. In this section, we will copy the already prepared `darknet/cfg/yolov2-voc.cfg` configuration file and rewrite its contents.

```
cp darknet/cfg/yolov2-voc.cfg darknet/cfg/obj.cfg
```

Adjust the newly created `obj.cfg` as follows: The value of Filters will be 40 if the number of classes is 3: (`(3+5)*5 = 40`)

```
Line 3: batch=64
Line 4: subdivisions=8
Line 237: filters=(actual number of classes to be recognized + 5) * 5
Line 242: anchors (*)
Line 244: classes=3 (actual number of classes to be recognized)
```

(*) You can continue learning with the values of anchors as they are, but recalculating the values of anchors using the following tools may improve the accuracy.

Reference: Recalculation tool for anchors values https://github.com/Jumabek/darknet_scripts

7. Start the learning process

In the darkent folder, execute the following to generate the training model data (*.weights) under the darknet/backup/ folder.

```
. /darknet detector train cfg/obj.data cfg/obj.cfg darknet19_448.conv.23
```

When the training starts, the output will be similar to:

…
Region Avg IOU: 0.905463, Class: 0.626457, Obj: 0.836158, No Obj: 0.006927, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.837368, Class: 0.976554, Obj: 0.842887, No Obj: 0.006556, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.893971, Class: 0.960798, Obj: 0.876742, No Obj: 0.007555, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.872773, Class: 0.711470, Obj: 0.888209, No Obj: 0.006718, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.894234, Class: 0.986793, Obj: 0.884681, No Obj: 0.006898, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.847557, Class: 0.994500, Obj: 0.849292, No Obj: 0.006747, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.802273, Class: 0.650419, Obj: 0.794552, No Obj: 0.006926, Avg Recall: 1.000000, count: 8
Region Avg IOU: 0.866342, Class: 0.840821, Obj: 0.860806, No Obj: 0.006149, Avg Recall: 1.000000, count: 8
9531: 0.661146, 0.704077 avg, 0.001000 rate, 2.704480 seconds, 609984 images
…

The number at the beginning of the last line (shown in red) represents the number of steps. As the learning progresses, the average value of the error (shown in blue) becomes smaller.

Learning can be stopped at any time, but to create a highly accurate recognition model, the average value of the error should be less than 1. In some cases, more than tens of thousands of steps may be required, and depending on the operating environment, it may take several days or weeks to complete.

8. Resume learning

To resume learning, do the following: Under the backup folder, there will be a number of files with the extension .backup or .weights. If you specify one of the files with the most recent extension (usually .backup), you can resume training using the last saved training model.

```
. /darknet detector train cfg/obj.data cfg/obj.cfg backup/(the name of the lat
est file under the backup folder)
```

As a result of the above steps, the following three files will be available after the training is completed.

darknet/backup/*.weights (the part marked with * specifies the name of the latest file generated under the backup folder)
darknet/cfg/obj.cfg,
darknet/cfg/obj.names, darknet/cfg/obj.names

4. Converting a machine learning model to the format used by TensorFlow

The following example shows how to convert three files (`*.weights`, `obj.cfg`, `obj.names`) into protocol buffers format (`*.pb`) that can be used by programs written in TensorFlow. We are using the DarkFlow conversion tool (https://github.com/thtrieu/darkflow):

1. Installing the TensorFlow environment

Follow the official TensorFlow installation procedure to install TensorFlow in the computer environment where you will work. (Python 3 example is shown here.)

https://www.tensorflow.org/install

```
pip3 install --upgrade pip
pip3 install tensorflow==1.15
```

(Note: The version of TensorFlow must match the environment in which the converted model will be used.)

2. Install OpenCV for Python.

```
pip3 install opencv-python
```

3. Set up DarkFlow with the following steps.

```
git clone https://github.com/thtrieu/darkflow.git
cd darkflow
sed -i -e 's/self.offset = 16/self.offset = 20/g' darkflow/utils/loader.py
python3 setup.py build_ext --inplace
pip3 install -e .
pip3 install .
```

4. Create a models folder to store the three files created in section 3, “Creating Machine Learning Models”. The name of each file to be stored is assumed to be obj.weights, obj.cfg, and obj.names.

```
mkdir models
```

(Store obj.weights, obj.cfg, and obj.names under the models folder)

5. Under the DarkFlow folder, convert each file with the following command.

```
flow --model models/obj.cfg --load models/obj.weights --labels models/obj.name
s --savepb
```

6. The above steps will generate two files under the `built_graph` folder after the conversion.

`obj.pb` … Neural network model file for machine learning
`obj.meta` … Meta information file

These files can be deployed to your Gravio infrastructure for distribution to the edge.

8.2. Annotation Data Creation with LabelImg

In this chapter, we learn how to use LabelImg.

At the end of “2. Create the Training Data” of our instructions on how to create Image Inferences, we introduced the annotation data creation tool **LabelImg**. Learn here how to create annotation data using LabelImg. These are the steps:

1. Check your OS version (Windows, Mac, Linux, etc.) and the current python version support in the “Installation” section of the (LabelImg Github Page) and install it according to the description below.

2. Follow the “Steps (YOLO)” section in the “Usage” part of the LabelImg documentation.

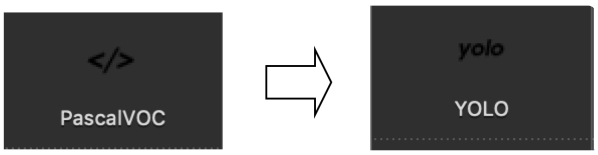
(1) Write the name of the class you want to recognize in `data/predefined_classes.txt` under the folder where LabelImg is installed. (2) Write the same `obj.names` that you created in step 1 of Create Model. For example, if you want to recognize `person`, `dog`, and `cat`, the contents should be as this:

```
person
dog
cat
```

3. Start LabelImg with the python command as described in “Installation”. For example, in a python3 environment, type the following command in a terminal environment to start it.

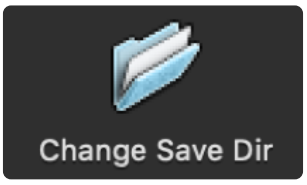
```
python3 labelimg.py
```

4.Click on the “PascalVOC” button in the toolbar on the left of the LabelImg screen to switch the display to “YOLO”. (It is important to switch the setting to YOLO when you start the program.)



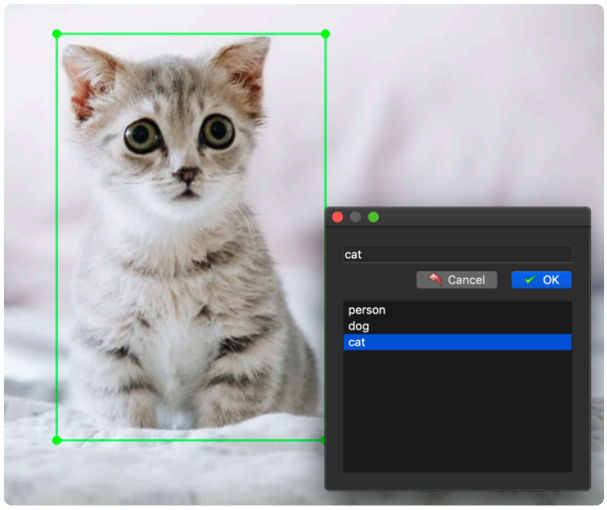
5. Click on the “Open Dir” button on the toolbar on the left of the LabelImg screen, and select images, which you created as a folder for storing images, in the working folder on your PC. The first image in the images folder will be displayed in the center of the LabelImg screen.

6. Specify the destination for the annotation data file after the operation. Click the “Change Save Dir” button on the toolbar on the left of the screen, and select the “labels” folder in the working folder on your computer that you created in step 3 of Create Model



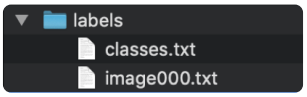
7. Press “w” on the keyboard while the screen is displayed to add annotations to the image. Drag the mouse over the image to determine the rectangular area.

8. Once you have determined the rectangular area with the mouse, a screen for selecting the appropriate object from the object names listed in `data/predefined_classes.txt` will appear. In the following example, `cat` is selected.

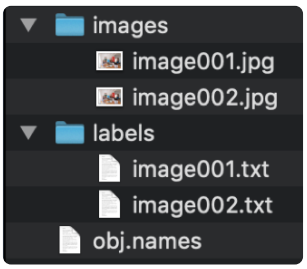


After you have finished determining the annotation rectangle area for all objects in one image, press the “Save” button on the toolbar on the left of the screen. An annotation data file named ~.txt corresponding to the image name will be generated under the “labels” folder specified in step 6 above. At the same time, a file named `classes.txt` will also be generated under the labels folder.

See this example:



10. After completing the annotation process for all images in the images folder, move the `classes.txt` file in labels to a working folder on your computer and rename it `obj.names`.



This is it. That’s how you use LabelImg to annotate your pictures. If you have any questions or concerns, please don’t hesitate to join our slack channel at https://bit.ly/gravioslack

8.3. Using the Gravio Coordinator

Because Gravio is a distributed system consisting of either computers (macOS, Windows or Linux) running the HubKit or the Gravio Hub itself, there is a need to coordinate those edge nodes.

More information about the Gravio Coordinator can be found in the Gravio Coordinator documentation

9. Integrating Third Parties

10. Gravio Hub (Gravio Hub 2)

For the Gravio Standard package or higher, you get one or more Gravio Hubs. The Gravio Hub is the Edge Device that gathers all the data and processes. It is essentially a small computer. The specifications of the Gravio Hub are below:

CPU: Rockchip RK3399 (6 core) RAM 4GB
Radio: Wireless LAN, Zigbee, Bluetooth
Internal Storage: 32GB eMMC/Ubuntu 1804 with Gravio HubKit pre-installed
Dimensions: W97mm × L97mm × H29mm
Weight: 135g

10.1. Indicator light status and button operation on Gravio Hub 2

indicator light status

The LEDs on the side of the Gravio Hub 2 light up in various colors and patterns to indicate the following

LED	Status	Comment
blue	when first booted out of the box	Access point mode
White light on	after Wifi setup or wired LAN connection	normal operation mode
Solid green	LTE Mode	disconnected WAN, connect to the Gravio Hub via its Access Point Wifi and the IP address 192.168.96.1
Flashing green for 5 seconds	LTE Mode + WAN connected or wifi router mode (HubKit activated) , connect to the Gravio Hub via its Access Point Wifi and the IP address 192.168.96.1	
Red Dot Light	Docker initialization error	
Blue LEDs blinking with gradient	command mode	See button operation

Button Functions

In Gravio Hub 2, double-clicking the button enters the command mode to execute the following commands.

AP Mode (Setup mode)

Press the button once to put the Gravio Hub 2 in AP mode; the LED will be lit blue.

Caution : When in LTE mode, it does not switch to AP mode.

Initialize Wifi information and return to access point mode

Clicking the button once will initialize the Wifi information and return to access point mode.

Power Off

If the button is held down, the Gravio Hub 2 will power down and the LED will turn solid RED.

Exit the Command Mode

Double-click to exit command mode; the LED will return to a solid white LED.

10.2. Migration Guide from Gravio Hub to Gravio Hub 2

This section describes the procedure for customers using Gravio Hub (“Hub1”) to migrate to Gravio Hub 2 (“Hub2”).

Before starting

It is assumed that the data in the currently operated Hub1 will be backed up and restored to Hub2 for operation.

Please start Hub1 and configure the network settings for Hub2 using UpdateManager in advance.
Since Hub2 will be connected to the same network as Hub1, set up a WiFi access point or wired LAN connection or set up a temporary IP address using DHCP/static address while performing the migration procedure.

Please start UpdateManager on Hub1 and Hub2 and keep them logged in before proceeding with the migration procedure.

migration procedure

The migration procedure to Hub2 consists of the following 7 steps.

1. Update the version of Coordinator
2. Update the version of HubKit
3. Update the version of Zigbee
4. Backup your data in Hub1
5. Download the backup file in Hub1
6. Restore on Hub2
7. Change the network on Hub2

For more information on how to use UpdateManager, please refer to here.

1. update the version of Coordinator

Make sure that the version of the Coordinator software in Hub1 and Hub2 are the same.

In Hub1, update to the latest version from the Gravio Coordinator software update in the Gravio Settings tab.
In Hub2, update the Gravio Coordinator software to the latest version from the Gravio Settings tab.

2. Update HubKit Version

Make sure that the HubKit software versions of Hub1 and Hub2 are the same.

In Hub1, update to the latest version from the Gravio HubKit software update in the Gravio Settings tab.
In Hub2, update to the latest version from the Gravio HubKit software update in the Gravio Settings tab.

3 Updating the Zigbee Version

Make sure the Zigbee software versions of Hub1 and Hub2 are the same.

On Hub1, update the Zigbee software to the latest version from the Gravio Settings tab.
In Hub2, update the Zigbee software to the latest version from the Zigbee software update in the Gravio Settings tab.

4. Backup data in Hub1

On Hub1, click the Backup button in the Gravio Settings tab under HubKit / Coordinator Data Backup to back up the data.
It will take some time to process the data stored in Hub1 so that it can be downloaded.

If an error occurs during the process, it may be due to the large amount of data being saved, so please delete unnecessary data by deleting the media file and try again.
Refer to here for the location of the media file.

5. download the backup file in Hub1

In Hub1, under the Gravio Settings tab, press the Export button in HubKit / Coordinator Data Backup to download the backup file to your local PC.

6. Restore in Hub2

In Hub2, import the backup file saved in the local PC by HubKit / Coordinator Data Backup in the Gravio Settings tab.
By importing the backup file of Hub1 into Hub2, the data of Hub1 has been migrated to Hub2.

7. Change the network of Hub2

If you want to operate Hub2 as a new hostname or IP address different from that of Hub1, the migration is completed at this point.
The network configuration information is not inherited by importing a backup file, so if you want to use the same hostname and IP address as Hub1, stop Hub1 and change it in the network settings of Hub2.

11. Gravio Hub (Gravio Hub version 1)

Gravio Hub is available for rental to subscribers of Gravio Standard or higher service menu.

Gravio Hub version 1 is a square housing model.

Gravio Hub version 1 users can exchange it for Gravio Hub 2 equipment, please contact us.

11.1. Gravio Hub (Gravio Hub version 1) initial settings

This section will discuss the initial setup of Gravio Hub.

To use the Gravio Hub, you will need to do the following two things.

1. Configure WiFi network settings to connect Gravio Hub to your network.
2. Gravio Hub is running on a LinuxOS based on Ubuntu1804. It is designed to log in to LinuxOS as needed. For this reason, we ask that you change the password for your login account to login to Linux in the initial settings when you use it to increase security.

Please prepare your WiFi network settings (network name and password to connect to) and the new password for the login account in advance.
Note: Please save the password for your new login account in a safe place. If you lose the password, you will not be able to log in to Linux.

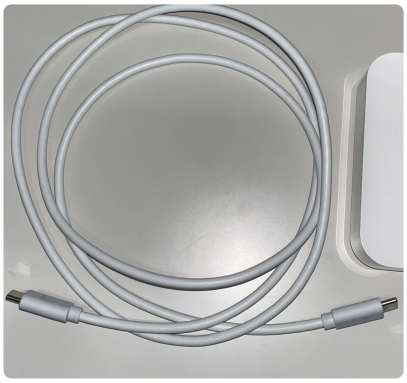
To configure these settings, we have prepared a function and an application that allows you to do so from your iOS or Android device.
Please follow the instructions below to set them up.

The following accessories are rented with the Gravio Hub, including the following Gravio Hub

- Gravio Hub main unit

- Gravio USB-C adapter !

- Gravio USB-C Cable



- Power supply box !

The Gravio Hub has one USB-C connector to which you can connect a USB-C adapter or a USB-C cable.

When connecting the USB-C cable to the power cable only

Connect the power box to the power supply by making the following connections.

installing the Gravio Hub in the LAN

Connect the Gravio Hub to the power supply and start it up.

11.2. Indicator light status and button operation on the Gravio Hub (Gravio Hub version 1)

indicator light status

The Gravio logo on the Gravio Hub lights up in various colors and patterns to indicate the following

LED	Staus	Comment
Blue	When first started up out of the box	Access Point Mode
White	After Wifi setup or wired LAN connection	Normal Mode
Flashing green	At initialization of Docker	This process takes about 2 minutes
Flashing green for 5 seconds	Finished setting of Docker	
Flashing red for 5 seconds	Docker initialization error	
Blue	Initialize Wifi information and return to access point mode	Hold the button down for 3 seconds to enter this state.
Yellow	Prepare to stop power supply	Holding the button for 5-6 seconds, the Gravio Hub will be ready to stop.
Red	Power off	If the button is held down for more than 7 seconds, the Gravio Hub will stop powering

button operation

initialize Wifi information and return to access point mode

Hold the button for 3 seconds to initialize Wifi information and return to access point mode

prepare to power off

Press and hold the button for 5-6 seconds to prepare the Gravio Hub for deactivation

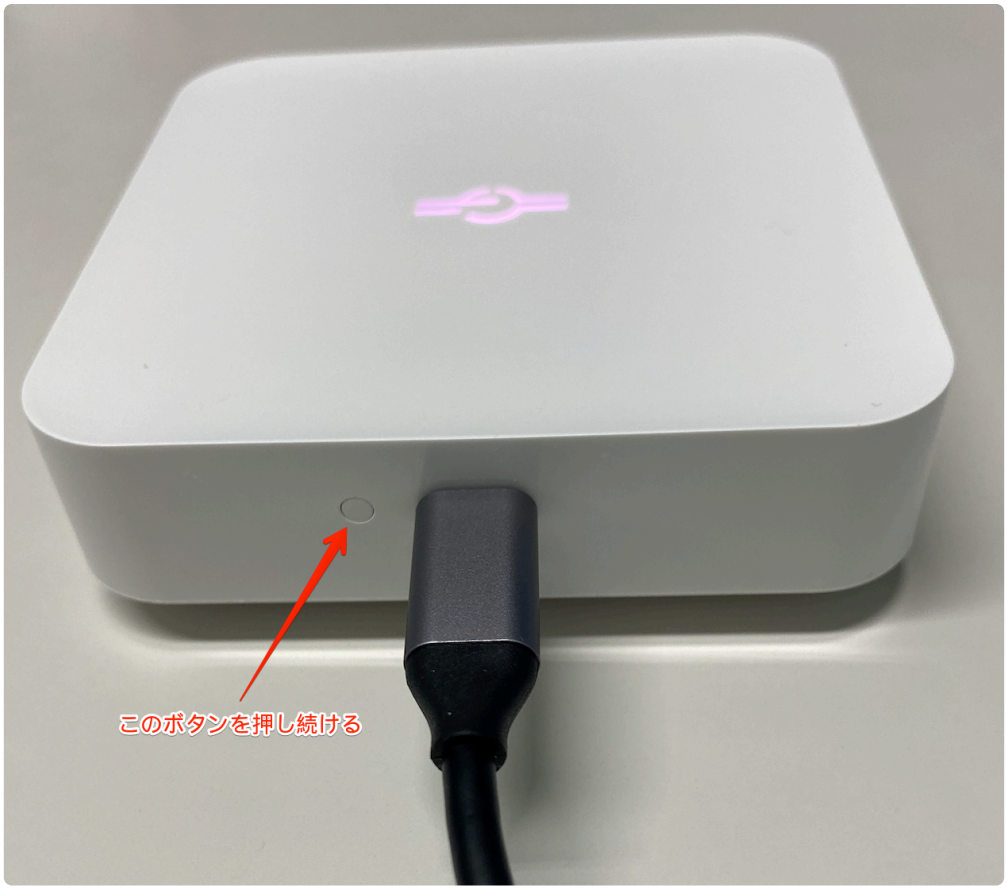
power off

Holding the button down for more than 7 seconds will power off the Gravio Hub

11.3. How to stop Gravio Hub

How to stop Gravio Hub (version 1)

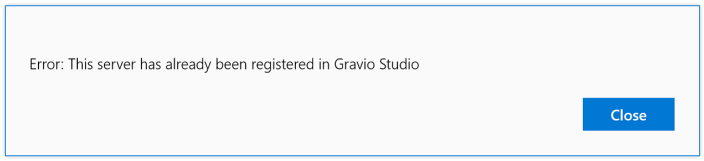
To stop Gravio Hub, please follow the steps below.
Press and hold the button shown below, and the Gravio symbol on the top surface will change from “blue” to “yellow” to “red” and finally the light will turn off.
This will turn off the Gravio Hub.



12. Troubleshooting

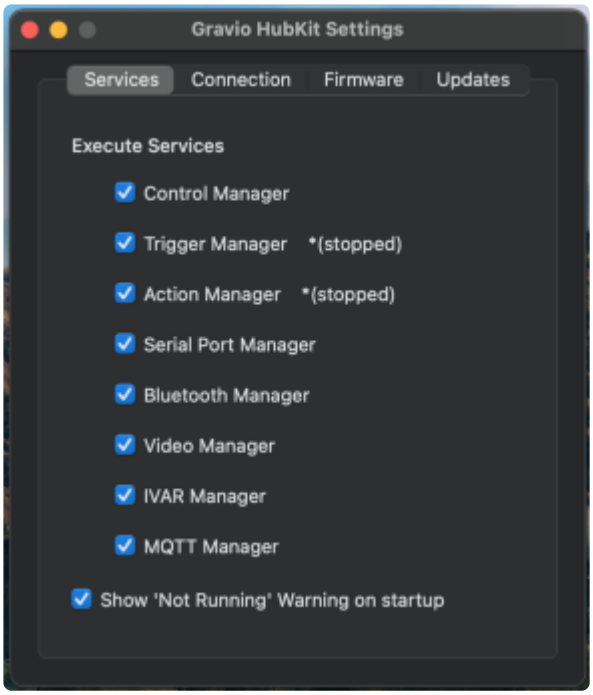
In this section we talk about how to identify and rectify issues with Gravio.

12.1. General Gravio Troubleshooting

Symptom	Explanation	Remedy	Comments
Gravio Studio is not behaving as expected	Gravio Server connection is unstable or Gravio Server is not running	Ensure that there is a stable connection between Gravio Studio and the Server is running	
Windows: The Gravio Edge Service remains in “starting…” mode	There is likely to be an issue with the stored data in C:\ProgramData	Use the Task Manager > Details view, find the Gravio*exe files and end them by clicking on “End Task”	
Some Sensors are not sending information	Either they are out of range or out of power	If the sensors are energy harvesting, please put them in the sun or other natural light for a few hours and see if that fixes the issue. Also move closer to the dongle receiver.	D4
			
Step of Action is not being executed	There may have been an error in the previous step	Check if any of the previous steps had an error	You can set t exception ha and determin Gravio shoul case of an er exception
New dongle or device doesn’t appear in options or drop-down	Driver may not be installed	Find the respective drivers and install them	
Gravio Studio is not connecting properly to Gravio Servers	The relevant ports may not be open	Please check the port requirements under Which ports does Gravio require?	
Gravio Studio is not connecting properly	The Server may have stopped	You can use the GravioTools (Windows), the Menu Bar App (Mac OSX) or the command `systemctl stop/start gravioxxxxxx.sevice`	

		to restart the server. The best order is to stop edge, action and then control. And vice versa for start.	
The Gravio Dongle is not recognised	The driver may not be installed	Install the driver by launching the Gravio Tool and select “install driver” from the tray	
Finding out Gravio Version		In Windows, you can see it in the System Settings, in macOS you can see it in the “about” screen and in Linux you can use the command `sudo apt search gravio`	
Where do I find the Sensor DB?	Place where the sensor data is stored	on Linux that’s in `/var/opt/gravio/action/dB` however in Mac, you will find it in `/Library/Application Support/Gravio/action/db` and in Windows in `C:\ProgramData\Gravio\action\db`	
How do I update the Zigbee dongle firmware?	There may be a newer version of the Firmware	Mac: You can use the file `/Applications/Gravio\ HubKit.app/Contents/Resources/gssosx/zigbeefirmwareupdate` and run `sudo ./zigbeefirmwareupdate -f </path/to/your/firmware/file> -d /dev/<yourdeviceid>`, for example `$ sudo ./zigbeefirmwareupdate -f /Users/username/Downloads/V0.8.0.2.bin -d /dev/cu.usbserial-DO00HOXS` , the device ID starts with `.cu`. The firmware upload can take up to 5	

		minutes	
How do I find out the ID of my USB dongle?	You may need to find out if the device is connected, and what ID it has	Mac: Try to run `/Applications/Gravio\ HubKit.app/Contents/Resources/gssosx/iotool`, if it shows something like `6015,0403,usbserial-DO00HOXS` this means your dongle is connected	More details be found in `/ y/Logs/gra gbeefirmwa`
How to disable camera detection?	If your computer is overloaded, you may want to switch off the camera detection to release resources	You can set `OpencvTypeInitializationException` to `true` in `edge/conf/opencv.config`	This is usuall hardware spe especially wit hardware tha able to cope load.
I cannot connect to my local HubKit on Windows. Why?	Loopback may be disabled	Please run the `WindowsLoopbackManager.exe` and ensure that “Gravio” is ticked on as an exception	If you do not the LoopBack Manager, you download it fr here
I cannot connect to my Gravio Hub and it’s not connected to the network	You may have locked yourself out of the Gravio Hub	Please long press the reset button while the Hub is connected to power to reset the Wifi settings to Access Point mode so you can reconnect using the password `graviohub`.	This will only the Gravio Hu settings (no c settings or da Access Point meaning you able to conne Wifi following instructions. the Gravio Lo is blue, this s that the Hub Access Point

	Please check if the ports are being used elsewhere, including from an old Gravio installation	Uninstall the other software	If it's an old G ensure that y uninstalled it including the o* folders fro r/local/bi ibrary/App on Support /Library/I
Some of the Gravio services are not starting	On Mac, there could be a conflict with Brew	There is a fix using http://flat-leon.hatenablog.com/entry/mac_runtime_error_libjpeg	Run these commands to Gravio Brew compatible: c r/local/li mv libjpeg b libjpeg. b.backup ln -s /Sys brary/Fram s/ImageIO. ork/Resour bJPEG.dyli PEG.dylib mv libtiff b libtiff. b.backup ln -s /Sys brary/Fram s/ImageIO. ork/Resour bTIFF.dyli IFF.dylib mv libpng. libpng.dyl kup ln -s /Sys brary/Fram

			s/ImageIO. ork/Resour bPng.dylib G.dylib mv libGIF. libGIF.dyl kup ln -s /Sys brary/Fram s/ImageIO. ork/Resour bGIF.dylib F.dylib

I like to increase the logging level	Logging is not insightful enough	You can add a `config.json` file to `./HubKit/vidmgr/config/`	{
			"I
			"Level": '
			}
			}

12.2. Sensors Troubleshooting

Symptom	Explanation	Remedy	Comments
Energy harvesting door/contact sensor not reacting	The sensor may have run out of battery	Put the sensor into direct sunlight for a few hours. You can also see if the sensor connects to the dongle if a light flashes up at the dongle if you put the magnet next to the sensor.	The sensor also emits a regular signal every few hours with the current status (open/close) to indicate it's "still alive"

13. Appendix

13.1. Which ports does Gravio require?

Each server module of Gravio's HubKit, the Update Manager, and the Coordinator use the following ports:

• Inbound

Service Name / Process Name	Protocol	Port
Update Manager	HTTP	8080
Coordinator	HTTPS	443
Gravio Control Manager	HTTPS	29442
Gravio App Service	HTTPS	29443
Endpoint for IVAR	HTTPS	29448

• Outbound

Service Name	Protocol	Port
DNS	UDP/TCP	53
NTP	UDP	123
HTTPS Server	HTTPS	443

When installing HubKit and Gravio Studio on a separate PC, it is necessary to open the port for "Gravio Control Manager – HTTPS: 29442" in the firewall among these port numbers.

When using HubKit and Chrome cast, it is necessary to open the port for "Gravio App Service – HTTPS: 29443" on the PC where HubKit is installed among these port numbers.

13.2. Gravio HubKit Data Directory

When the Gravio HubKit is installed, it creates a data directory to store configuration files and data.

On Windows 10

`C:\ProgramData\HubKit\`

On macOS

`/Library/Application Support/HubKit/`

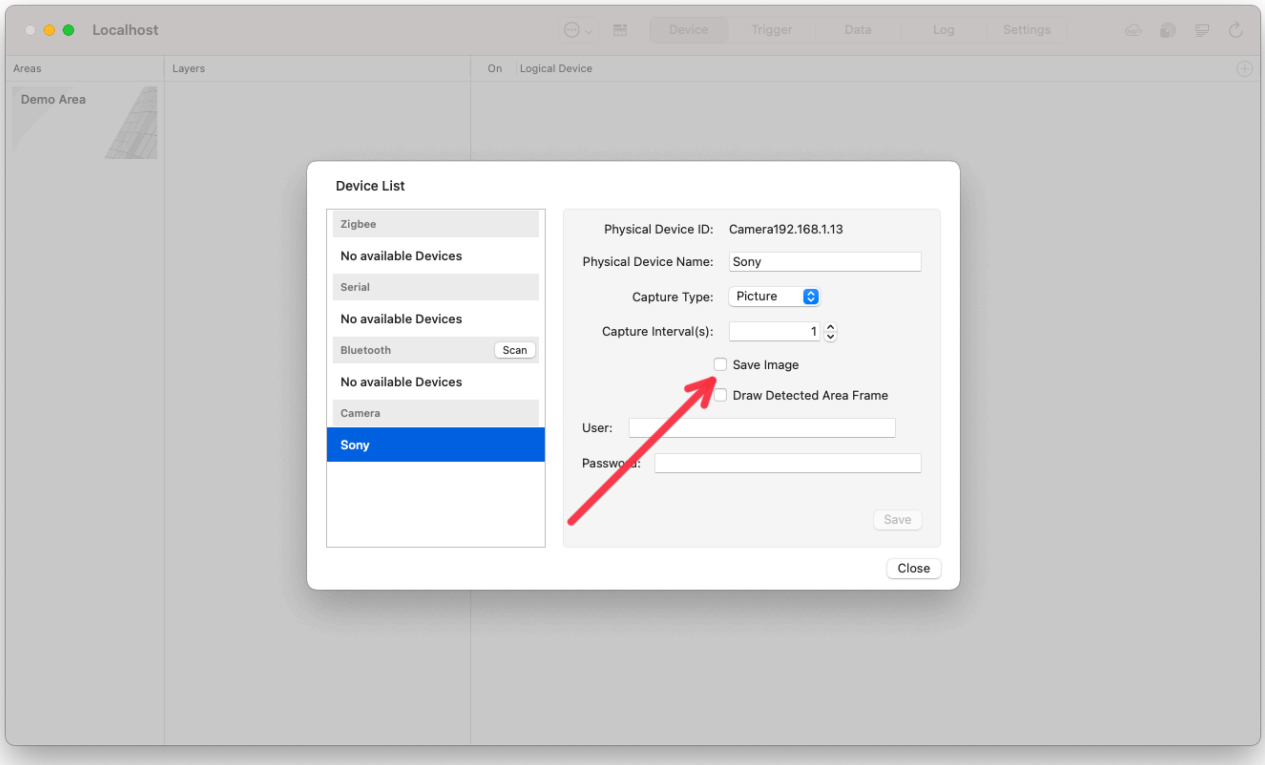
On Linux/Raspberry

On Linux/Raspberry, the following directories are mounted as volumes because the system runs on Docker.

/home/gravio/hubkitrepo4/data

13.3. Saving Media data path for Camera device

Images acquired from the camera by ONVIF or USB will be stored in the following directory for each camera device and stored in each date directory.



Windows 10

`C:/ProgramData/HubKit/mediadata`

Mac

`/Library/Application Support/HubKit/mediadata`

Linux/Raspberry/Gravio Hub

On Linux/Raspberry, since it runs on Docker, the following directories are mounted as volumes
`/home/gravio/hubkitrepo4/data/mediadata`

When viewed from within Docker, the directory is
`/var/opt/hubkit/mediadata`
directory.

If you like the path of the image to be stored in the sensor database, you will need to add a Camera layer and assign the camera to that layer. Note, for data to be captured in the database, it needs to be in a layer.

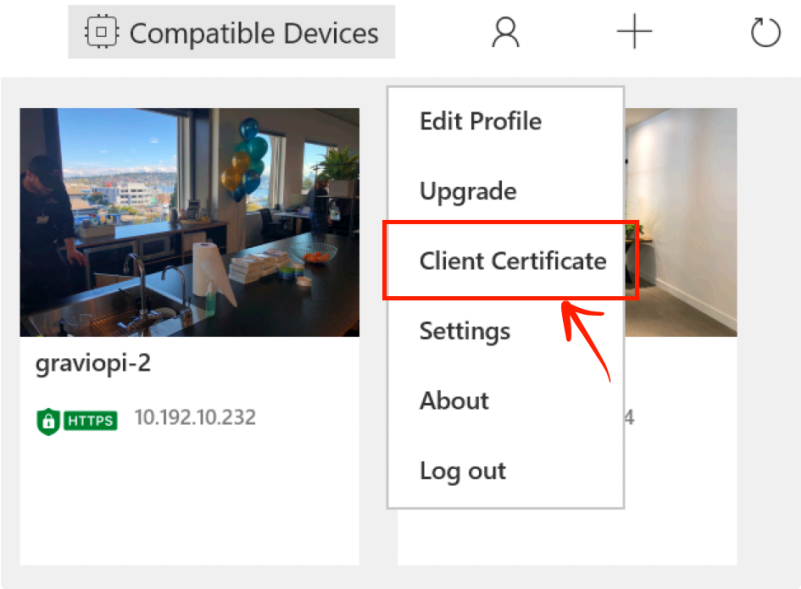
Also if you like your harddisk not to be filled up, you can set the amount of days the pictures should be stored under “Settings” then “Disk Management”.

You can find more about how to setup a ONVIF Camera by watching this video:
https://www.youtube.com/watch?v=8dkSIz_5810

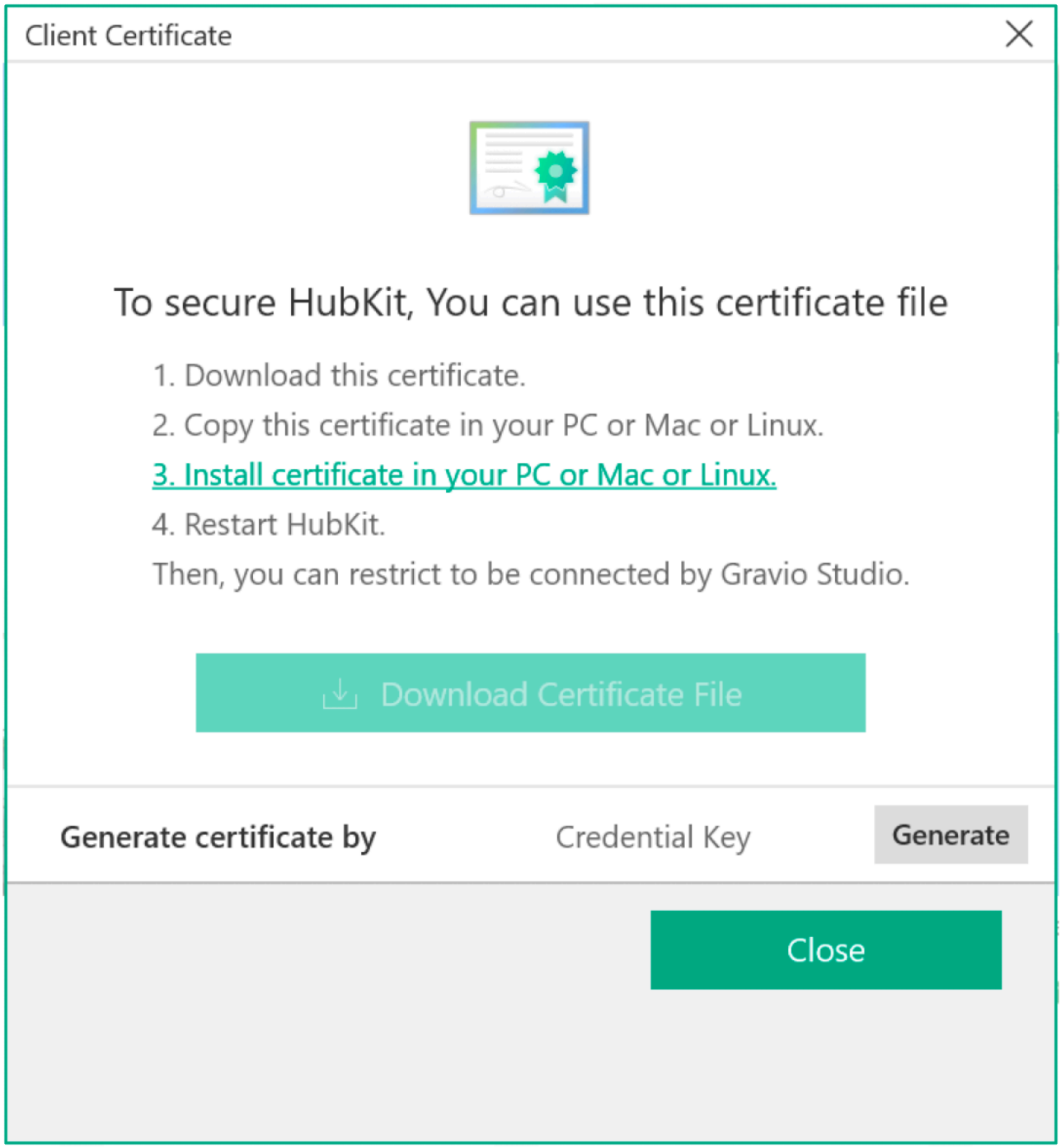
13.4. Restricting access to the Gravio Hub from Gravio Studio

To restrict Gravio Studio access to the Gravio Hub, install the Gravio Studio client certificate used for the that particular Gravio Hub.
Note: This feature is available only in Gravio Basic, Gravio Standard and Gravio Enterprise subscriptions.

1. Log in to Gravio Cloud with Gravio Studio and select “Client Certificate” from the menu:



2. Enter a *Credential Key* (= password) at the bottom and press the “Generate” button to create a certificate.



3. Select “Download Certificate File” to download the certificate to your PC.

For MacOS

Put the downloaded certificate in the following directory.
/Library/Application Support/HubKit/clientcert

Open the “Settings…” menu in the Gravio HubKit Toolbar application, and select the “Connection” tab and check the “Use client authentication” checkbox.
After that, start the HubKit.

For Gravio Hub / Linux

Login via SSH, and put the downloaded certificate in the following directory.

/home/gravio/hubkitrepo4/etc/clientcert

Create “config.json” file which contains the following information in “/home/gravio/hubkitrepo4/etc/ctrlmgr/conf” directory

```
{
  "Client" : {
    "UseClientCertificates" : true
  }
}
```

Then Restart Gravio Hub

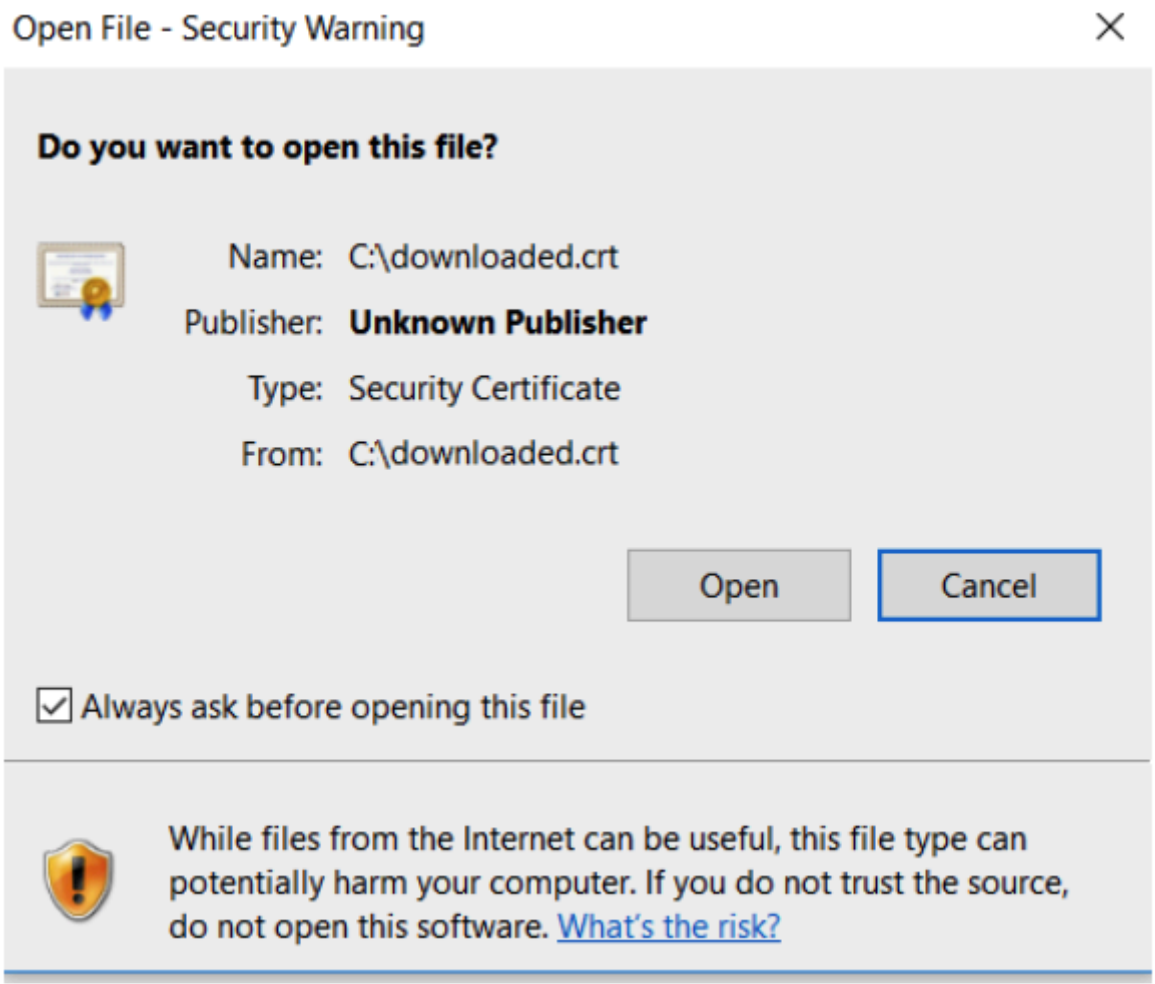
After registering the certificate to the Hub, restart the Gravio HubKit. Restarting will enable client authentication.

13.5. How to install Gravio server Root CA certificates.

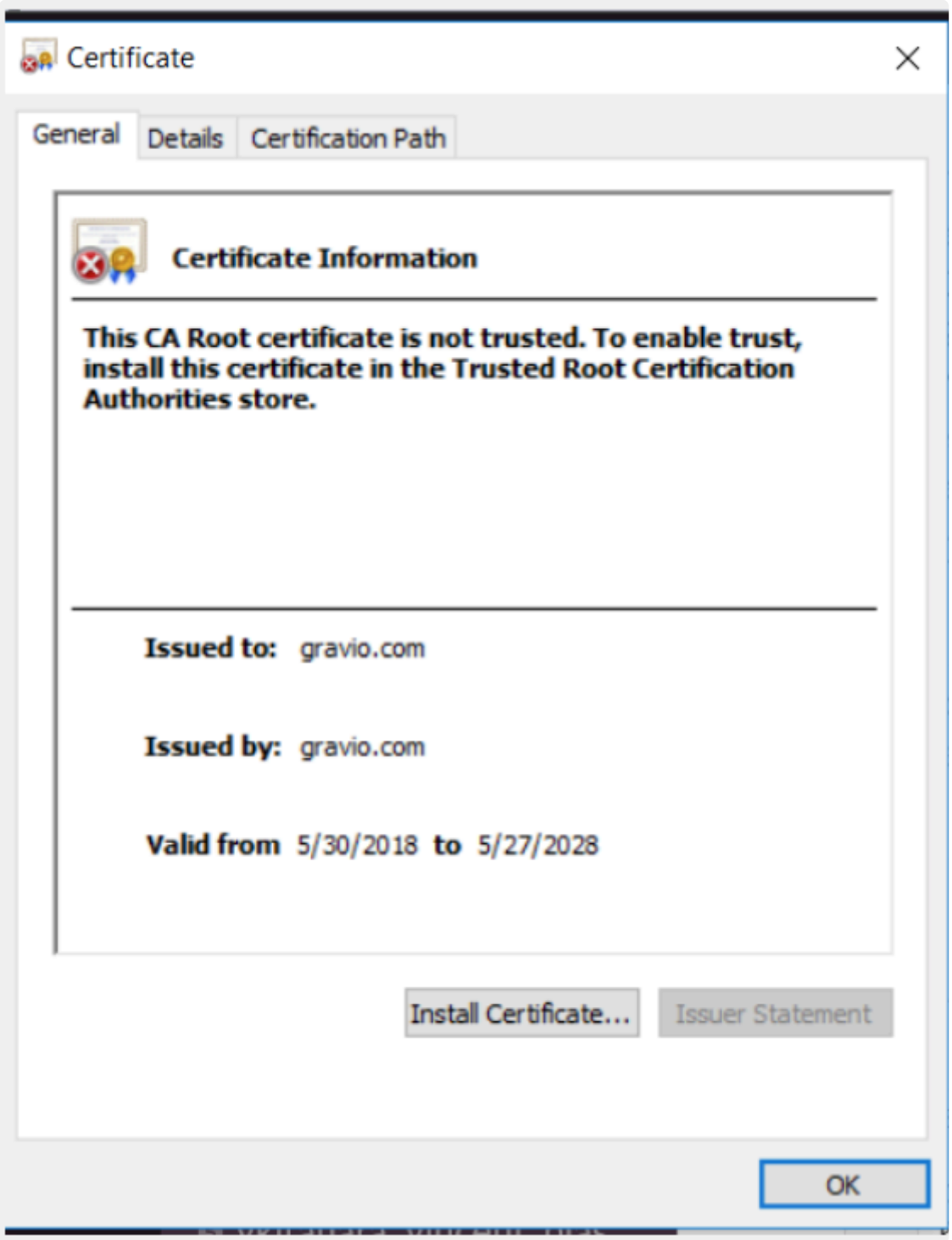
In order to make the secure connection, you will need to install the downloaded certificates to your operating system. HubKit can then verify that you are allowed to access it.

Installing the Certificate on Windows 10

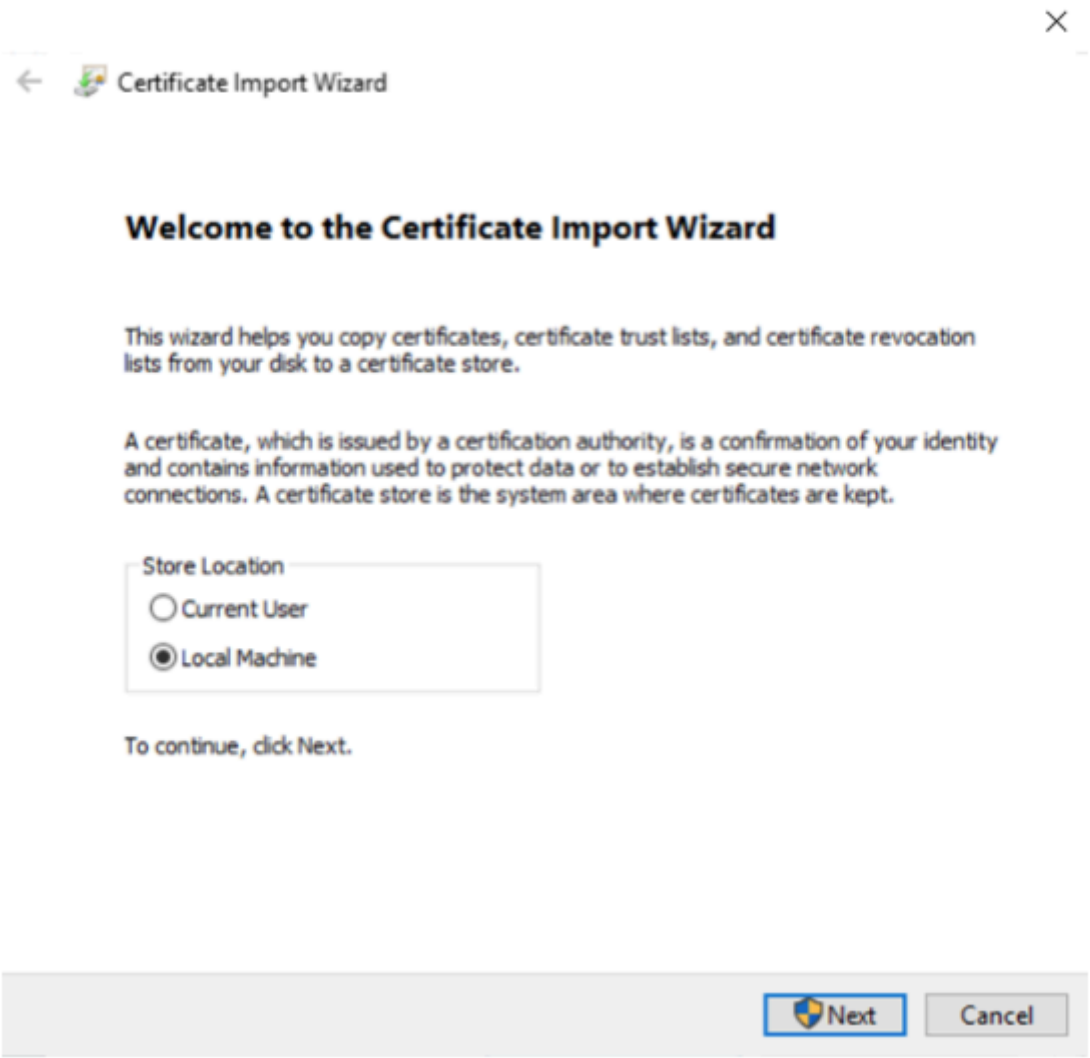
Double click on the certificate file from Explorer



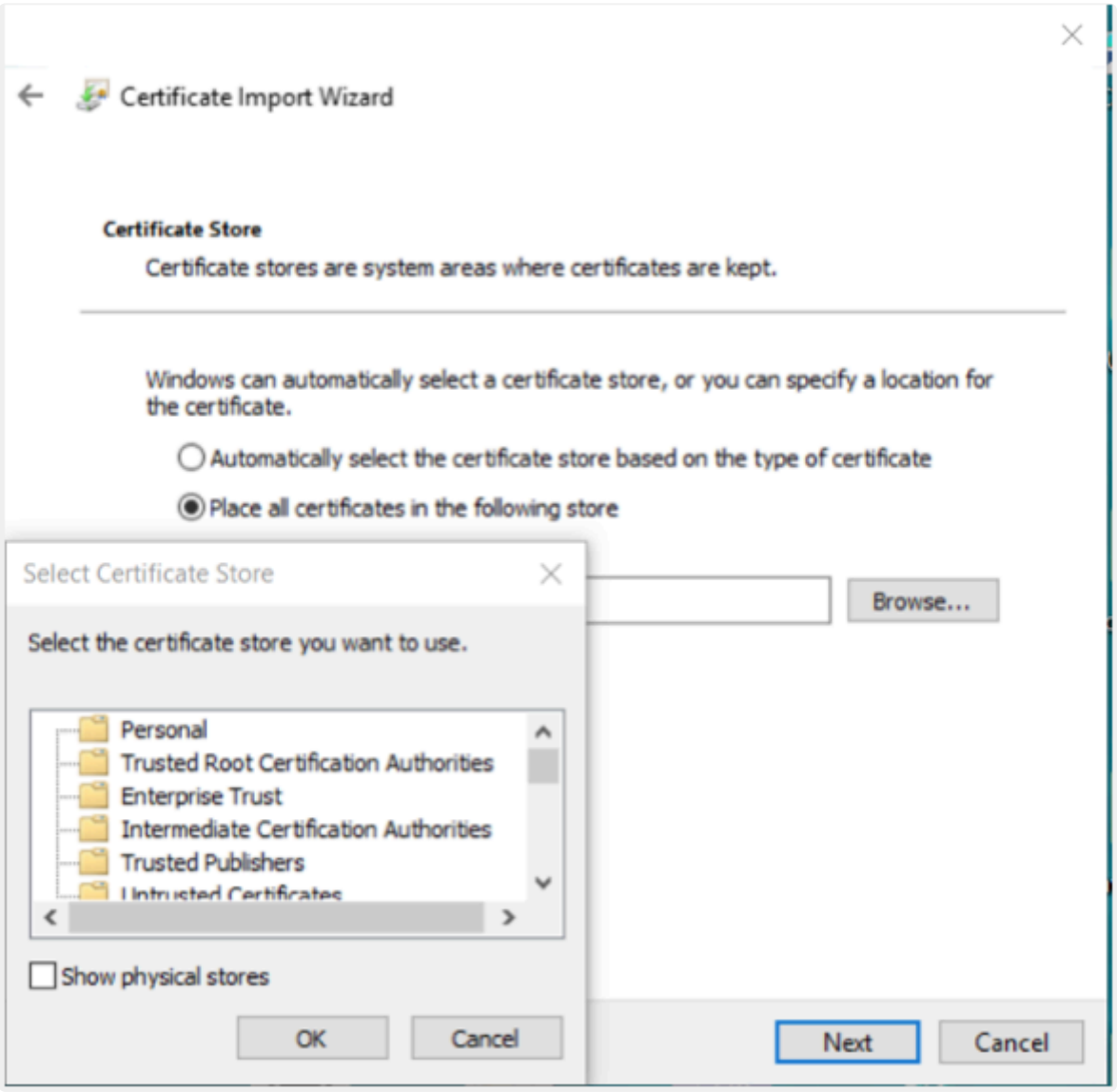
On the certificate window, click on “Install Certificate”



On the Import Wizard, select “Local Machine” and press Next

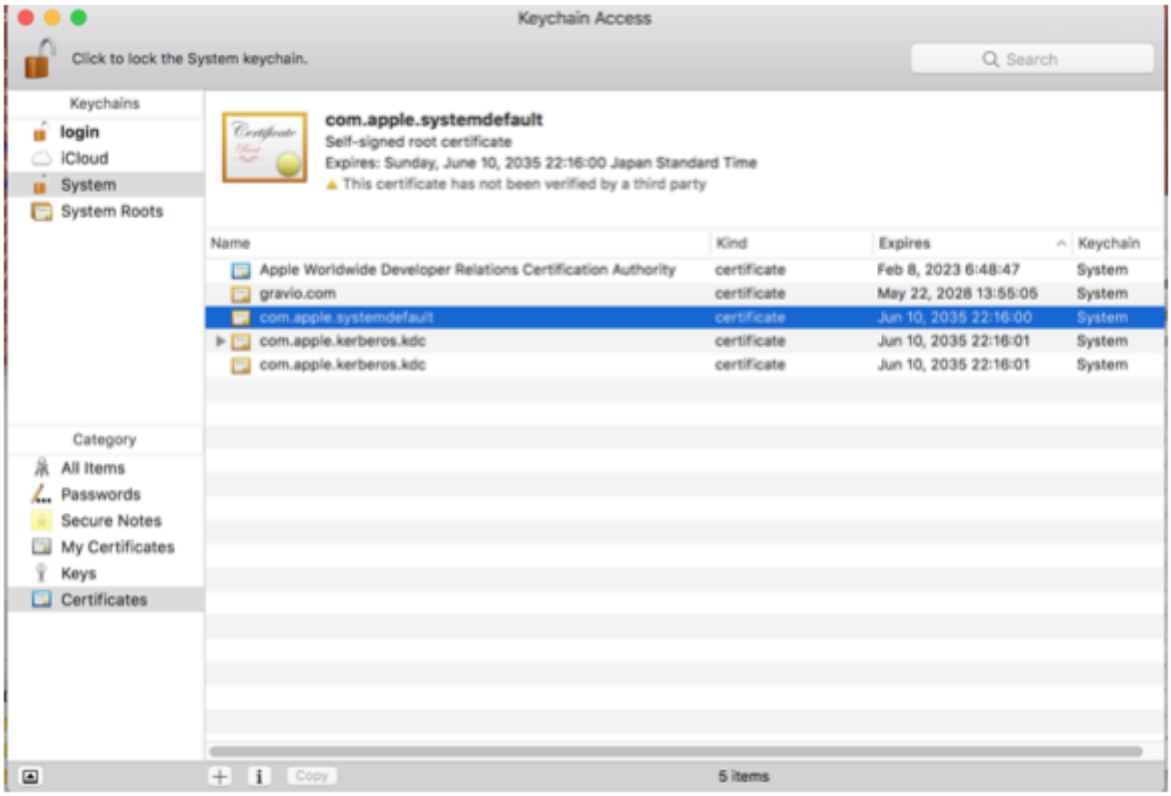


On the screen that follows, select the “Place all certificates in the following store” radio button, and then select “Trusted Root Certification Authorities”. Press Next and then Finish.

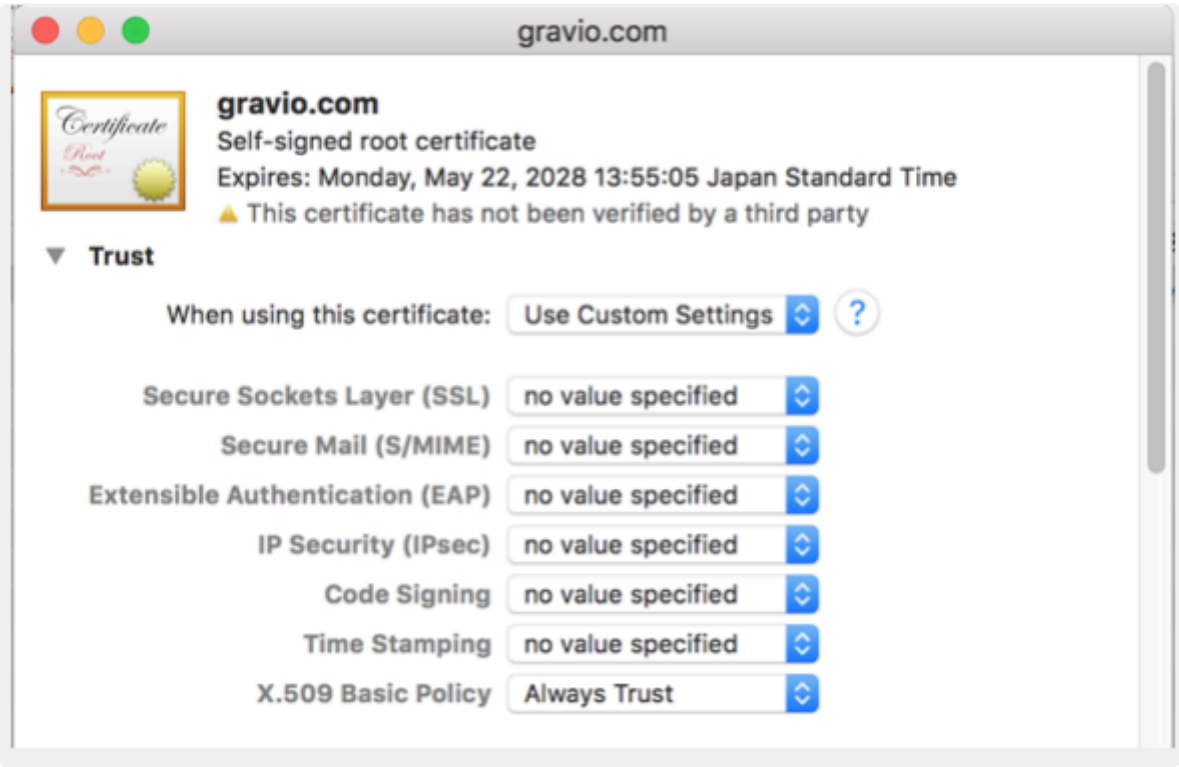


Installing the Certificate on Mac OS X High Sierra

Open “Keychain Access” using Spotlight Search. Select the “System” Keychain category, and drag and drop the key from Finder or the Desktop. You will be prompted for your password.



Right click on the installed certificate (gravio.com), and select “get info”. Click on the down arrow next to “Trust” and on the dropdown next to X.509 Basic Policy, select “Always Trust”. Close the Window and you will be prompted for your password. The certificate is installed.



Installing the Certificate on Linux (Ubuntu)

1. Install the ca-certificates program using “@sudo apt-get install -y ca-certificates@”

2. Move the Gravio certificate file to the folder “@/usr/local/share/ca-certificates@”
3. Update certificates by using “@sudo update-ca-certificates@”. (Note: The Gravio services may need to be restarted to be able to verify the certificate).

13.6. How to Side-Load Gravio Studio for Windows

Sideloading procedures for Gravio Studio for Windows

Sideloading is the process of installing an application from an installer file rather than from the App Store. In some cases, for example in a corporate environment, the App store may not be available. Or a computer may not be connected to the internet. In this case, you can install Gravio Studio via side-loading. Please note, that side-loading does not support in-app purchases.

1. Download the and extract Sideloading Pack

Download the ZIP file from the Gravio.com website

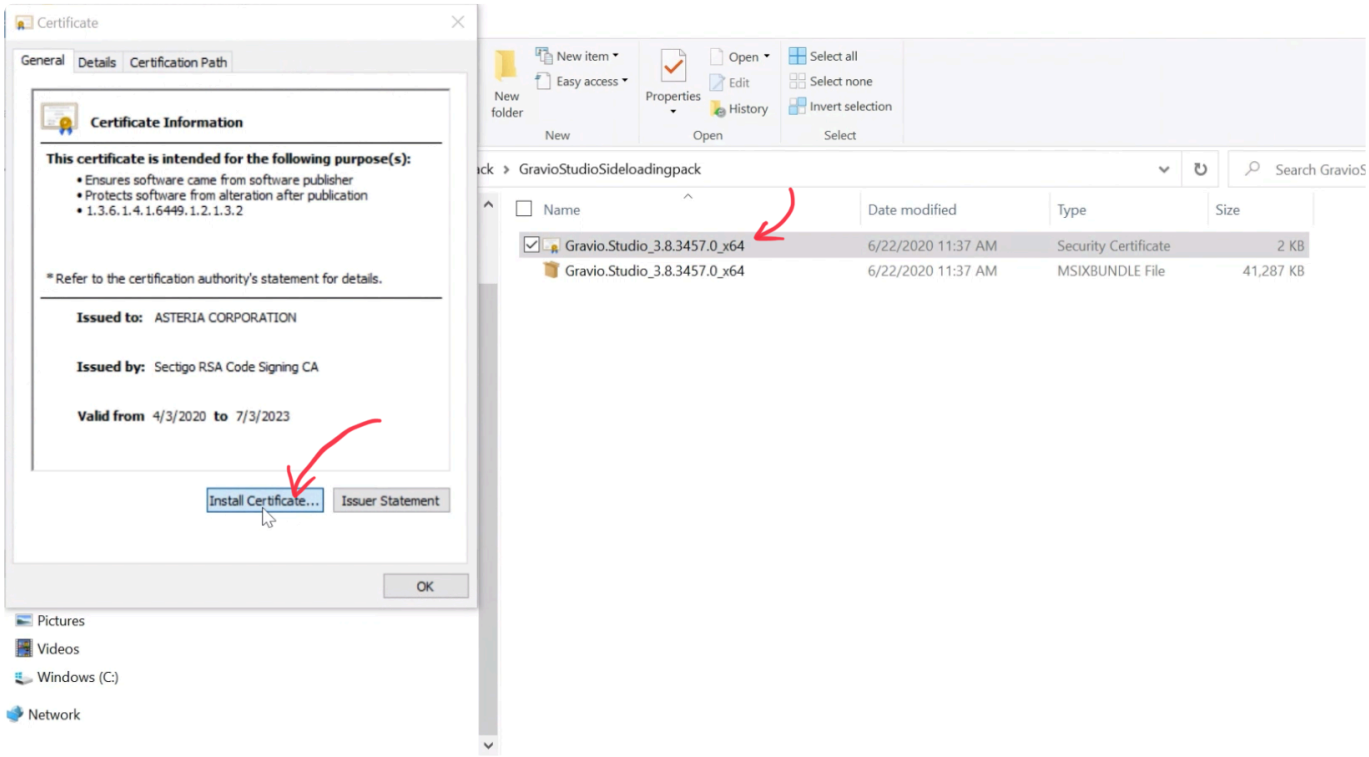
Ensure that you always download from the Gravio.com domain.

2. Installing the Certificate

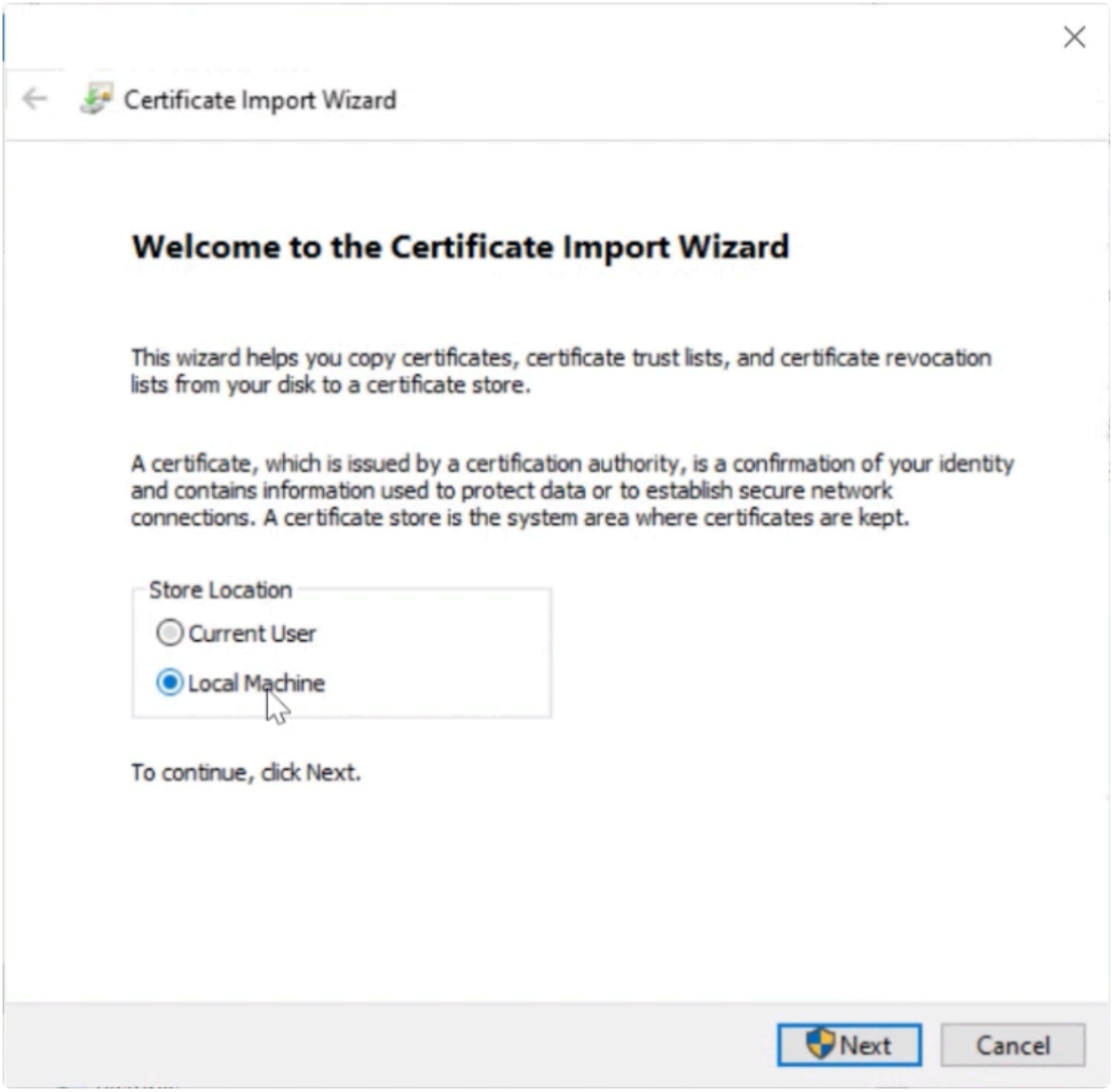
The pack contains 2 files: the installer file and the certificate.

Name	Date modified	Type	Size
Gravio.Studio_3.8.3457.0_x64	6/22/2020 11:37 AM	Security Certificate	2 KB
Gravio.Studio_3.8.3457.0_x64	6/22/2020 11:37 AM	MSIXBUNDLE File	41,287 KB

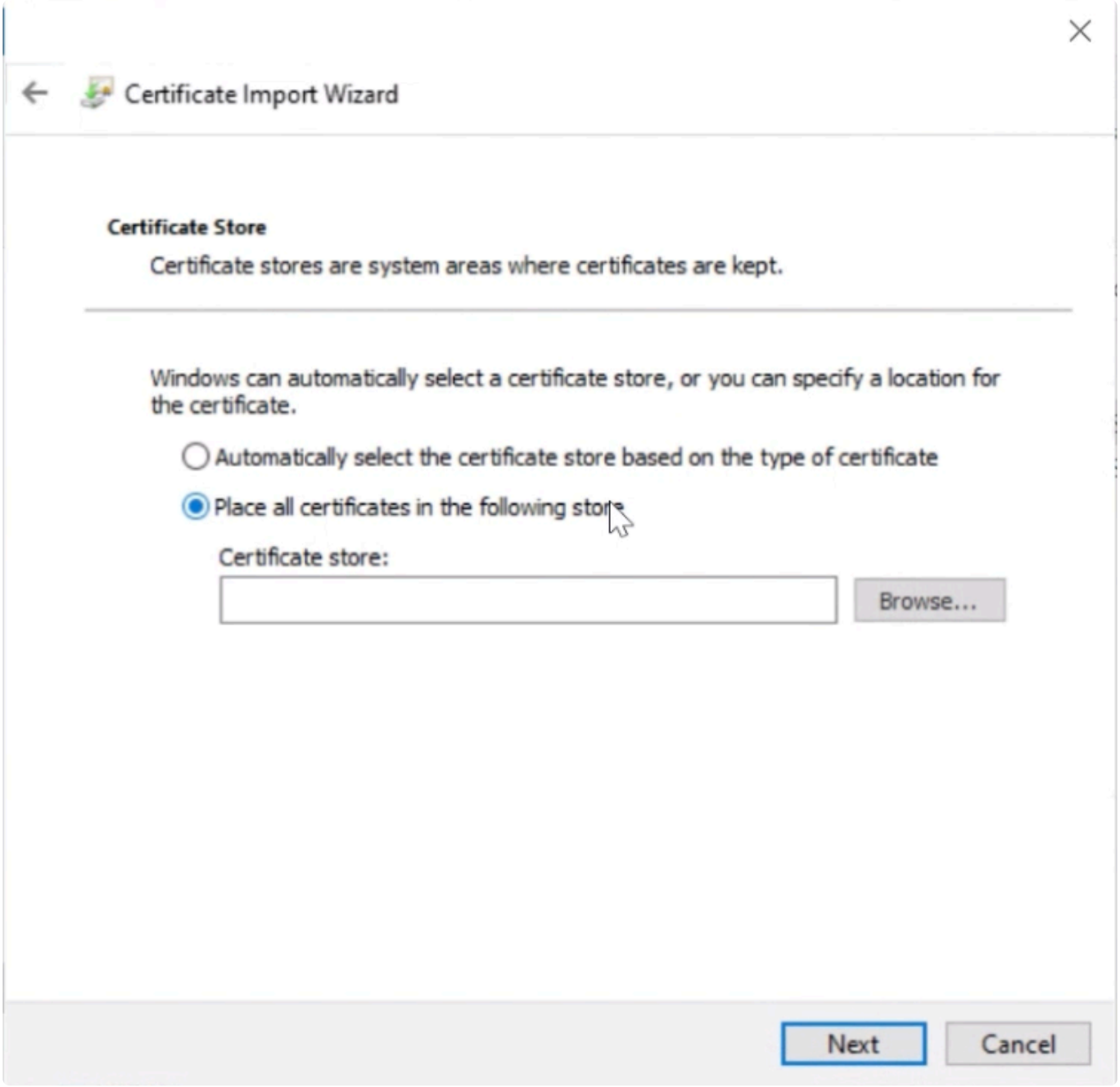
Click on the certificate first to launch the certificate installation process.



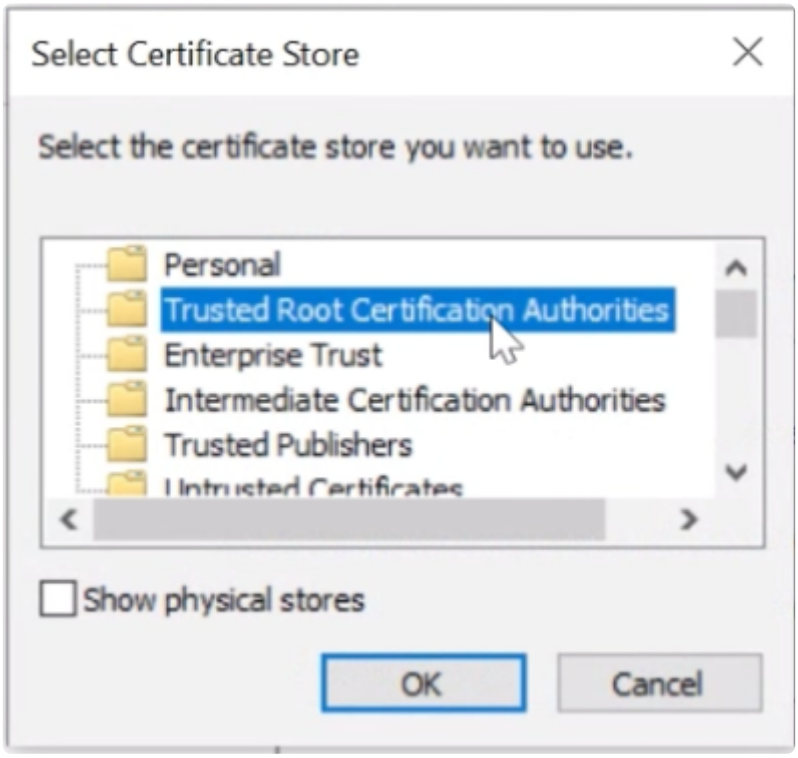
and install on your local machine:



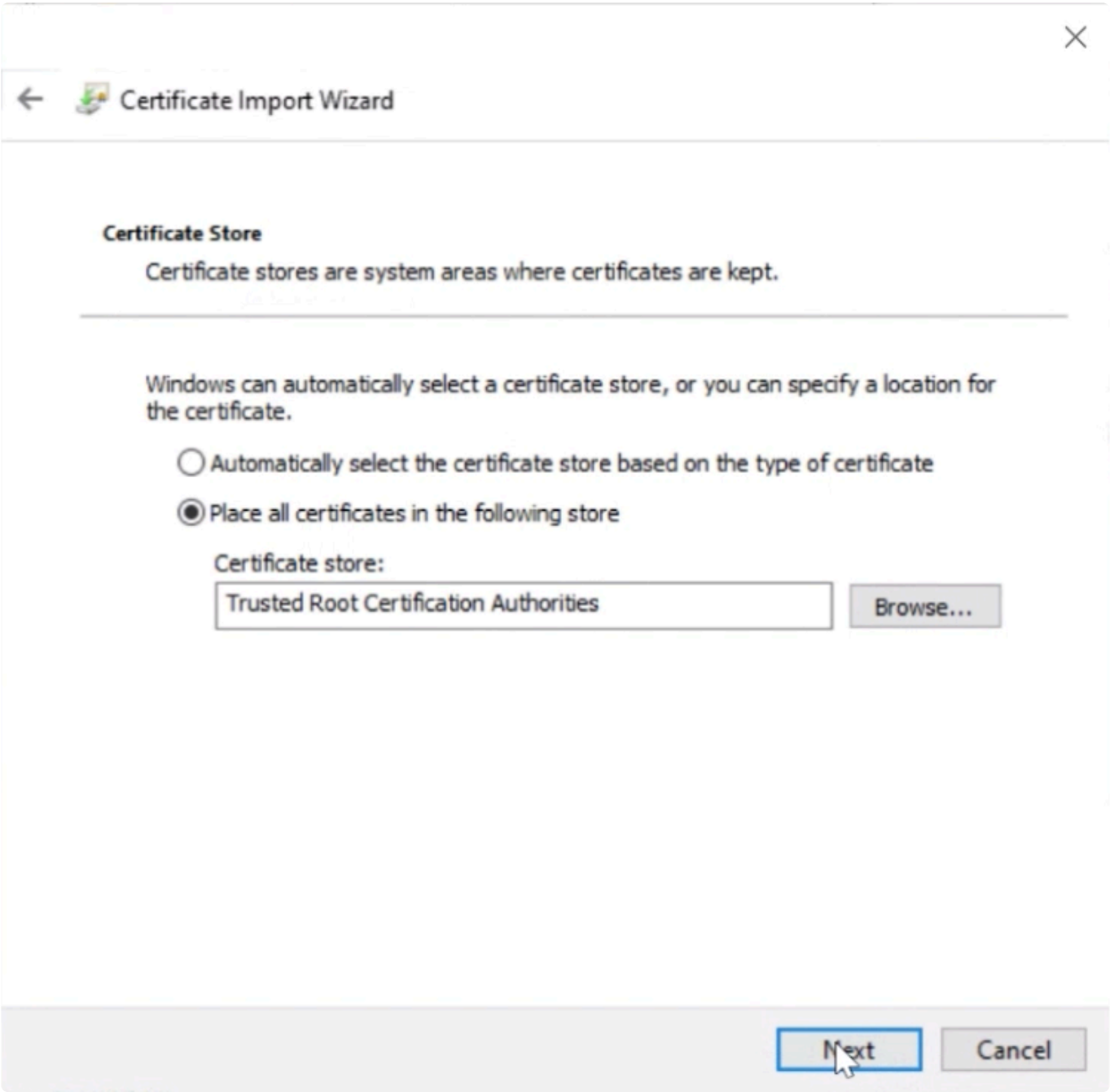
After clicking “Next” you will see a confirmation screen, click “Yes” to confirm that you want to go ahead with the installation, then you will see the Certificate Import Wizard. Here please select to manually choose the store:



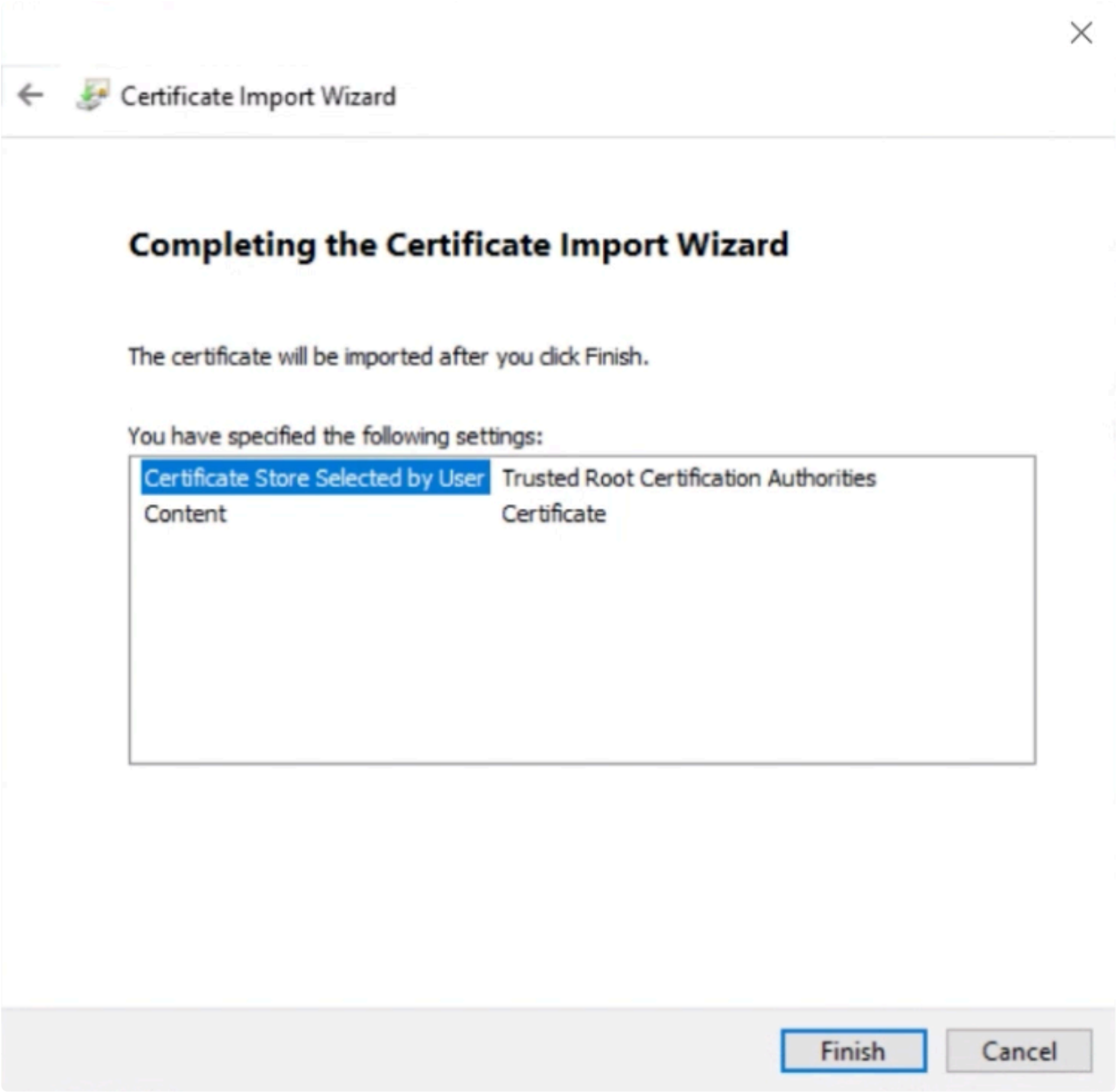
And click on the “Browse” button. In the popup that opens, please select the “Trusted Root Certification Authorities” option:



After this, click the “Next” button to proceed:



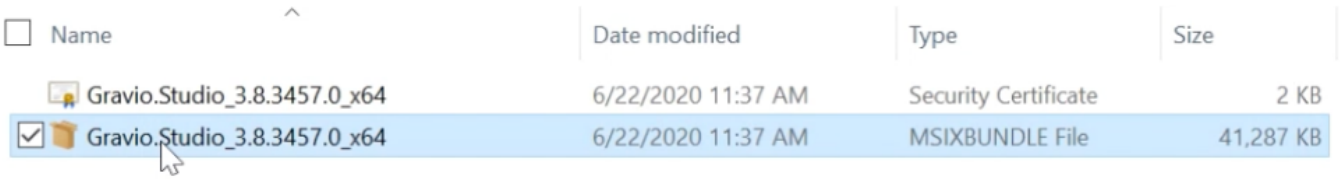
On the next screen, confirm the options you have chosen and click on “Finish”:



This concludes the installation of the certificate.

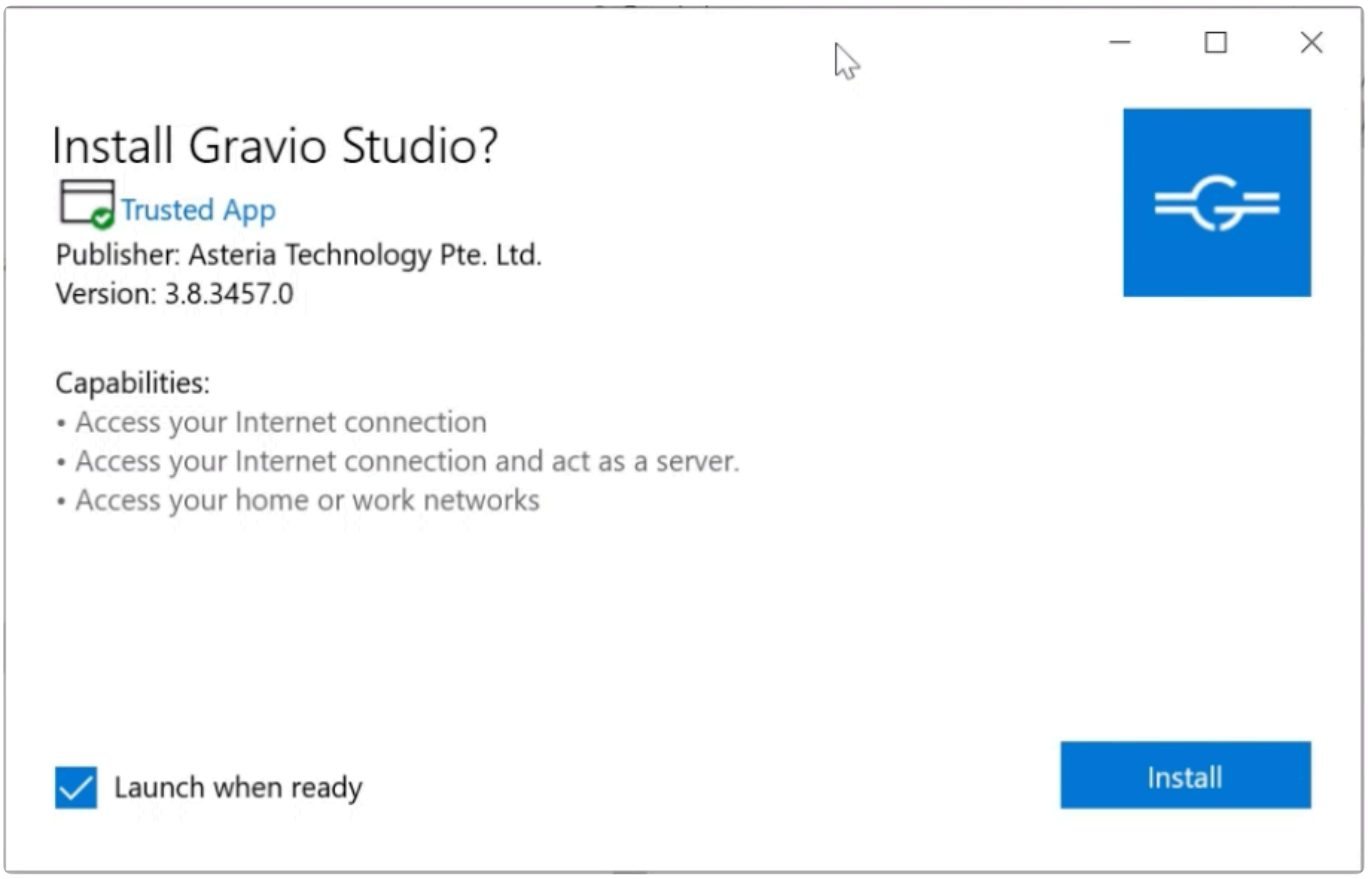
3. Installing the Application

Now, please ensure that you have Windows 10 version 1904 or later, and then double click on the installation package to launch the installation:



The following confirmation window will open. Click on the “Install” button and follow the steps install

Gravio Studio:



If you encounter any problems, please join our Slack Channel and ask for help: Gravio World Wide Community

13.7. Zigbee Firmware Update

dongle is shown below.

Download the firmware from Here

1. When using a Mac

Prepare a Mac with the Gravio HubKit4 installed.

Plug the Zigbee dongle into the USB port of your Mac and launch the terminal.
Insert the dongle with the Gravio logo on the top.

In Terminal app.

```
$ ls -lsa /dev/cu.usbserial-*
```

to see the device name of your Zigbee dongle. In the example, it would be /dev/cu.usbserial-DO00HPT3。

```
sudo /Applications/Gravio HubKit 4.app/Contents/Resources/gssosx/zigbeefirmwareupdate -f <path>/V0.9.2.bin -d /dev/cu.usbserial-DO00HPT3
```

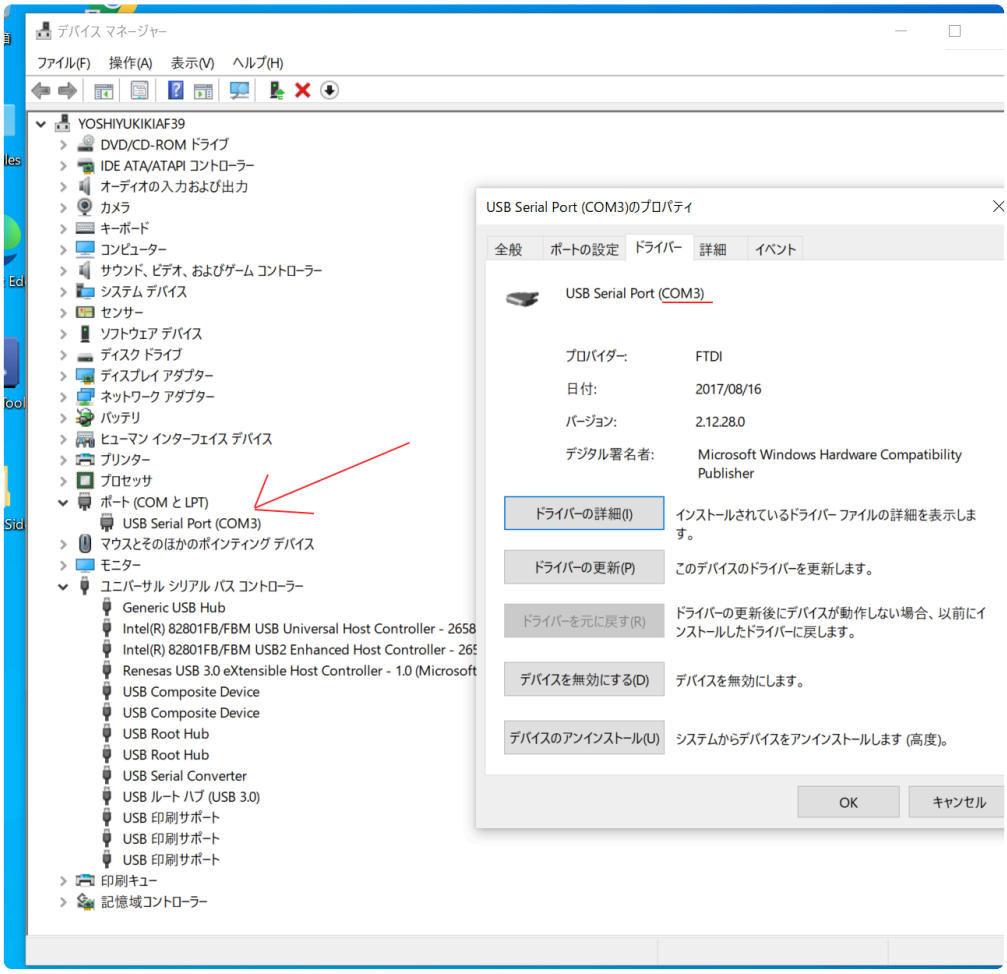
DO00HPT3 should be replaced with the actual device name you are using.

Please use the firmware update once the update is finished. Also, the sensors already paired together can be used as they are (the pairing will be maintained).

2. When using Windows

Prepare a Windows environment in which Gravio HubKit4 is installed.

Plug the Zigbee dongle into the Windows USB port and launch PowerShell.
Insert the dongle with the Gravio logo on top.



Launch Device Manager and see the COM port (COM and LPT) of the Zigbee dongle. In the example, let's say COM3.

In Powershell or CMD.exe

```
cd "c:¥Program Files¥GravioHubKit4"
```

and change the current directory by typing

```
.¥ZigbeeFirmwareUpdate.exe -f <path>/V0.9.2.bin -d COM3
```

Replace COM3 with the actual COM port you are using.

The firmware will be updated when the update is finished and you can use it. Also, the sensors you have already paired can be used as they are (the pairing will be maintained).

13.8. How to set up the Panasonic i-PRO Camera for Motion Detection, Congestion and Alarms

The camera used is the WV-S1135V for i-PRO camera settings.

Access the WV-S1135V camera's web page to open the settings. Please refer to the i-PRO manual for instructions on how to use the WV-S1135V and how to operate the web page.

AI Motion Detection

To use AI motion detection, you need to install AI-VMD (WV-XAE200WUX) on the camera.
Please use version 3.2 or higher.

For AI motion detection settings that detect AI-recognized actions, count function settings, detailed settings, etc., please refer to the i-PRO manual for how to set up on the i-PRO side.
To integrate with Gravio, open the count function settings screen.

Please register the URL of Gravio's i-PRO settings screen for HTTP transmission.

AI Congestion Detection

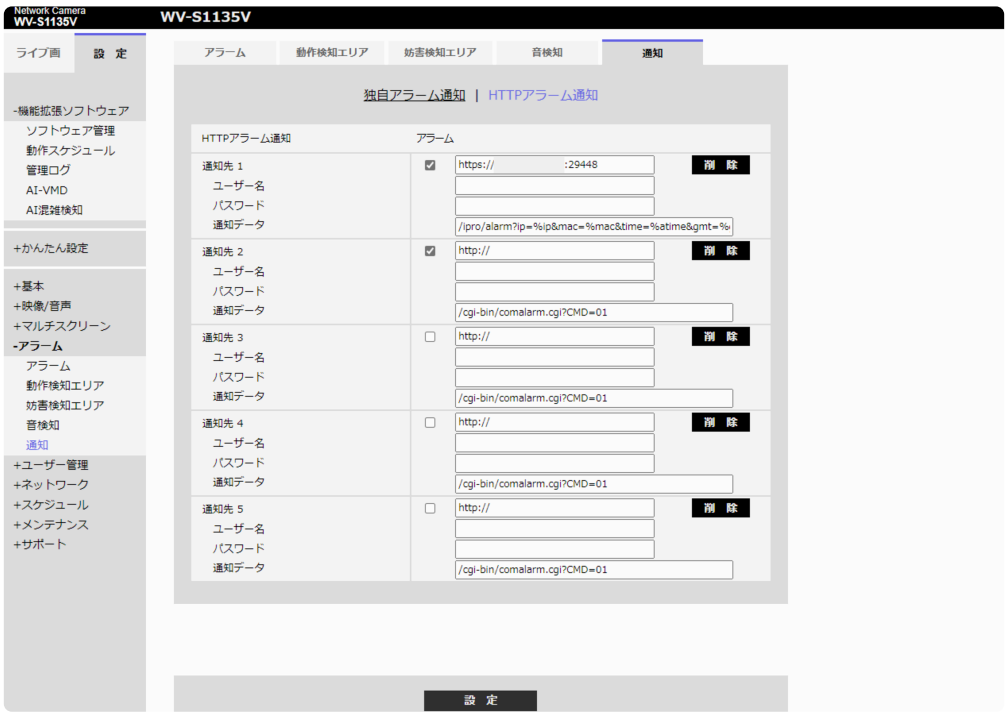
To use AI congestion detection, you need to install WV-XAE207W on the camera.
Please use version 1.6 or higher.

For basic settings that detect AI-recognized actions, MQTT transmission settings, detailed settings, etc., please refer to the i-PRO manual for how to set up on the i-PRO side.
To integrate with Gravio, open the HTTP transmission screen.

Please register the URL of Gravio's i-PRO settings screen as the transmission destination.

Alarms

Alarms are set up from the menu to set the motion detection area for the alarm and open notifications.
Please register the URL of Gravio's i-PRO settings screen for HTTP alarm notifications.


Network Camera
WV-S1135V
WV-S1135V
ライブ画
設 定
-機能拡張ソフトウェア
ソフトウェア管理
動作スケジュール
管理ログ
AI-VMD
AI混雑検知
+かんたん設定
+基本
+映像/音声
+マルチスクリーン
-アラーム
アラーム
動作検知エリア
妨害検知エリア
音検知
通知
+ユーザー管理
+ネットワーク
+スケジュール
+メンテナンス
+サポート
アラーム
動作検知エリア
妨害検知エリア
音検知
通知
独自アラーム通知 | HTTPアラーム通知
HTTPアラーム通知
アラーム
通知先 1
https:// :29448
削 除
ユーザー名
パスワード
通知データ
/ipro/alarm?ip=%ip&mac=%mac&time=%atime&gmt=%
通知先 2
http://
削 除
ユーザー名
パスワード
通知データ
/cgi-bin/comalarm.cgi?CMD=01
通知先 3
http://
削 除
ユーザー名
パスワード
通知データ
/cgi-bin/comalarm.cgi?CMD=01
通知先 4
http://
削 除
ユーザー名
パスワード
通知データ
/cgi-bin/comalarm.cgi?CMD=01
通知先 5
http://
削 除
ユーザー名
パスワード
通知データ
/cgi-bin/comalarm.cgi?CMD=01
設 定

13.9. How to obtain an API Key for OpenAI

To use the OpenAICompletion, OpenAIChat, OpenAIImage, OpenAIAudio components, please create an account with OpenAI and obtain an API Key.

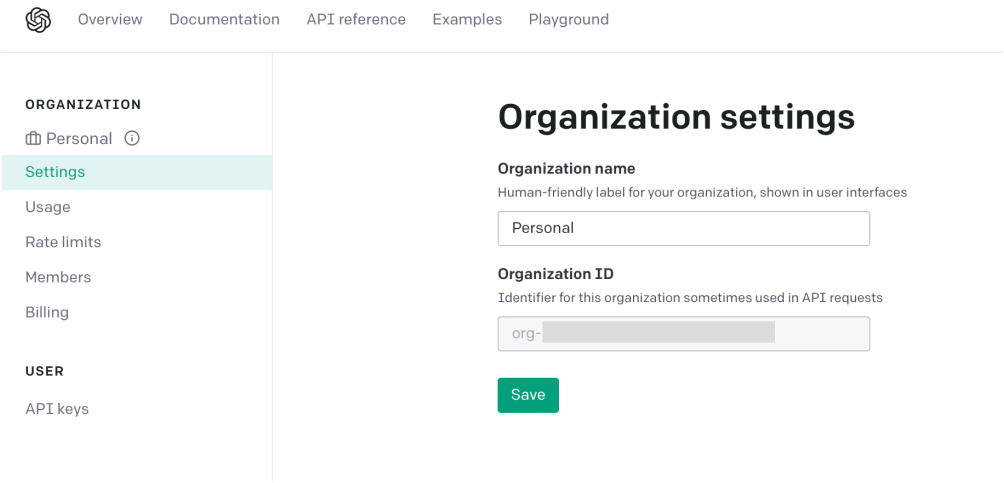
1. Creating an account

Sign up from the “Sign Up” at the top right of the here OpenAI’s site.

2. Confirming the Organization Id

Log in with the account you signed up for on the OpenAI’s site.

Open the Manage Account menu of your account.



When you select Setting from the left menu, the Organization ID will be displayed on the Organization settings screen.

You will use it in the Organization Id property of each component.To use the OpenAICompletion, OpenAIChat, OpenAIImage, and OpenAIAudio components, you need to create an account with OpenAI and obtain an API Key.

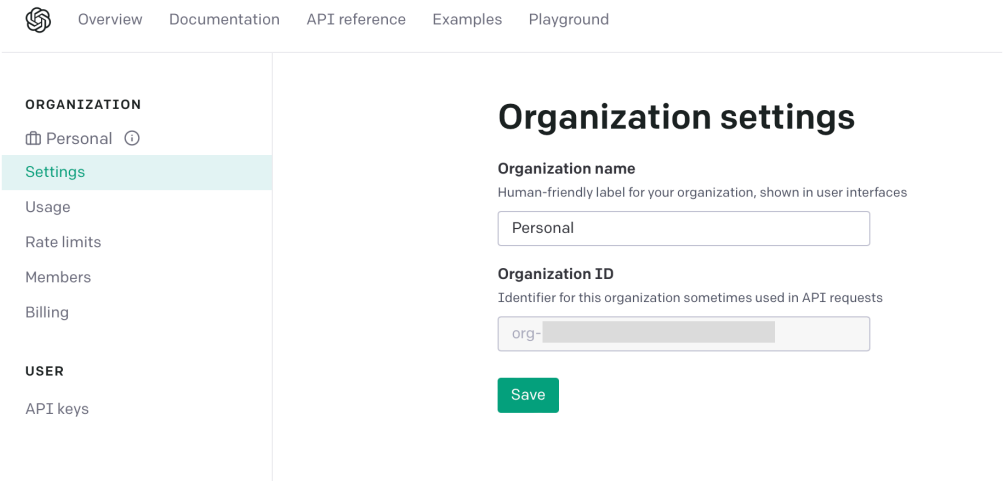
1. Creating an account

Sign up from the “Sign Up” at the top right of the OpenAI’s site here.

2. Checking the Organization Id

Log in with the account you signed up for on the OpenAI site.

Open the “Manage Account” menu of your account.

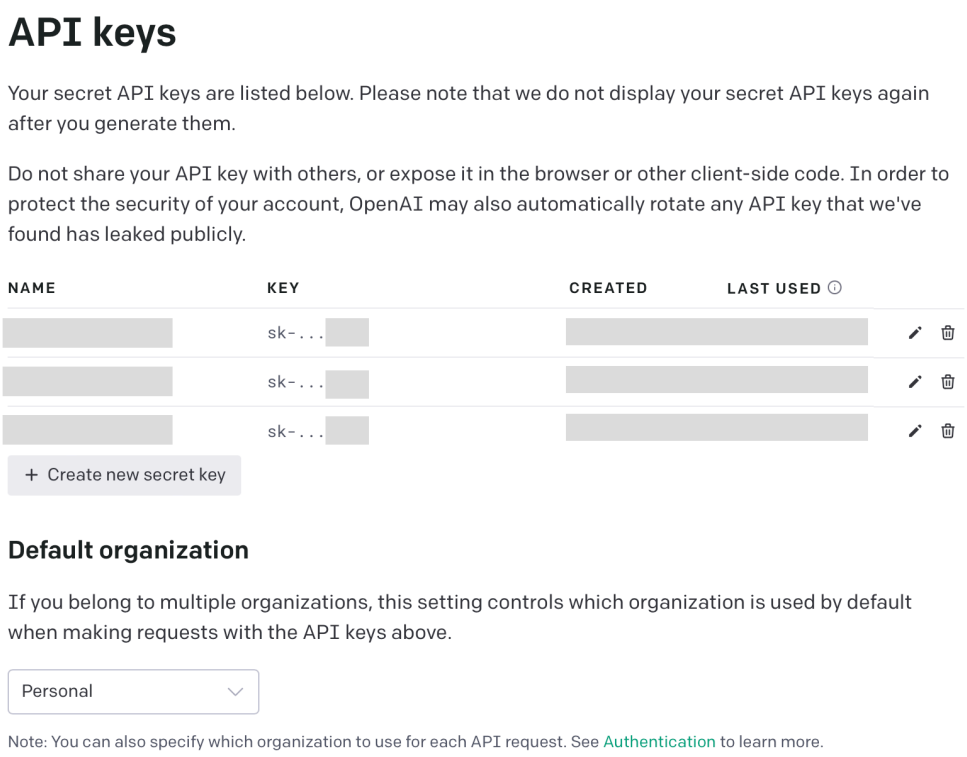


When you select “Settings” from the left menu, the “Organization ID” will be displayed on the “Organization settings” screen.

This will be used in the “Organization Id” property of each component.

3. Obtaining the API Key

When you select “API keys” from the left menu, the “API keys” screen will be displayed.



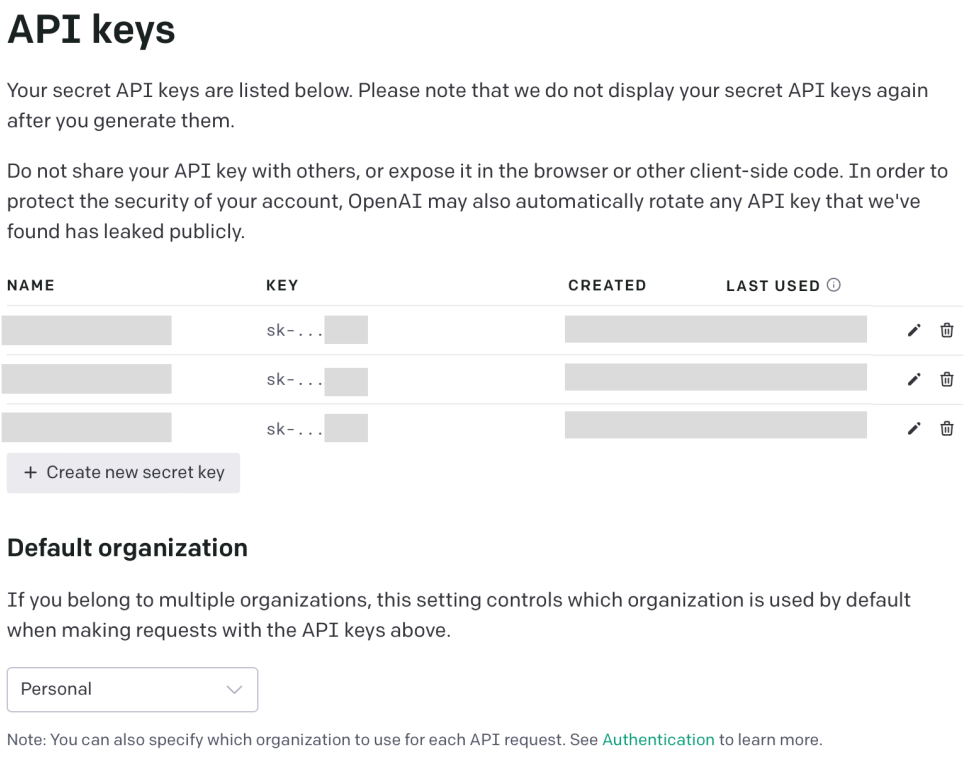
Press the “Create new secret key” button and create it by specifying the “Name”.

The “KEY” will be used in the “API Key” property of each component.

You have now obtained the API Keys for OpenAI.

3. Obtaining the API Key

When you select API keys from the left menu, the API keys screen will be displayed.



Press the Create new secret key button and specify a Name to create it.

You will use the KEY in the API Key property of each component.

This is how you can obtain OpenAI's API Keys.

13.10. How to use Text2SQL to create SQL from Prompts

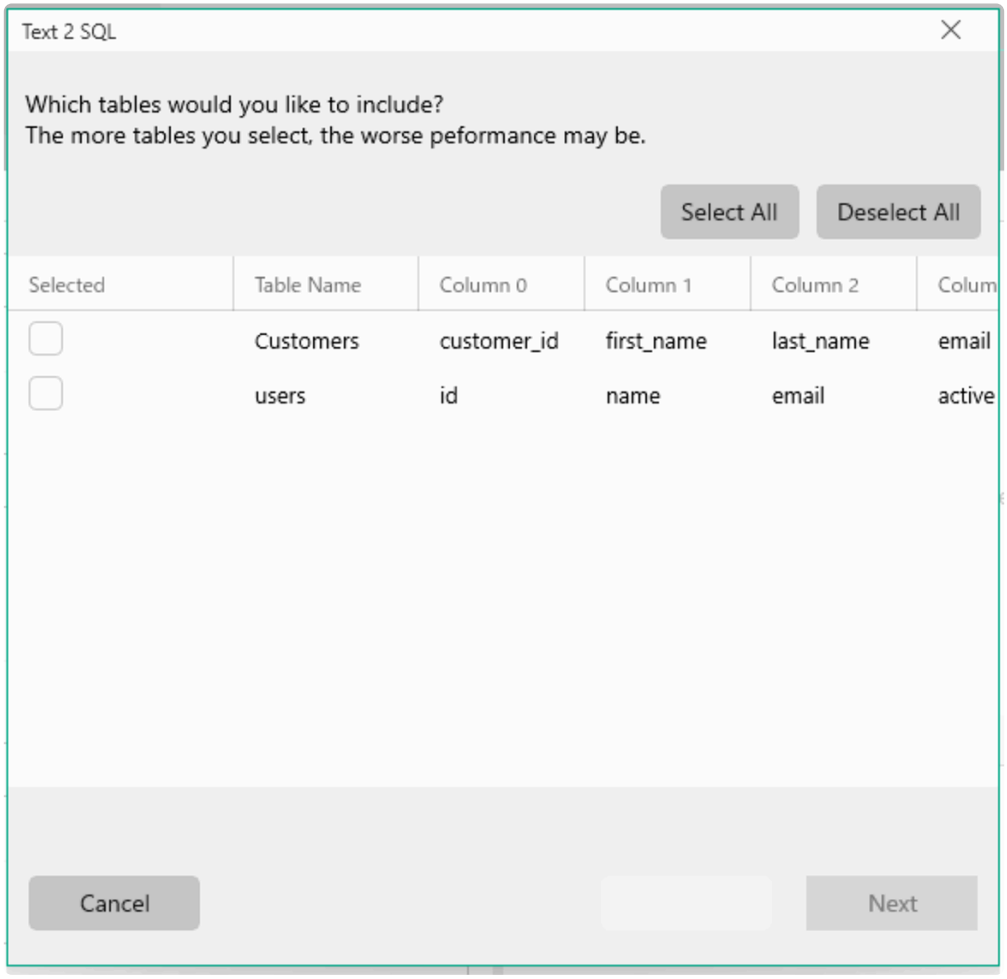
The TextSQL function provided in the database-related components is a feature that can generate SQL statements from Japanese input into the prompt to get the data you want using the ChatGPT function of OpenAI. In order to use the TextSQL function, you need to obtain the API Key and OrganizationId from OpenAI in advance. Please check here on how to get them.

1. Preparation

Please get the API Key and OrganizationId from OpenAI. Then, set the properties in each database-related component so that you can correctly connect to the database.

2. Selecting a Table in Text2SQL

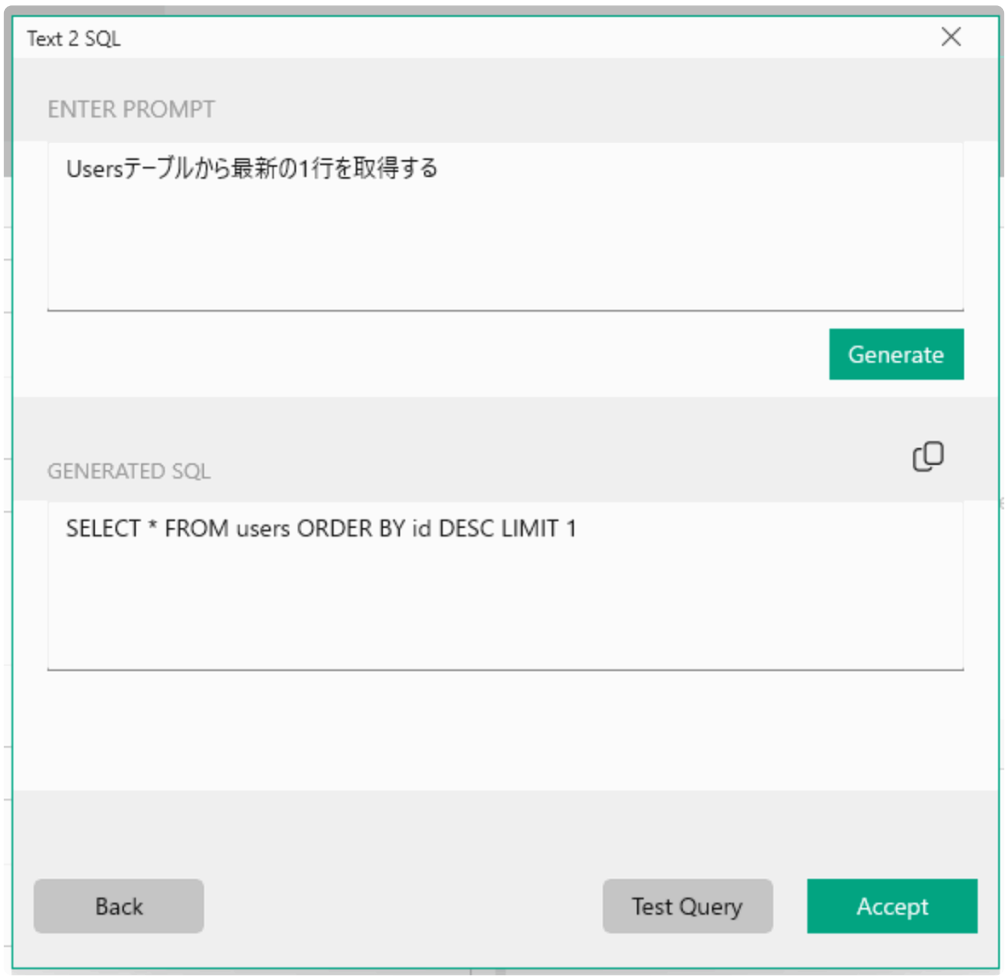
When you press the Text2SQL button, the Text2SQL panel will open.



3. Generating SQL Statements from Prompts

When the Text2SQL panel opens, if you can connect to the database correctly, the tables will be displayed in a list, so select the table you want to use.

When you press the Next button, the prompt input panel will be displayed.



Enter the data you want to get in Japanese in the prompt. When you press the Generate button, the generated SQL statement will be displayed in GENERATED SQL.

When you press the Test Query button, a panel for executing SQL and getting data will be displayed.

When you press the Accept button, the generated SQL statement will be copied to the SQL property of the component.

13.11. Setting up Google Cloud Storage

The following is an example configuration of how to send and receive files to and from Google Cloud Storage using the GCS Upload/Download component.

Configuration flow

Follow the steps below to configure the settings on the Google Cloud Platform

1. create a service account in IAM and create and download a key file.

2. connect the service account created in 1. as an authority to the bucket in Google Cloud Storage, and specify “Storage Legacy Bucket Owner” and “Storage Legacy Object Owner” as roles.

In the case of GCSDownload component, specify “Storage Legacy Bucket Read” and “Storage Legacy Object Read” as roles.

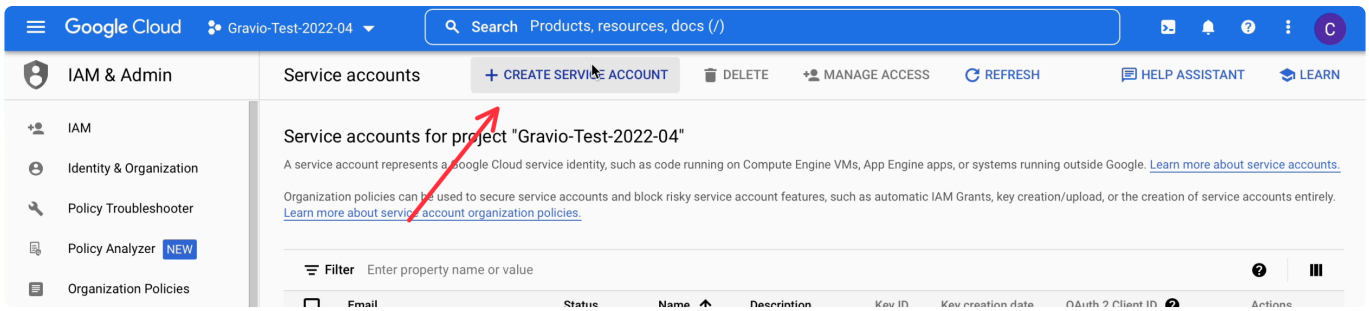
3. For the GCS Upload/Download component, specify the bucket specified in 2. and the key file specified in 1.

Advanced Preparation

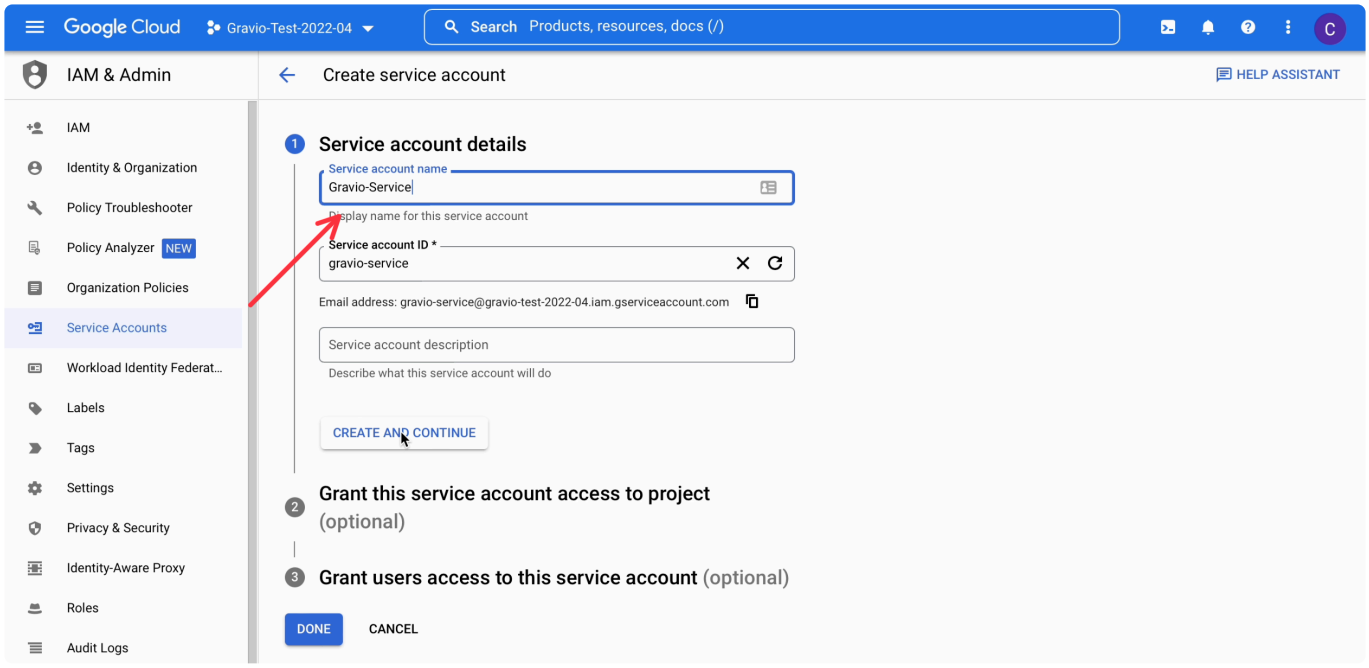
Create a bucket in Google Cloud Storage in Google Cloud.

Create a service account in the Google Cloud console

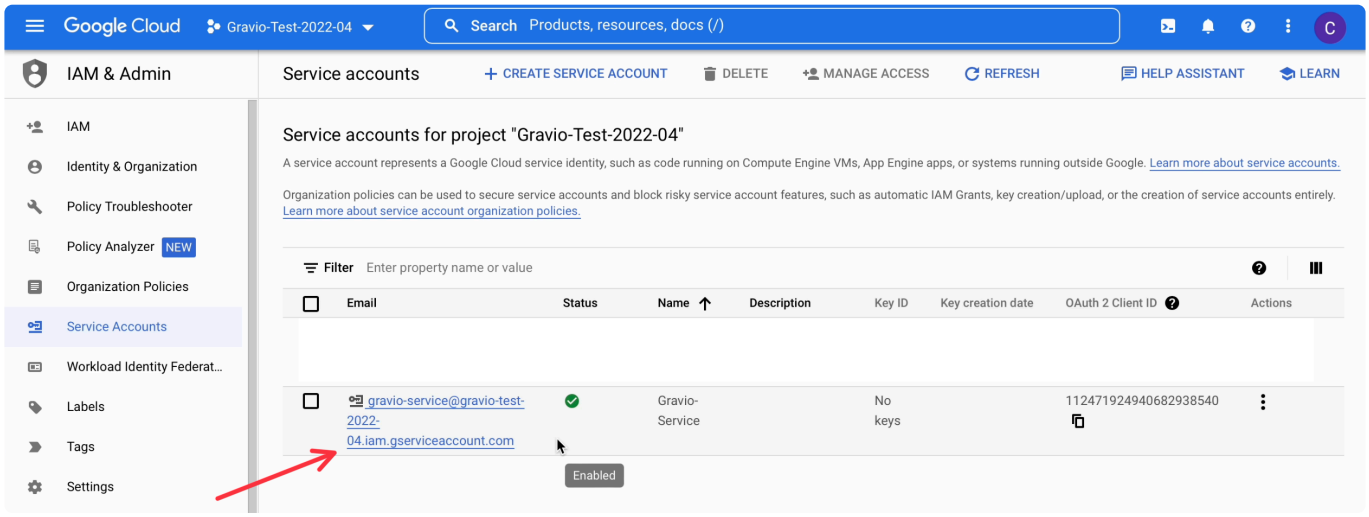
Go to IAM and Administration from the Google Cloud console and create a service account.



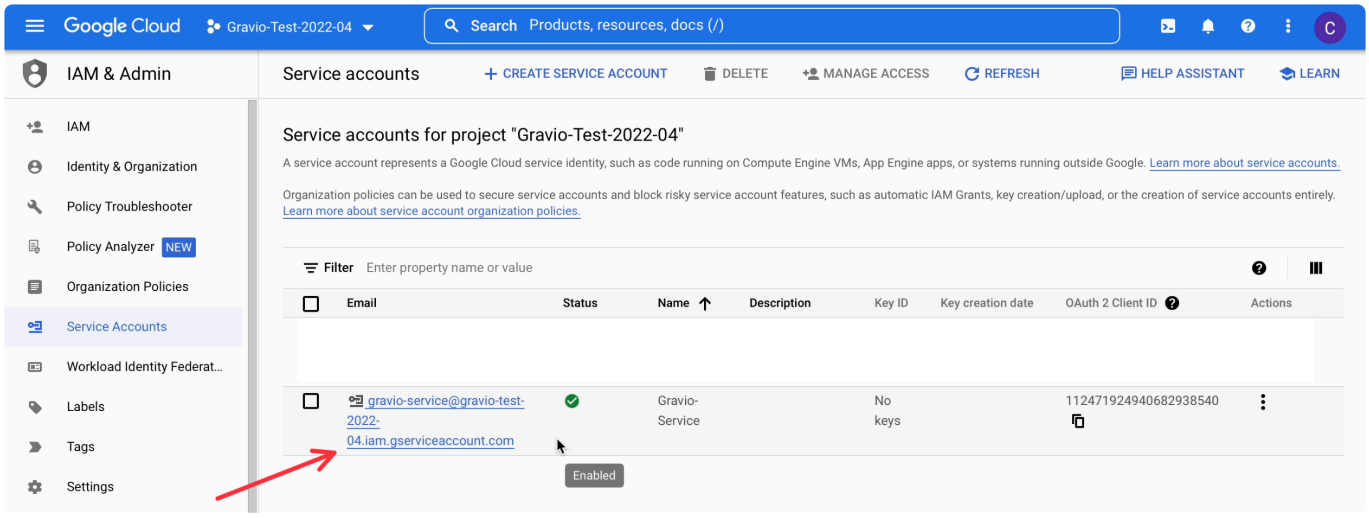
Set the Service Account Name and Service Account ID.



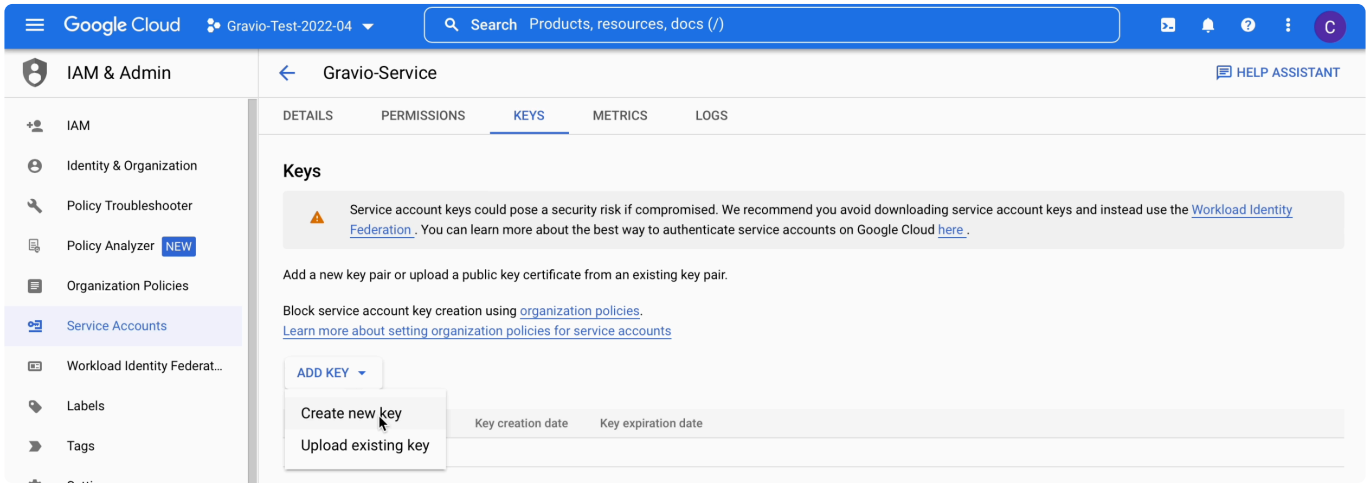
Allow this service account to access the project and “Allow users to access this service account” are optional, so do not specify anything and click “Finish” to save the settings to display them in the list.



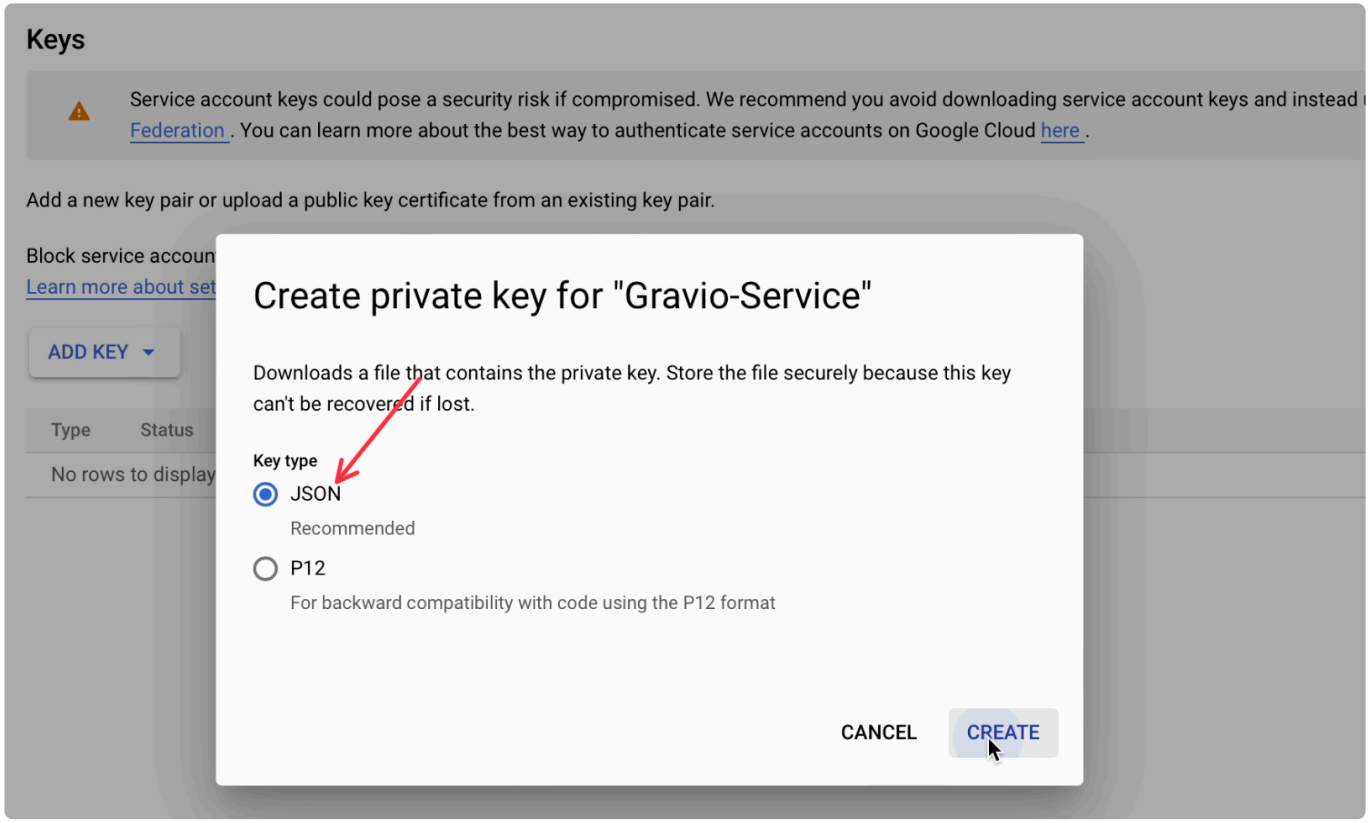
Select “Manage Keys” from the operations.



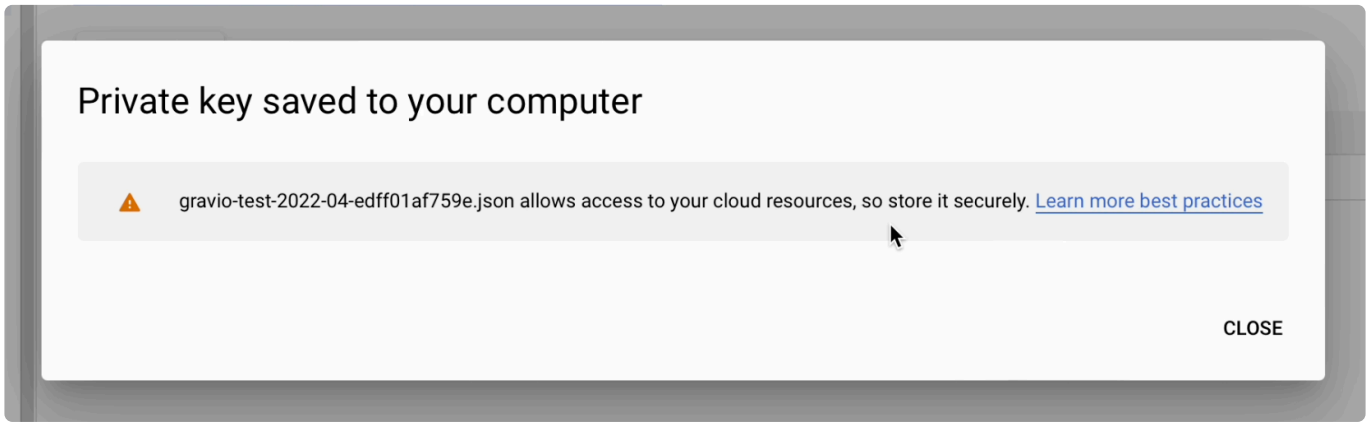
Select “Add a key”



Create the key type as “JSON”.



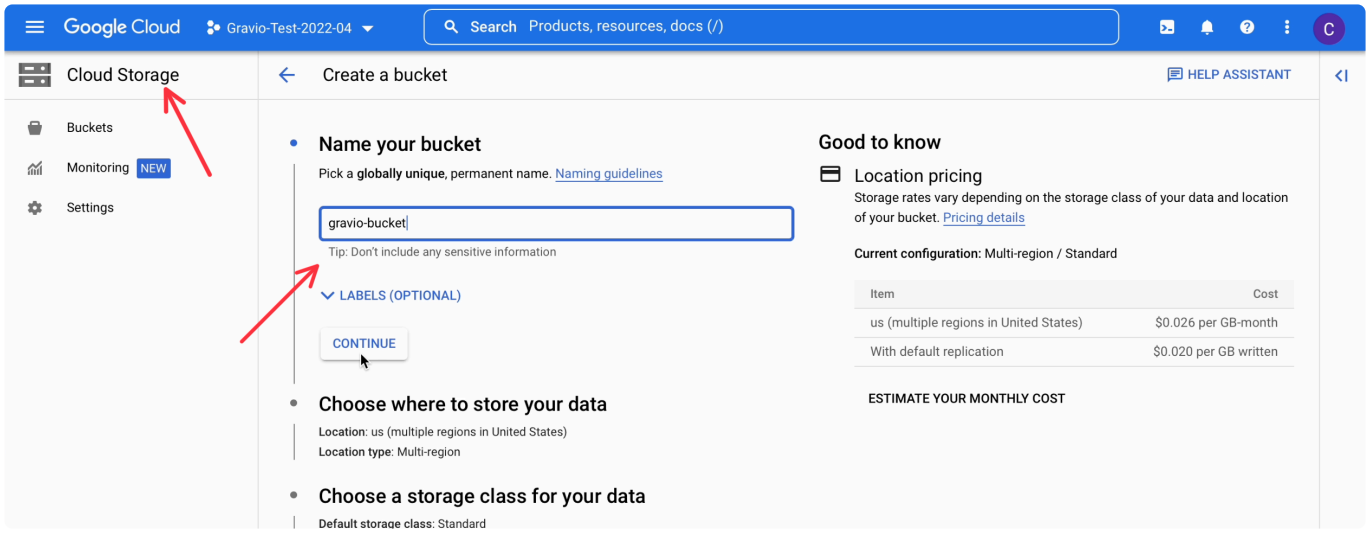
When created, the key file will be downloaded to your PC.



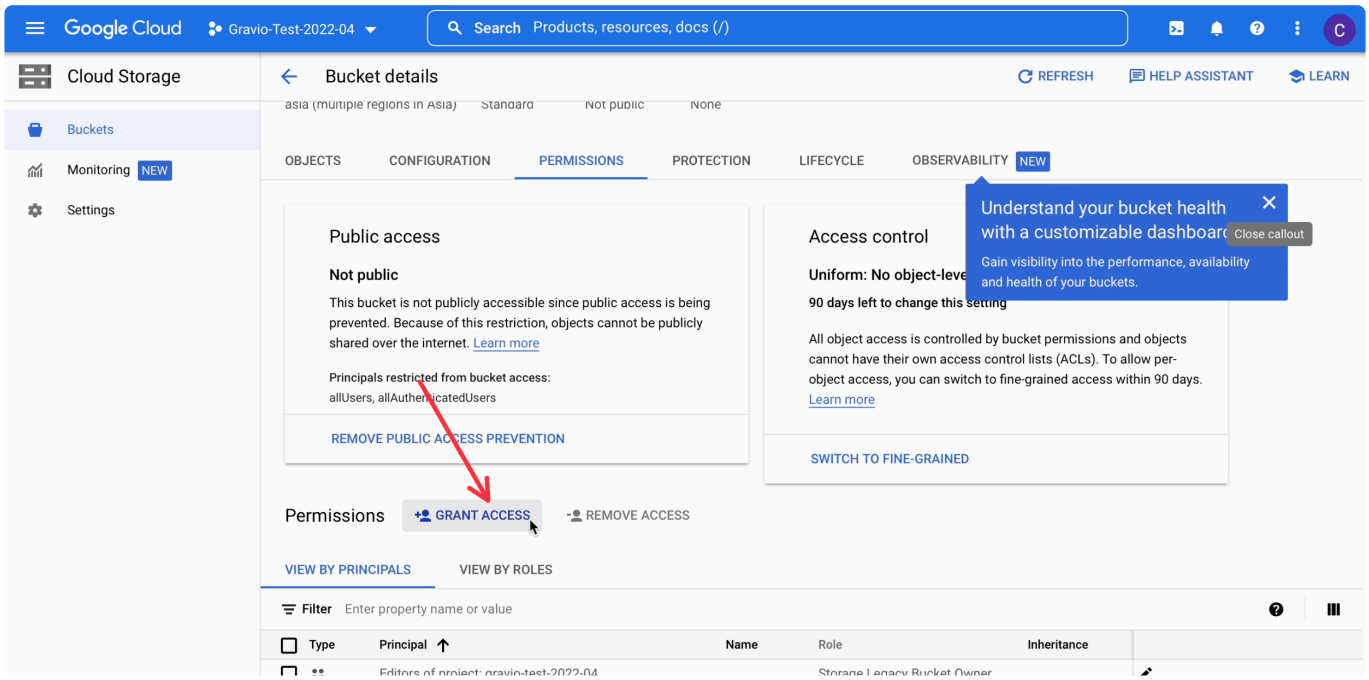
Set the service account as a principal in Google Cloud Storage

Set the service account created in IAM and Administration as a principal in Google Cloud Storage.

Make sure you have a bucket. If you don’t have one yet, create one:

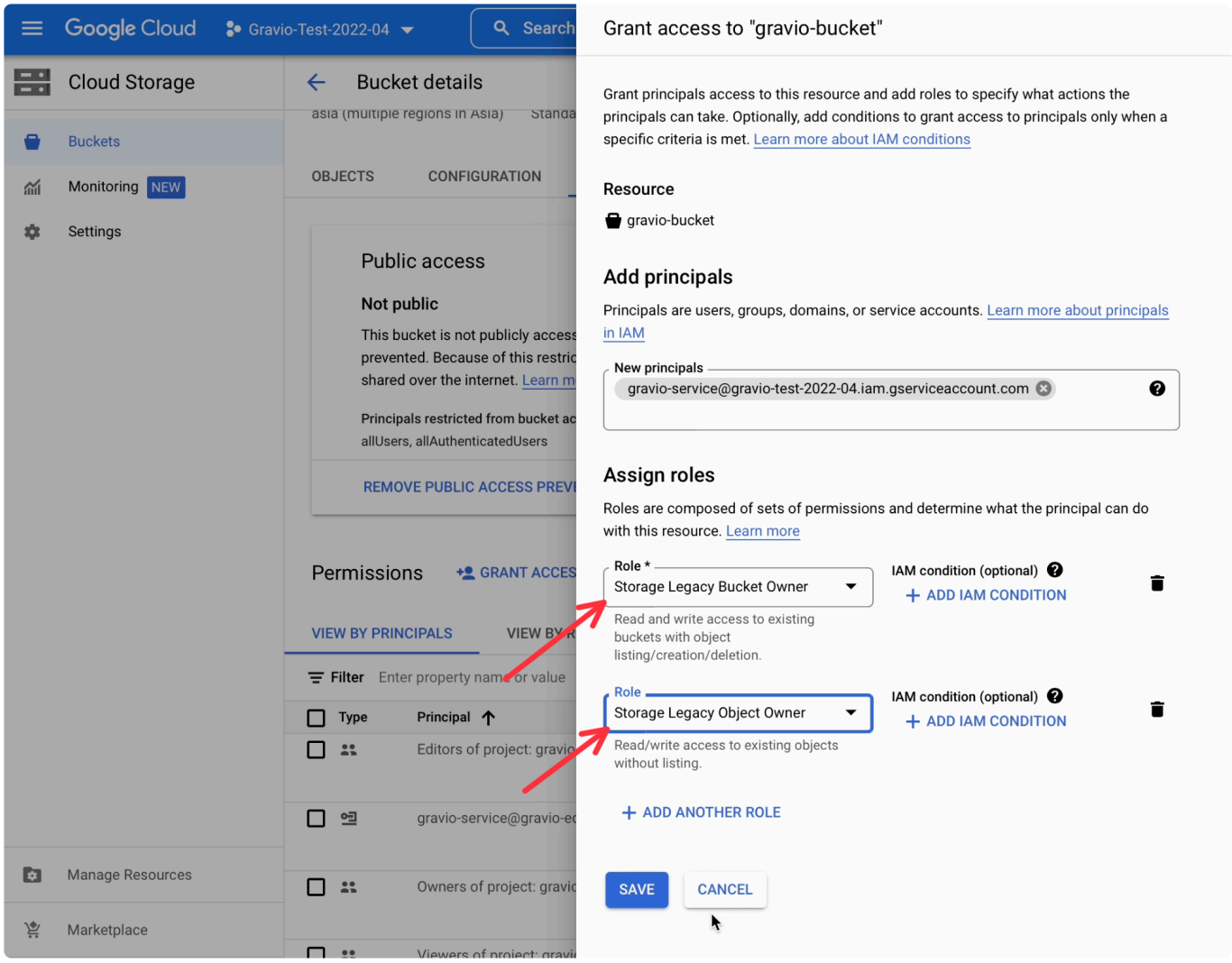


Select the bucket to read/write from the Google Cloud Storage browser menu and select “Create” for the principal from “Permissions”.

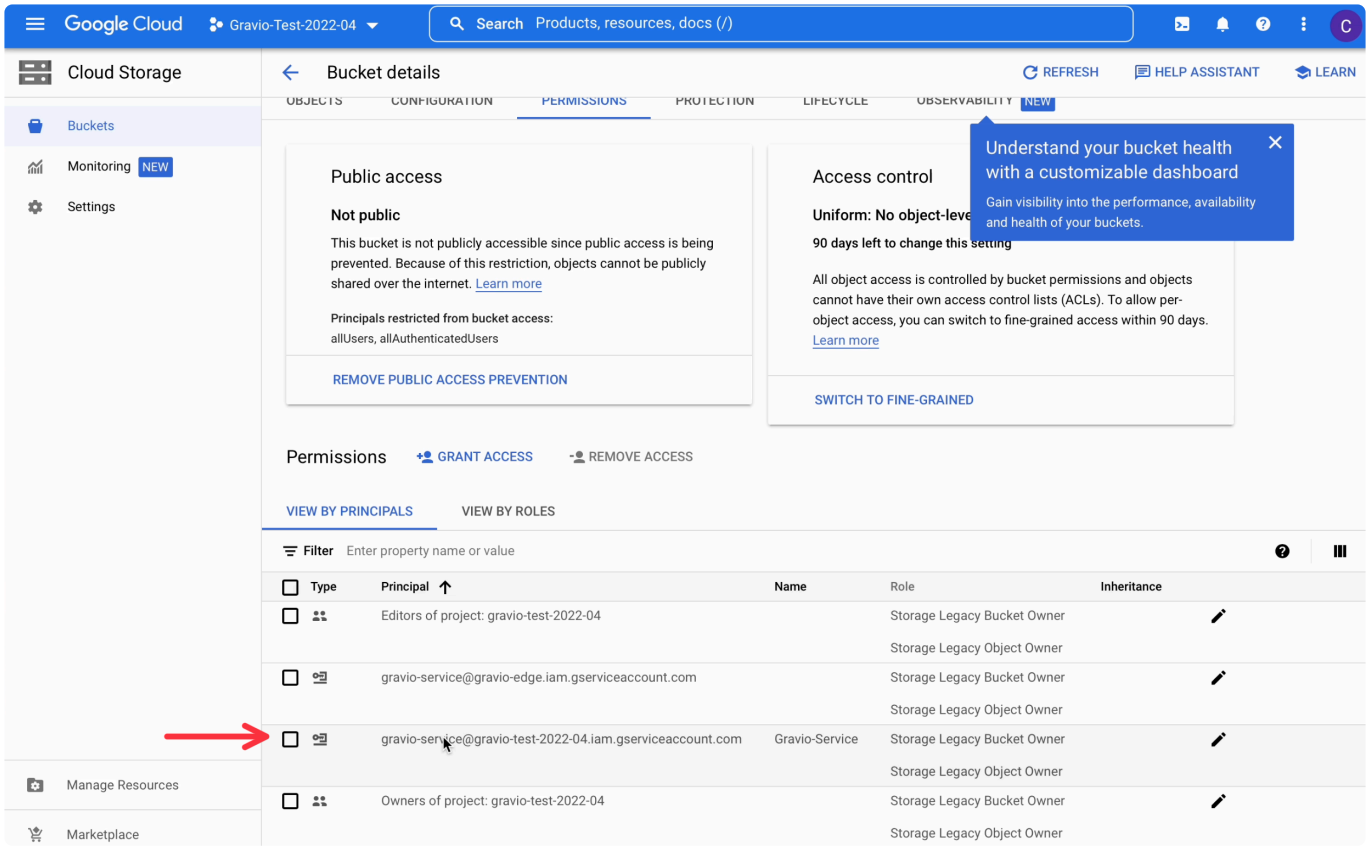


Enter the service account you created above in the “New Principal” field.

Add “Storage Legacy Bucket Owner” and “Storage Legacy Object Owner” to “Roles” and save.



Save the new principal and it will appear in the list.



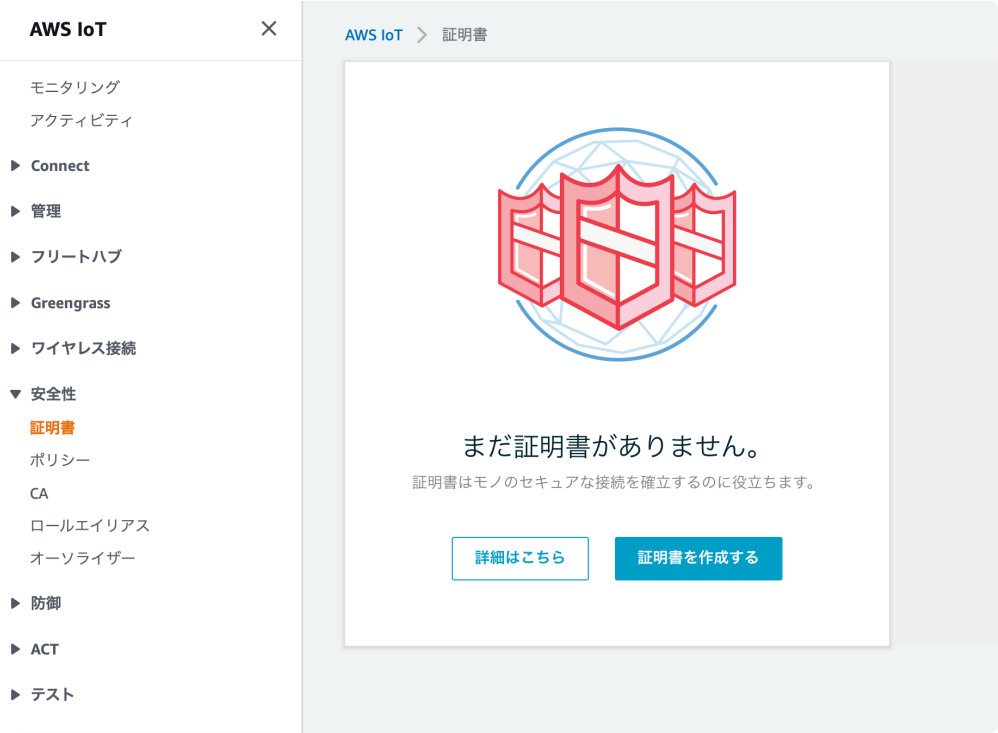
Configuration in the Google Cloud Console is finished.

13.12. How to set up AWS IoT MQTT

This guide takes you through the steps to use AWS IoT with MQTT

Create and Download the Certificate

Open AWS IoT, select “Certificates” in the left menu “Security” and click “Create Certificate”.



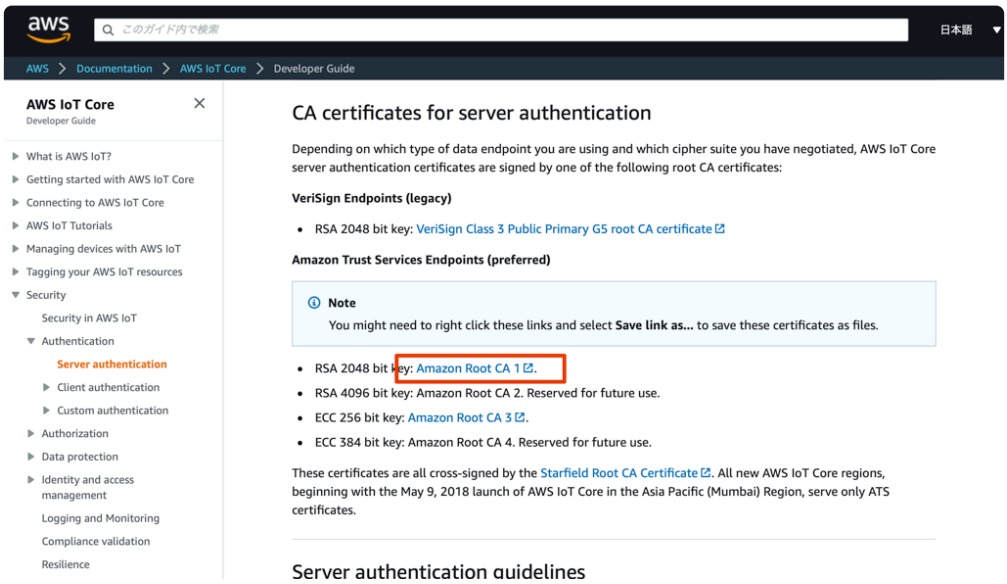
Select 1-Click Create Certificate (recommended).

Once the certificate is created, you will be able to download 3 certificates, download all of them.

Note that no public key will be used.

Select “Download” to download the certificate for the Amazon IoT root CA.

Click on the link RSA 2048 bit key : Amazon Root CA 1 to view the certificate and copy and paste the text into your editor.

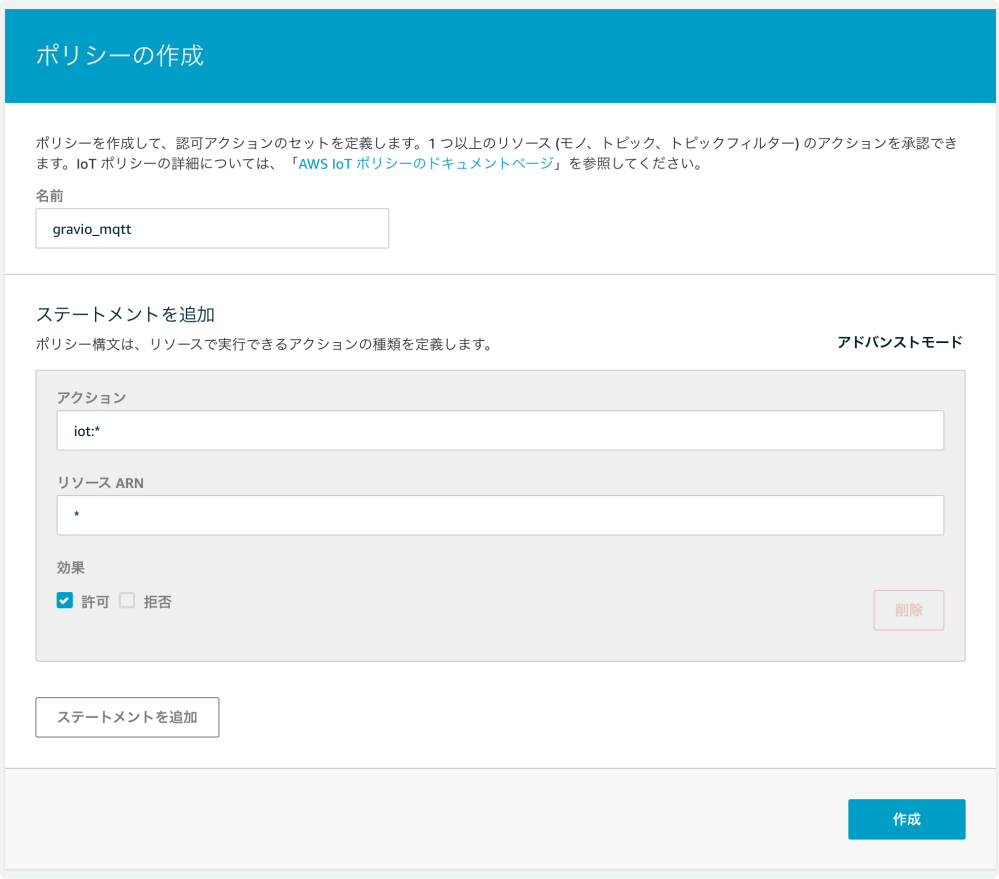


```
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM
9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
```

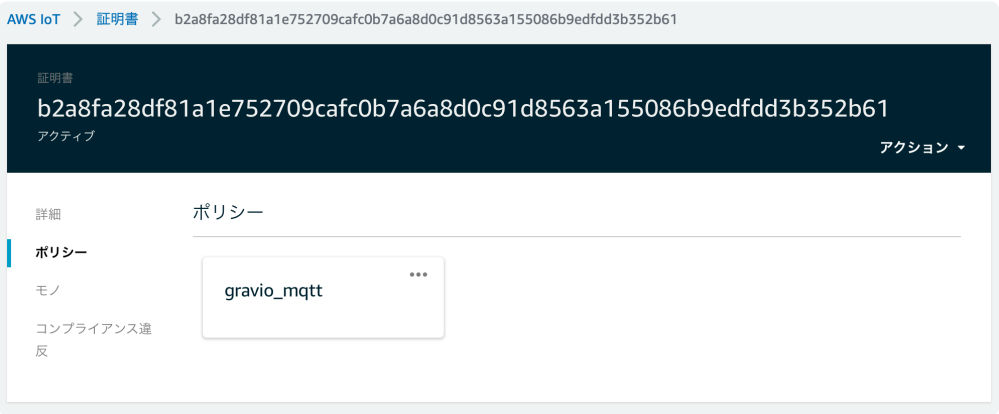

Configure the Policy

Next, configure a policy: open AWS IoT, select “Policies” under Safety, and click “Create Policy”.

Enter a name for the policy, enter “iot:**“ for the action, “**” for the resource ARN, check “Allow” for the effect and click “Create”.

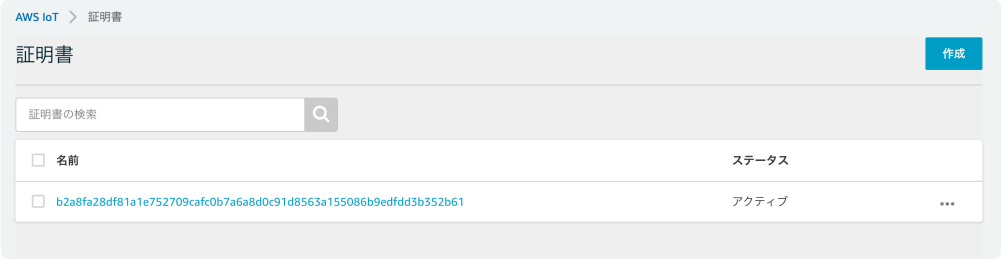


The policy has been created.



Associating a Policy with a Certificate

Select “Certificates” under Safety and click on the certificate you created.



Select the action and click on “Attach Policy”.

Check the policy you just created and click “Attach”.

Check the Entry Point

Check the MQTT publish and subscribe entry points.

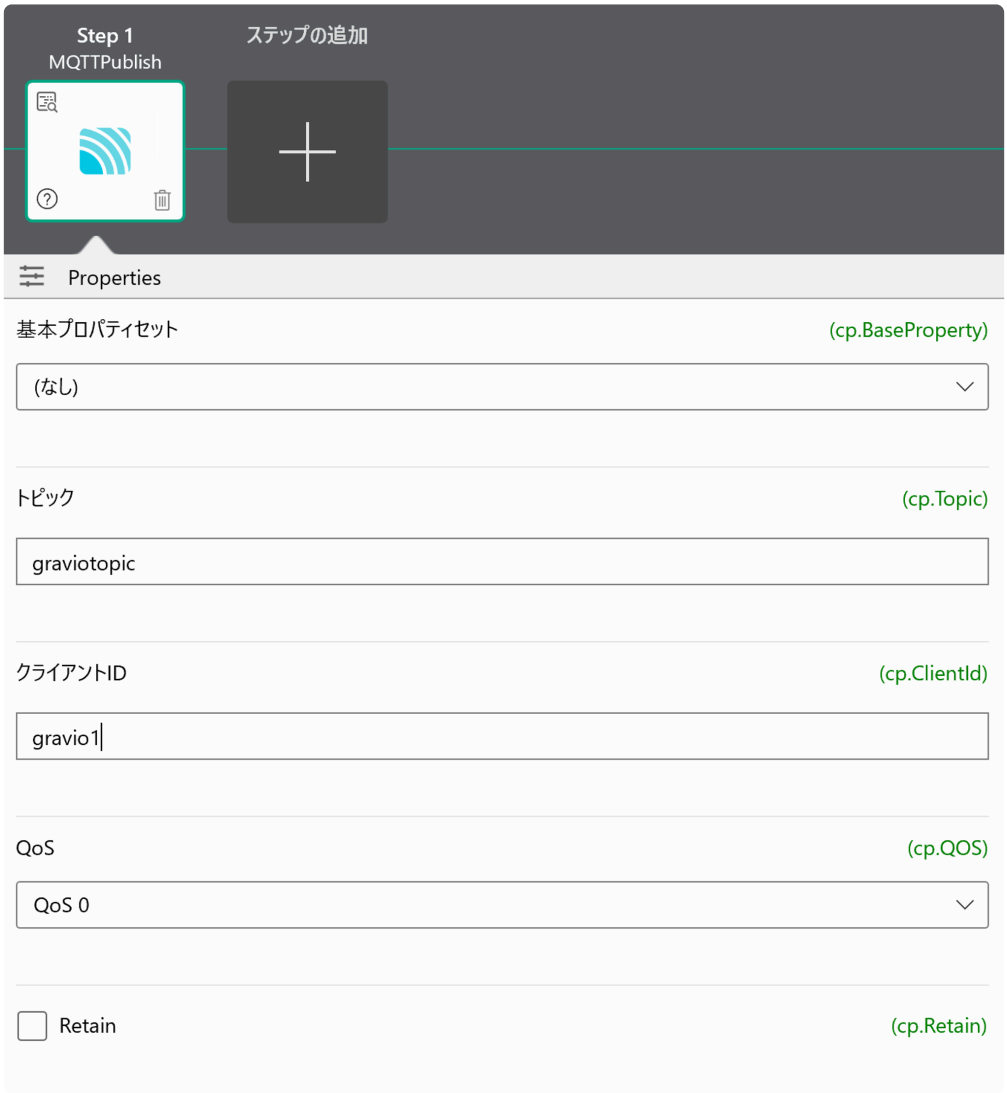
Click on the “Settings” menu on the left. Endpoints will be displayed and use this one.

After confirming the settings up to the endpoint, the configuration in AWS IoT is finished.

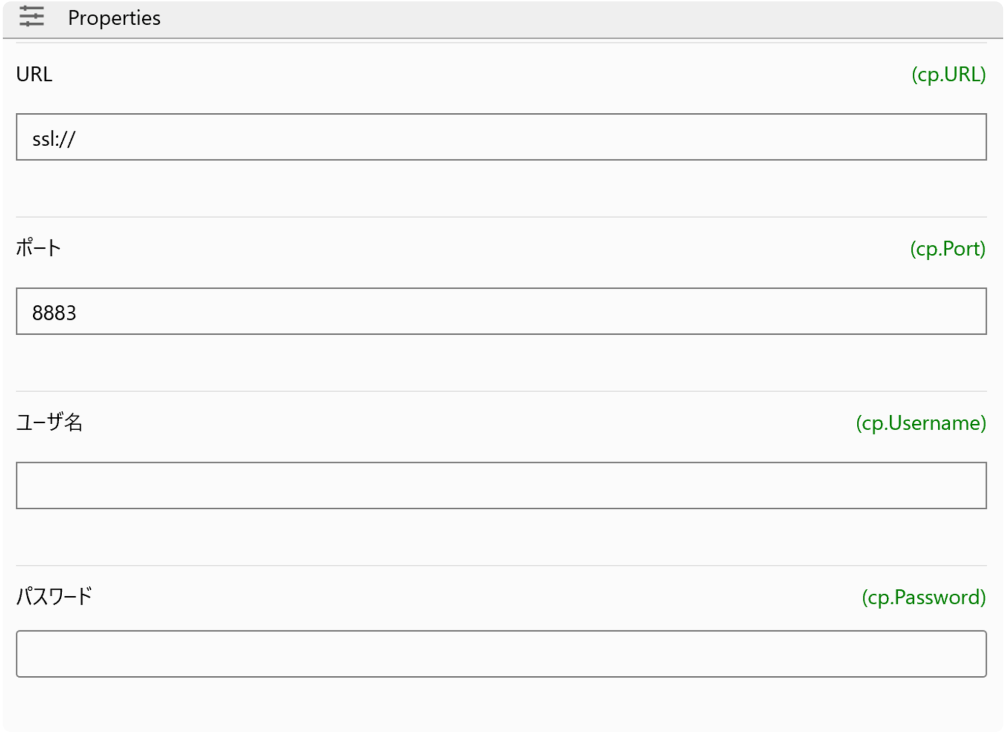
MQTTPublisher configuration in Gravio Studio

In Gravio Studio, set the MQTTPublisher properties as follows

- Topic can be any value.
- Client ID is a unique value.



URL is “ssl://” followed by the entry point.
The port is “8883” and the username and password are left blank.



Select the downloaded files for the certificate, private key, and root certificate. Leave the private key passphrase blank.

You are now set up to send data from MQTTPublisher to AWS IoT's MQTT.

13.13. How to configure Azure IoT Hub (MQTT)

Here is an example of how to send messages to Azure IoT Hub (MQTT).

Preparations

Prepare a Linux or Mac to create a certificate.

Downloading the tools

To use a certificate in Azure IoT Hub, please download the command line tools to your Linux or Mac from here.
For Windows, execute the bash command from PowerShell and use bash.

Enter the following commands to create a certificate that can be pasted into the field.

Unzip the downloaded zip file and move to the tools directory.
```
cd <path>/azure-iot-sdk-c-main/tools
```

Give the tool execution permissions.
```
chmod +x ./certGen.sh
```

Creating a CA certificate

Run the tool to create a CA certificate.
```
./certGen.sh create_root_and_intermediate
```

Creating a device certificate

Run the tool to create a certificate for the device.
```
./certGen.sh create_device_certificate <device id>
```
※ is where you specify the device ID.

Create a public key for the device.
```
openssl pkcs12 -in certs/new-device.cert.pfx -clcerts -nokeys -out <device i
d>.crt
```
※If you are asked to enter a password, enter the password that is displayed on the screen.

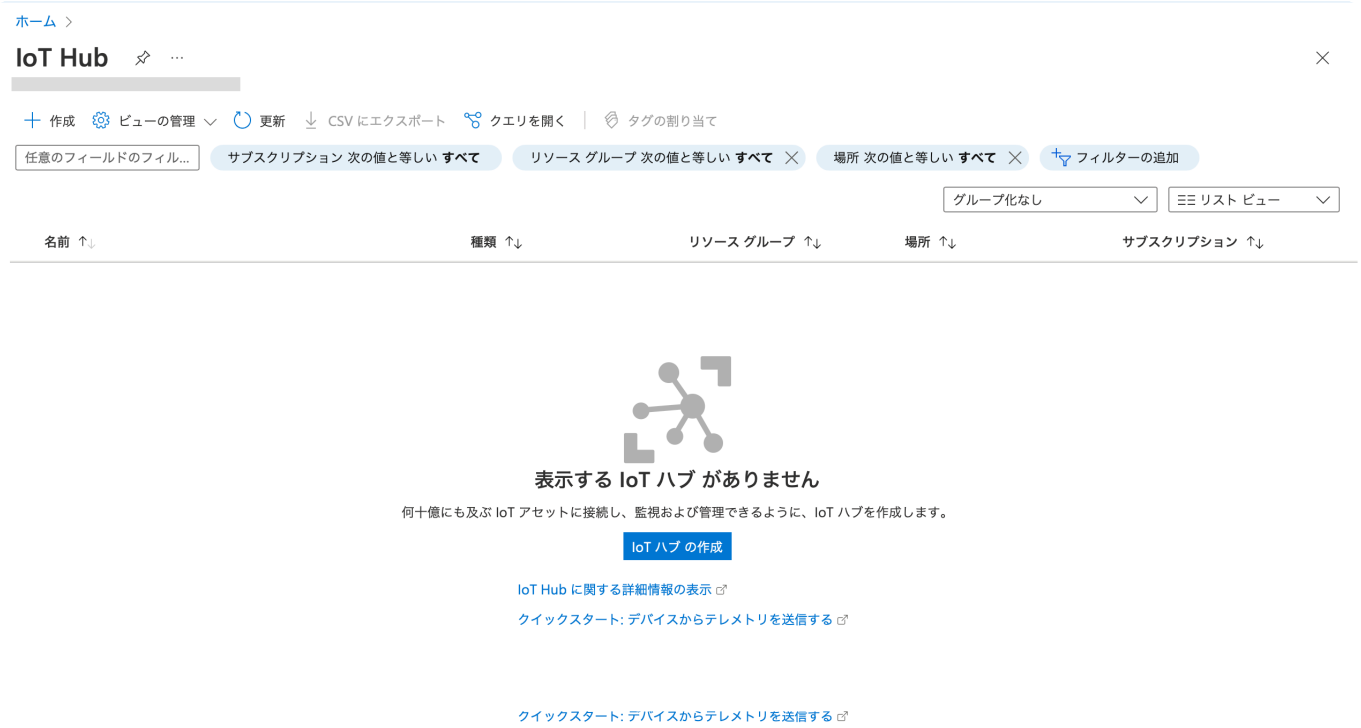
Create a private key for the device.
```
openssl pkcs12 -in certs/new-device.cert.pfx -nocerts -nodes -out <device i
d>.key
```
※If you are asked to enter a password, enter the password that is displayed on the screen.

.crt and .key will be used in Gravio Studio.

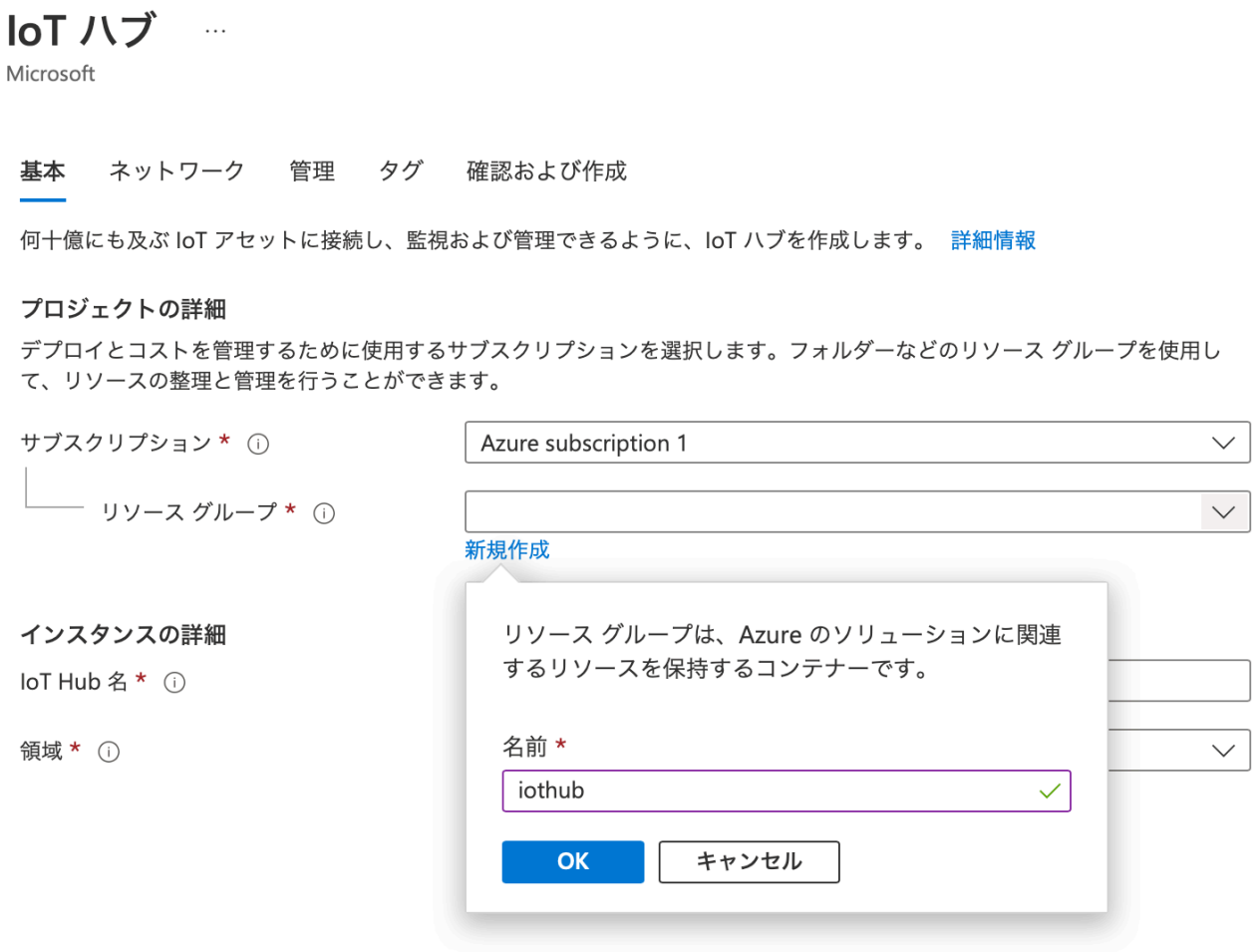
If you are using multiple devices, please create one for each device.

Creating an IoT Hub in Azure

Log into Azure and display and create the IoT Hub:

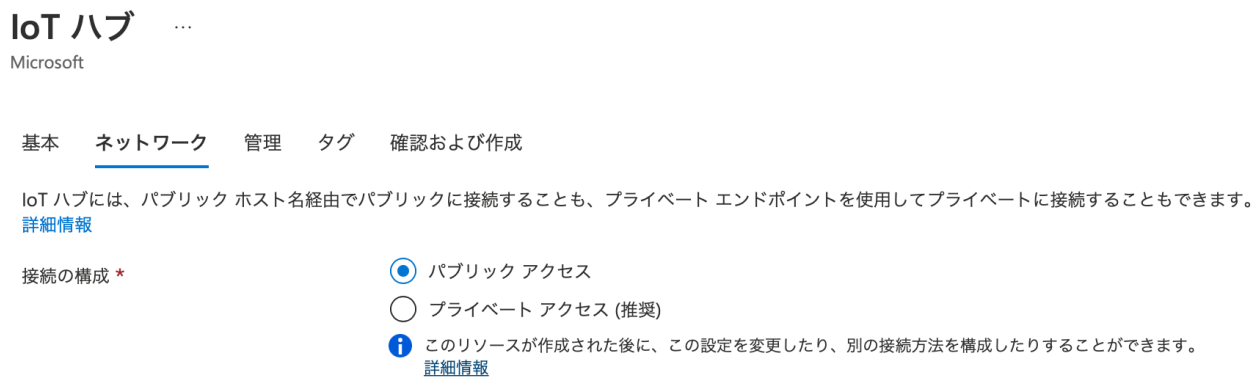


Create a new resource group. In the sample, we named it “iothub”.



Enter the IoT Hub name. In the sample, we named it “gravio-mqtt”.

Specify the network.



Specify the scaling level and units in Management. In the sample, we selected “F1:Free Level”.

IoT ハブ ...

Microsoft

基本 ネットワーク 管理 タグ 確認および作成

それぞれの IoT ハブには、特定レベルの特定数のユニットがプロビジョニングされます。レベルとユニット数によって、1 日に送信できるメッセージの最大クォータが決まります。 詳細

スケーリング レベルとユニット

価格とスケールティア * F1: Free レベル

ソリューションに合った正しい IoT ハブのレベルを選択する方法について

IoT ハブ F1 のユニット数 1

IoT ハブをスケーリングできる度合いを決定します。ニーズが増大した場合は、後でこれを変更できます。

Defender for IoT オフ

Microsoft Defender for IoT は、Azure IoT Hub、IoT Edge、デバイスに脅威からの保護のための追加のレイヤーを付加する個別のサービスです。このサービスに対しては個別に課金されます。Defender for IoT ではデータを処理した後、IoT Hub とは異なる地理的な場所に格納する場合があります。詳細情報

価格とスケールティア	F1	device-to-cloud メッセージ	有効
1 日あたりのメッセージ	8,000	メッセージ ルーティング	有効
月あたりのコスト	0.00 USD	cloud-to-device コマンド	有効
Defender for IoT	無効	IoT Edge	有効
		デバイス管理	有効

ロールベースのアクセス制御

アクセス許可モデルを Azure ロールベースのアクセス制御 (RBAC) のみに、または共有アクセス ポリシーと RBAC の組み合わせに変更します。 詳細情報

- ○ RBAC のみ
- ◉ 共有アクセス ポリシー + RBAC

インスタンス内の要素を管理するには、IoT Hub データ API に対するアクセスが必要です。以下の推奨されるロールを選択して、API へのフル アクセスを自分自身に許可してください。アクセスの制御 (IAM) を使用して、後で適切なロールを選択することもできます。 詳細情報

☐ IoT ハブ データ共同作成者ロールに自分を割り当てる

詳細設定

スケーリング

Device-to-cloud パーティション 2

Confirm and create with the content of the settings.

IoT ハブ ...

Microsoft

検証に成功しました。

基本 ネットワーク 管理 タグ 確認および作成

基本

サブスクリプション	Azure subscription 1
リソース グループ	iothub
領域	Japan East
IoT Hub 名	gravio-mqtt
ディザスター リカバリーが有効	はい

ネットワーク

接続の構成	パブリック アクセス
プライベート エンドポイント接続	なし
公衆ネットワーク アクセスを許可する	有効

管理

価格とスケールティア	F1
IoT ハブ F1 のユニット数	1
1 日あたりのメッセージ	8,000
Device-to-cloud パーティション	2
月あたりのコスト	0.00 USD
Defender for IoT	無効

タグ

When the deployment is complete, “gravio-mqtt” will be created.

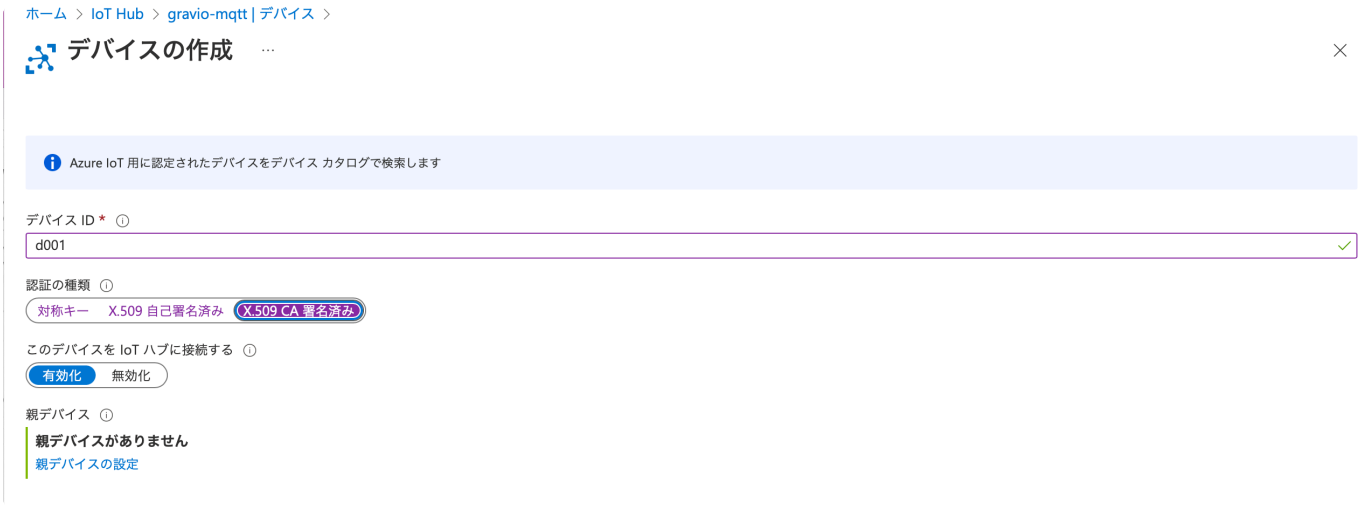
Select “gravio-mqtt”, open and create the certificate.

In Add Certificate, enter the certificate name and specify the CA certificate created in the pre-preparation. Set “Confirm certificate status at upload time” to On.

When you save the certificate, it will be displayed in the list.

Open and create the device.

The device ID is the ID specified in the device certificate. Select “X.509CA Signed” for the type of authentication.



When you save the device, it will be displayed in the list.

Open the overview and record the hostname. The hostname is the entry point for the Azure IoT Hub.

Setting up the MQTTPublisher in Gravio Studio

Set the properties of the MQTTPublisher in Gravio Studio as follows.

Property name	Description
Topic	Specify “devices/” + device ID + “/messages/events/”. For example, “devices/d002/messages/events/”
Client ID	Specify the device ID. For example, “d001”
QoS	(Leave as default)
Retain	(Leave as default)
Timeout	(Leave as default)
URL	Specify the hostname following “ssl://”. For example, “ssl://gravio-mqtt.azure-devices.net”

Port	Specify 8883
Username	Specify the hostname + “/” + device ID. For example, “gravio-mqtt.azure-devices.net/d001”
Password	(Don’t specify)
Certificate	Specify the public key of the device created in the pre-preparation
Secret Key	Specify the private key of the device created in the pre-preparation
Secret Key Passphrase	(Don’t specify)
Root Certificate	(Don’t specify)

With this, the settings for sending data from MQTTPublisher to Azure IoT Hub (MQTT) are complete.

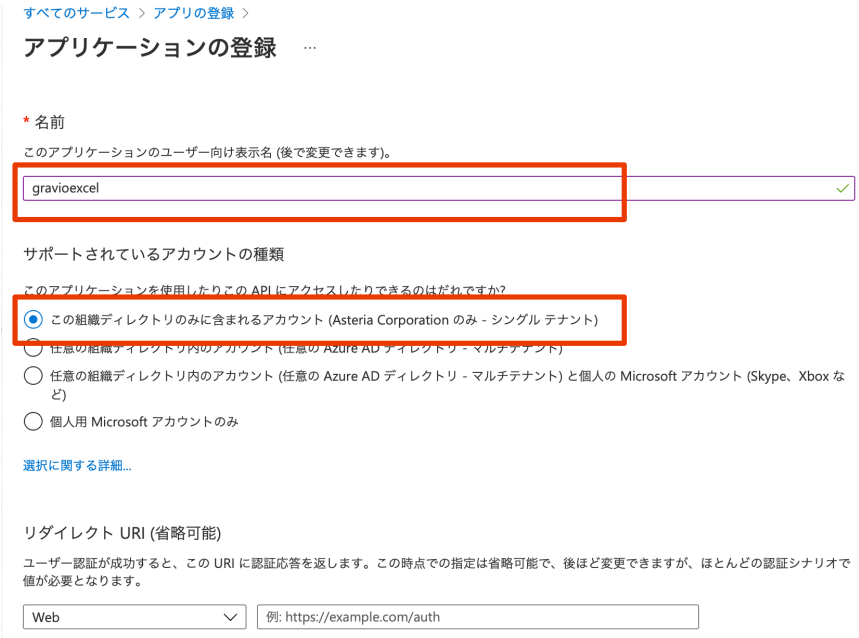
13.14. How to Register Azure Apps (Microsoft365)

```
```

Here is an example of how to post an Excel file to Microsoft365 using the WriteToExcel365 component. In order to use this component, you need to have a contract and prior setup with Microsoft365 for Business and OneDrive for Business.

Preparing Microsoft365 for Business and OneDrive for Business

Please prepare a Microsoft365 account and log in to the Azure Portal. From the contract page, select “Register App” to register your app.



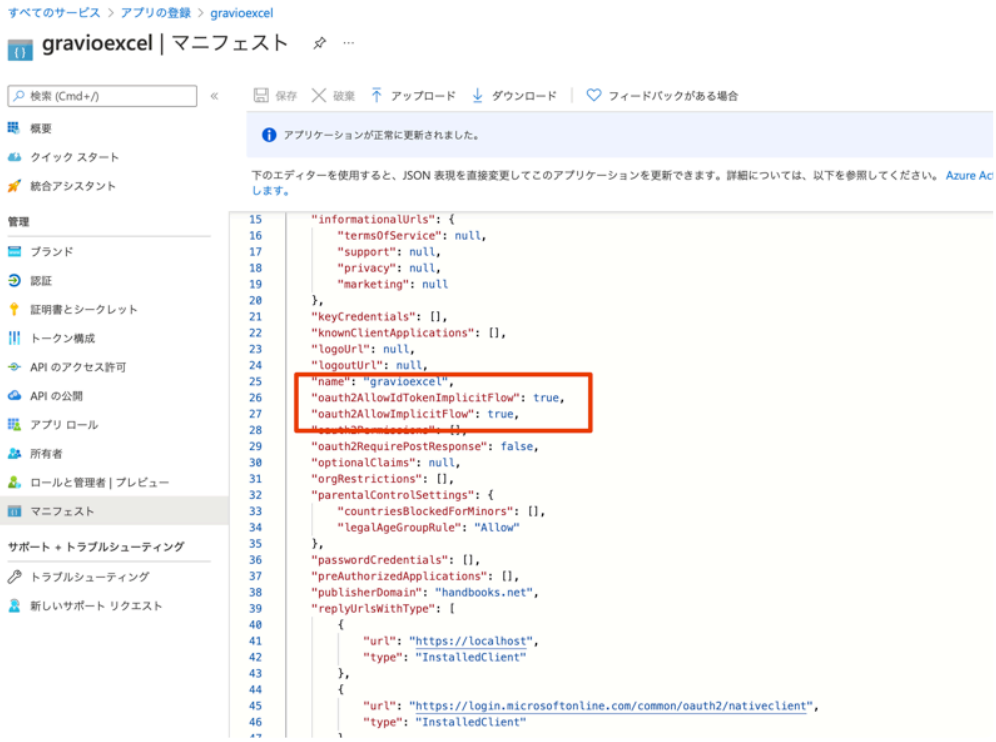
After creating the app, open the details page and note down the Application ID and Directory (Tenant) ID displayed. These values, along with your Microsoft365 account’s email address and password, will be set in the properties of the WriteToExcel365 component.

Next, from the authentication menu, confirm that the type of accounts that are supported is “Accounts only included in this organization directory” and set “Allow public client flow” to “Yes” and save.

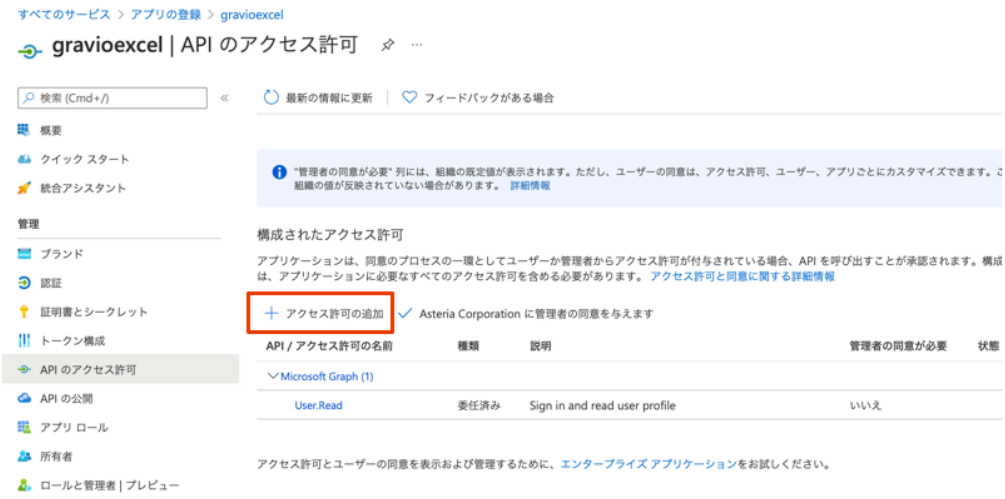
Select “Add Platform” and choose “Mobile and Desktop Applications” for the platform configuration.

Check the first item of the Redirect URI, enter “https://localhost” in the Custom Redirect URI, and press the configure button to save.

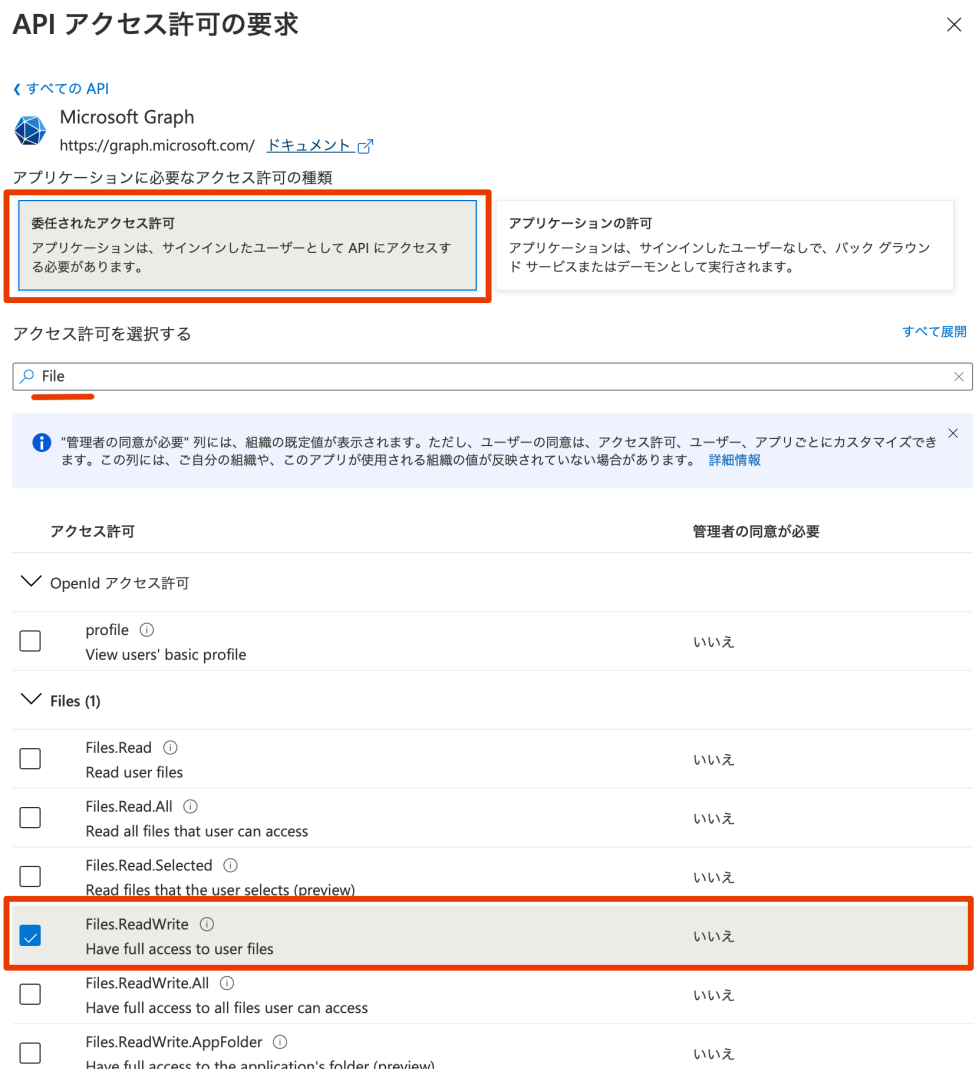
Next, from the manifest menu, set “oauth2AllowIdTokenImplicitFlow” and “oauth2AllowImplicitFlow” to “true” and save.



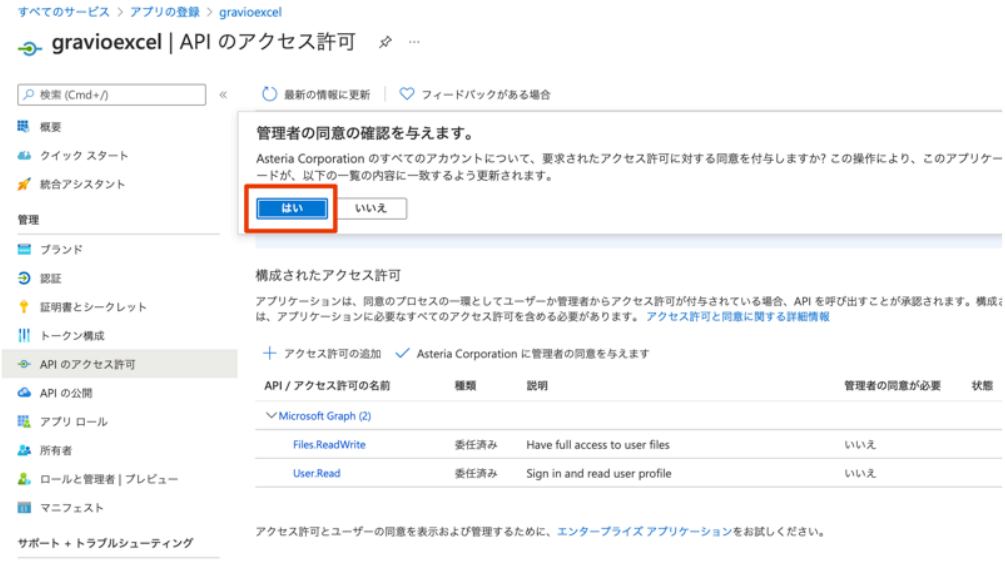
Next, from the API permissions menu, click “Add Permission” and select “Microsoft Graph” from the frequently used Microsoft APIs.



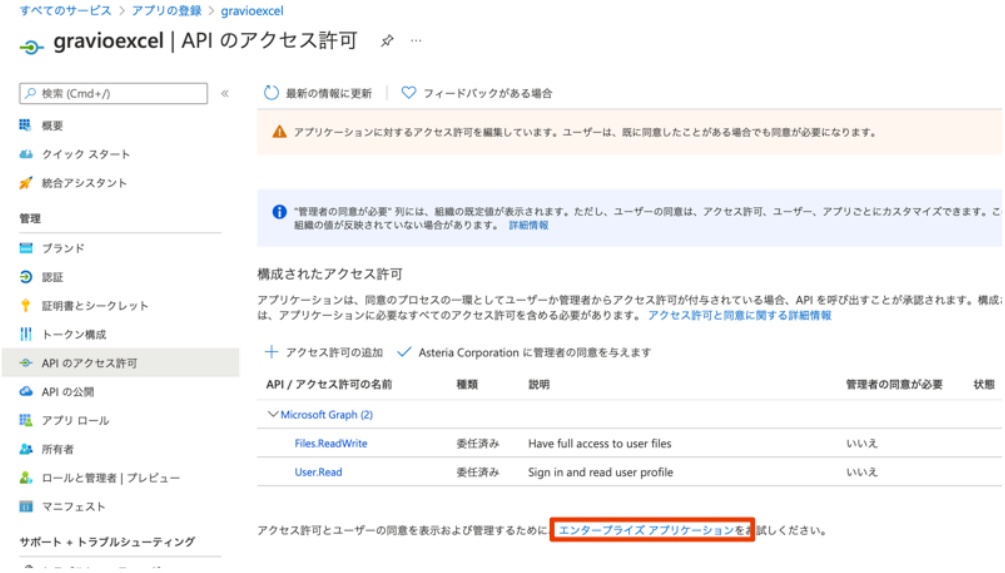
For API permission requests, select “Delegated Permissions” and to select permissions, enter “File” and the permissions “Files” will be displayed, so expand it and check “Files.ReadWrite” and press the configure button.

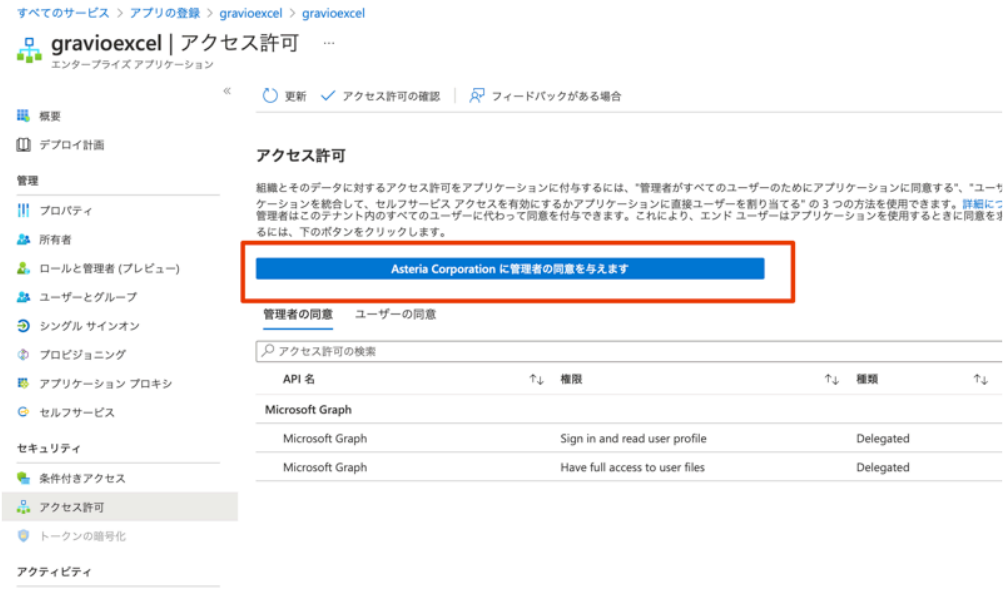


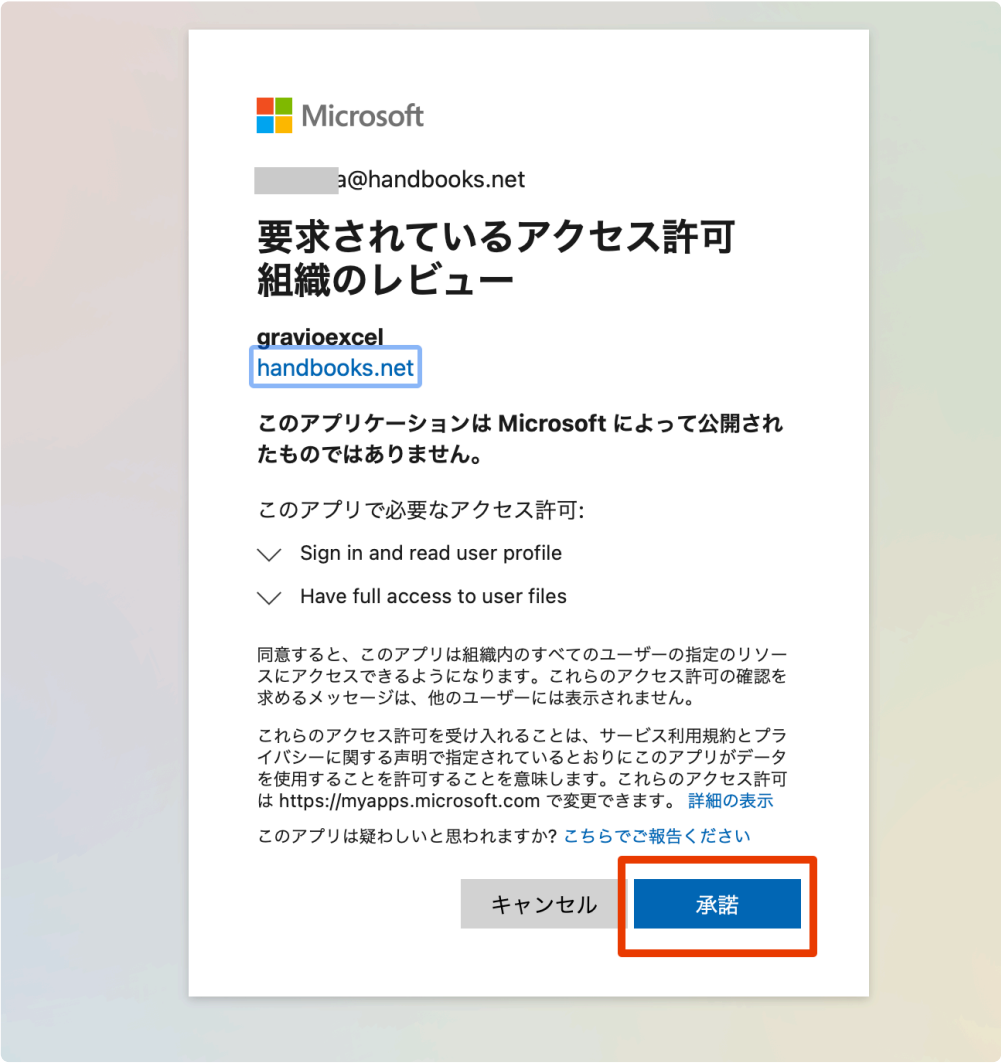
Confirm that “Files.ReadWrite” is displayed under API / Permission Name and click “Give Admin Consent”. When the confirmation dialog appears, click “Yes”.



Click “Enterprise Application” and click “Give Admin Consent” under permissions. When the “Review permissions for the organization of the required permissions” opens, click “Accept”.







With this, the preliminary preparations are complete.
```
```

13.15. How to register a bot for LINE WORKS (Pre-June 2023 Version)

In order to use the LINE WORKS component, the LINE WORKS administrator needs to initialize the Bot.

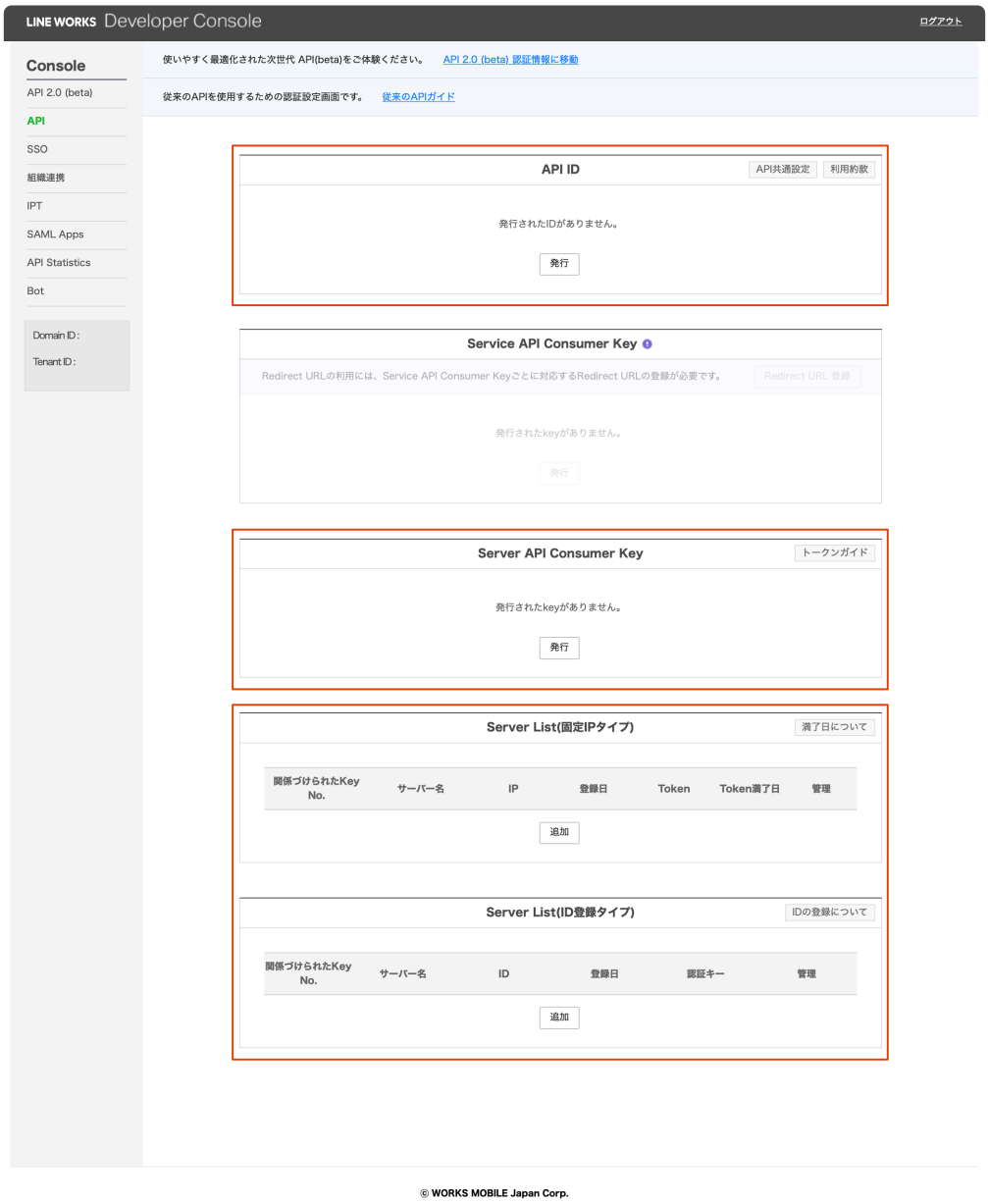
LINE WORKS Developers

1. Creating a LINE WORKS Developers account

Create an account from here and create a Bot for LINE WORKS.

2. API settings

The LINE WORKS component uses the API, so select the API menu. Do not use API 2.0 (beta).



3. Issuing an API ID

Press the “Issue API ID” button to create it. The created API ID will be used in the LINE WORKS component properties.

4. Issuing a Server API Consumer Key

Press the “Issue Server API Consumer Key” button to create it. The created Server API Consumer Key will be used in the LINE WORKS component properties.
The usage scope of the Server API is set to “Add/Inquiry”. Please set the Token validity period and automatic extension according to the operation rules. For the test environment, set it to automatically renew after 365 days.

5. Adding a Server List

You can use either a fixed IP type or an ID registration type for the Server List.
You can use the fixed IP type if the Gravio Hubkit is connected to the Internet and has a global fixed IP address.
Usually, please use the ID registration type. An ID and an authentication key will be issued for the ID registration type, which will be used in the LINE WORKS component properties.
Download the key file by pressing the download button for the authentication key.

6. Creating a Bot

Press the Bot registration button from the Bot menu.

Enter the Bot name and description, select the conventional API for the API Interface, check the Bot policy to allow invitation to multiple people's chat rooms, and set the main person in charge of the administrator.
When the registration is completed, the Bot No: will be displayed, which will be used in the LINE WORKS component properties.

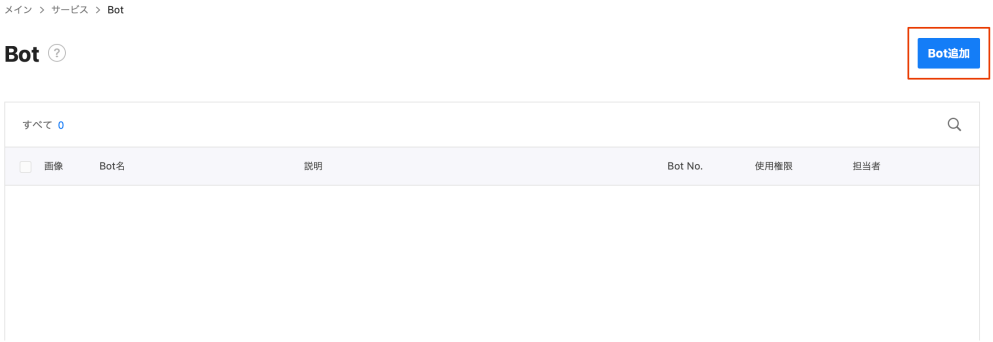
LINE WORKS Admin

1. Logging into the LINE WORKS admin screen

Administrators log in from here to display the LINE WORKS administration screen.

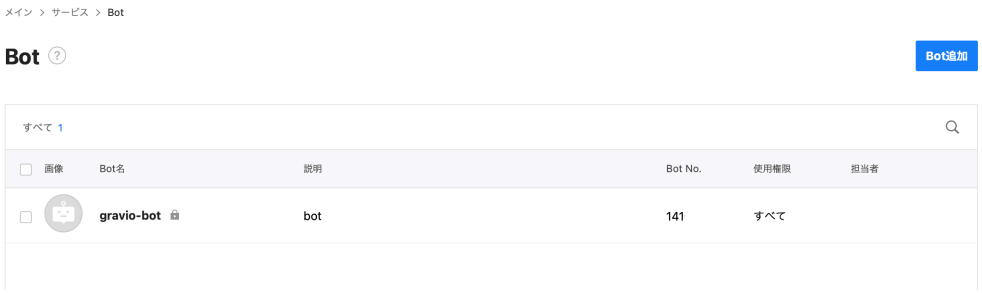
2. Bot settings

Select Bot from the service menu and press the Bot add button.



The Bot created in LINE WORKS Developers will be displayed, so press the add button.

When you redisplay the Bot screen, the added Bot will be displayed.



3. Adding a Bot to a chat room

If you want to use the Bot in a chat room, please create a new chat room or add it to an existing chat room.

4. Confirmation of the chat room ID

Even if you create a chat room with the LINE WORKS app or website, the chat room ID will not be displayed.
Therefore, please have a member send a message in the chat room where the Bot will be used.

After a while, select Talk from the audit menu and search, and the

13.16. How to register a bot for LINE WORKS

To use the LINE WORKS component, the LINE WORKS administrator must perform initial Bot settings.

LINE WORKS Developers

1. Creating a LINE WORKS Developers Account

Create an account from here and set up a Bot for LINE WORKS.

2. Creating an App

The LINE WORKS2 component uses API 2.0, so create a new app from the API 2.0 menu.

When you create an app, Client ID and Client Secret will be generated. Use these in the LINE WORKS2 component properties. For the Redirect URL, please specify “http://127.0.0.1:48888/callback”. This will also be used in the LINE WORKS2 component properties.

For OAuth Scopes, press the manage button and select ‘bot’ from the list.

3. Creating a Bot

Press the Bot registration button from the Bot menu.

Enter the Bot name and description. Select API 2.0 for the API Interface. For the Bot policy, check ‘Can be invited to multi-user chat rooms’ and set the main administrator. After registration is complete, the Bot No: will be displayed. Use this in the LINE WORKS2 component properties.

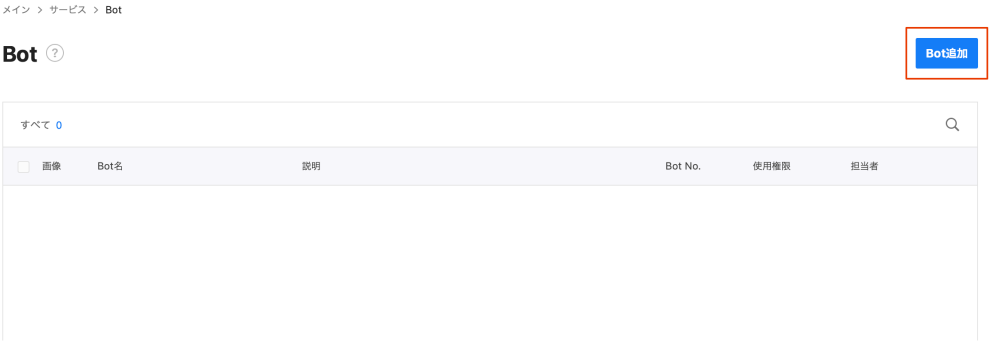
LINE WORKS Admin

1. Logging in to the LINE WORKS Management Screen

Administrators log in from here to display the LINE WORKS management screen.

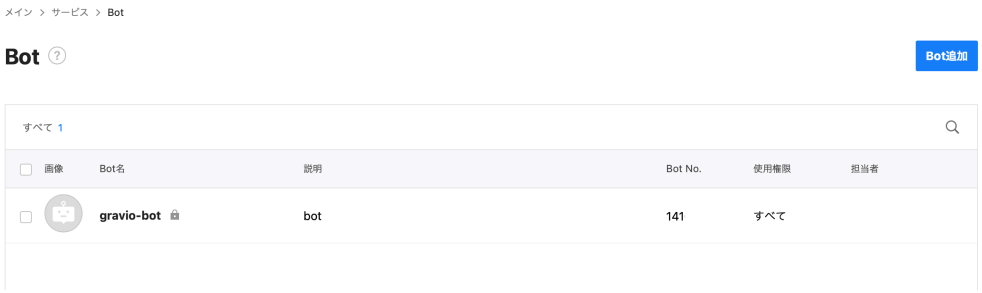
2. Bot Settings

Select ‘Bot’ from the service menu and press the ‘Add Bot’ button.



The Bot created in LINE WORKS Developers will be displayed. Press the ‘Add’ button.

When you re-display the Bot screen, the added Bot will be displayed.



3. Adding a Bot to the Chat Room

If you want to use the Bot within a chat room, create a new chat room or add it to an existing chat room.

4. Confirming the Chat Room ID

Even if you create a chat room using the LINE WORKS app or website, the chat room ID is not displayed. Therefore, have a member send a message in the chat room where the Bot will be used.

After a while, select ‘Talk’ from the audit menu and search. The chat room ID will be displayed. Use this in the LINE WORKS2 component properties.

With this, the preliminary preparation is complete.

13.17. About Gravio Hub and Gravio Zigbee Receiver

The maximum number of devices that can be connected to the Gravio Hub and Gravio Zigbee receiver is 64.

13.18. Note Regarding CO2 Sensors

It takes about 20 minutes after the data is received for Gravio CO2 to obtain a stable value of CO2 in the environment concerned.

If the CO2 sensor value is unnaturally high for a long time.

If the CO2 sensor value is unnaturally high for a long time, try resetting the CO2 sensor.

How to reset : Press and hold the pairing button for 18 seconds. Then release it, and if successful, a white LED will appear for about 2 seconds and then turn off.
This will reset it back to the “Reference value of 400 ppm CO2 in Sweden”.

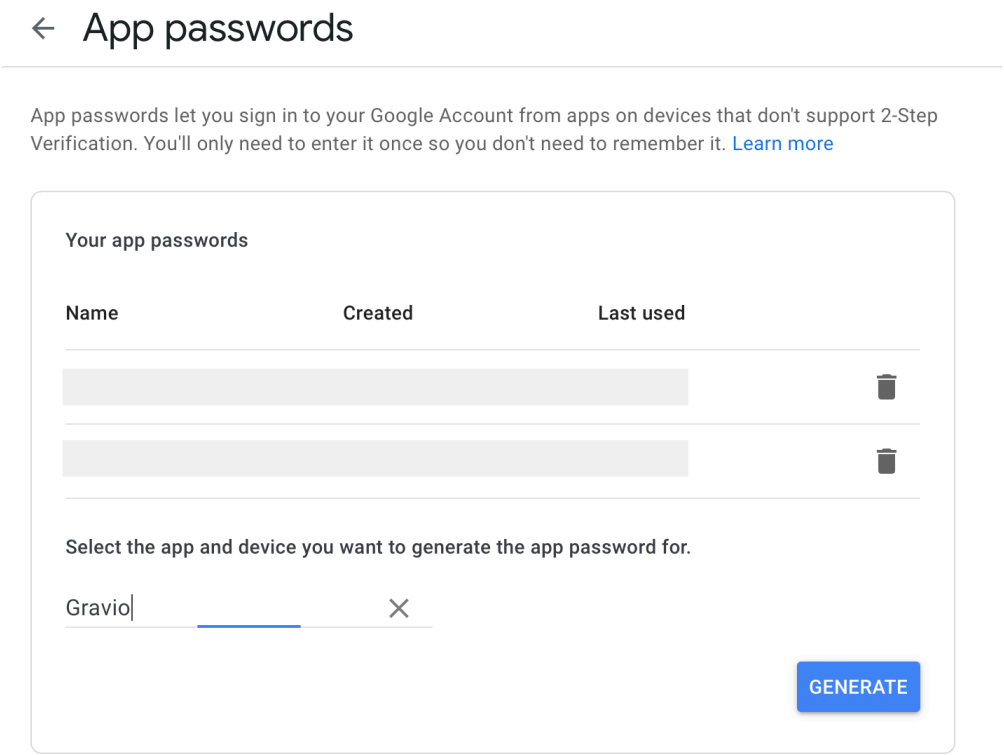
13.19. How to set up Gmail to send email : Basic Authentication (SMTP)

Example settings for sending email using a Gmail email address

Item	Description
Mail Server	smtp.gmail.com
Port Number	587
Sending Email Address	Gmail email address
SMTP Username	Username (left side of @ in the Gmail email address)
Password	Account password
Recipient Email Address	Email address to receive email notifications

※ The following settings must be done in Gmail settings.

Turn on “2-Step Verification” and create an “App Password” and specify that password.
https://support.google.com/accounts/answer/185833?hl=en&authuser=1



- For the app selection, choose ‘Mail’, and for the device name, enter any value like “Gravio”.

13.20. How to set up sending e-mail with OAuth2.0 : Advanced Authentication (OAuth2)

This document explains how to set up advanced authentication (OAuth2) for sending mail with the Mail component.

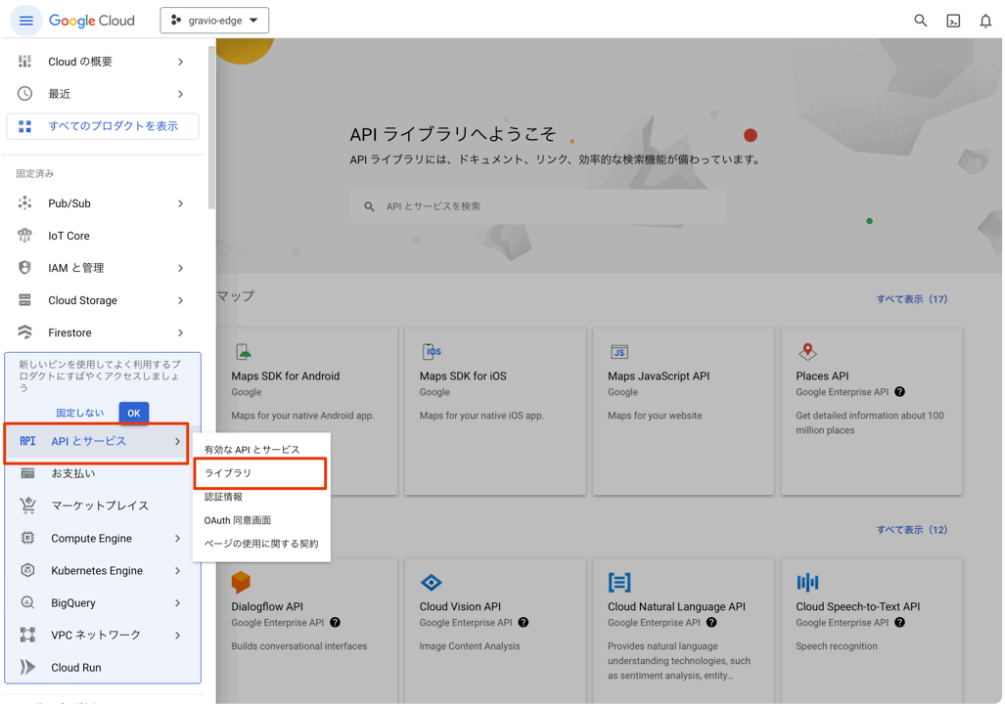
Advanced authentication (OAuth2) has been tested and confirmed to work with Gmail and Outlook.
To set up OAuth2, use GCP for Gmail and Azure for Outlook.
Correct settings require accurate knowledge of GCP or Azure.
If you are unsure about the detailed setting method, please contact Google for GCP and Microsoft for Azure.

1. How to set up mail sending in Gmail

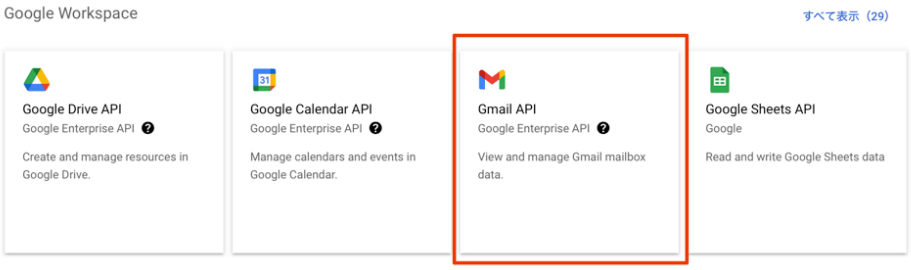
Flow of setting method

We will set up the following steps in GCP.

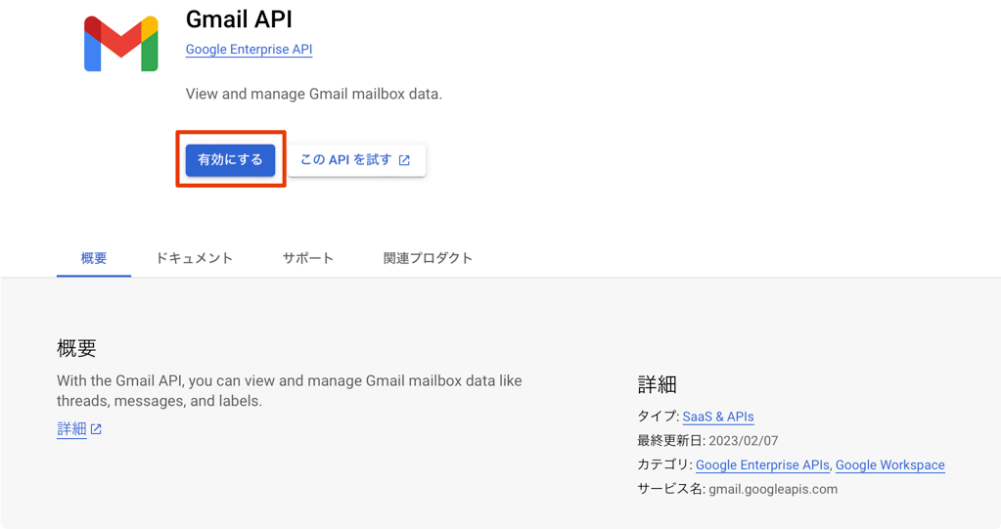
1-1. Enable GMail API from the API service library.



Select the library from the API service menu.



Select Gmail API.



Enable Gmail API.

1-2. Press the Create button by selecting “External” from the OAuth consent screen menu.

1-3. Register Gmail’s scope in the scope.

For scope, specify necessary permissions such as “Gmail view all mail, create, send, complete delete”.

1-4. Register the Gmail email address that sends GMail mail in the test user.

1-5. Check the contents registered in the summary.

1-6. Click Create Credentials from the Credentials Menu and select OAuth 2.0 Client ID.

1-7. Please select “Web application” for the type of application and give it a name. Please add “http://127.0.0.1:48888/callback” to the authenticated redirect URI and save it.

1-8. When you check the data saved from the Credentials menu in OAuth2.0 Client ID2, the Client ID, Client Secret ID, and Authenticated Redirect URI are displayed, so please take a note for use in the component.

※In the above settings, the public status is “Test”, and it is a mode where mail can be sent only to test users. By setting this mode to “Public”, you can send mail from email addresses other than the test user. If you are unsure about the detailed setting method, please contact Google. There may be cases where we cannot support individual GCP setting methods.

Now that we have the Client ID, Client Secret ID, and Authenticated Redirect URI required for OAuth 2.0, the preliminary preparation is complete.

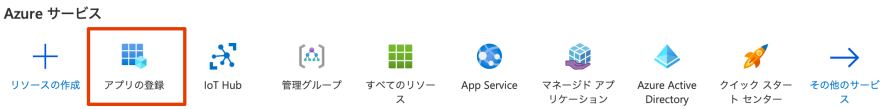
2. How to set up mail sending in Outlook

To use OAuth2.0 in Outlook, you need a contract with Microsoft365 for Business.

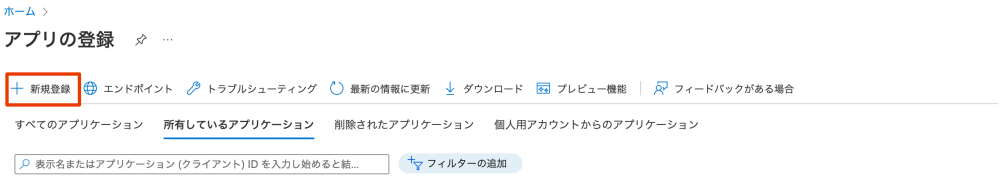
Flow of setting method

We will set up the following steps in Azure.
When sending mail, if an error like `MailSend: failed to send mail: 535 5.7.139 Authentication unsuccessful, SmtpClientAuthentication is disabled for the Tenant. Visit https://aka.ms/smtp_auth_disabled for more information.` occurs, please enable authenticated SMTP settings in Microsoft 365 management center from 2-9 onwards.

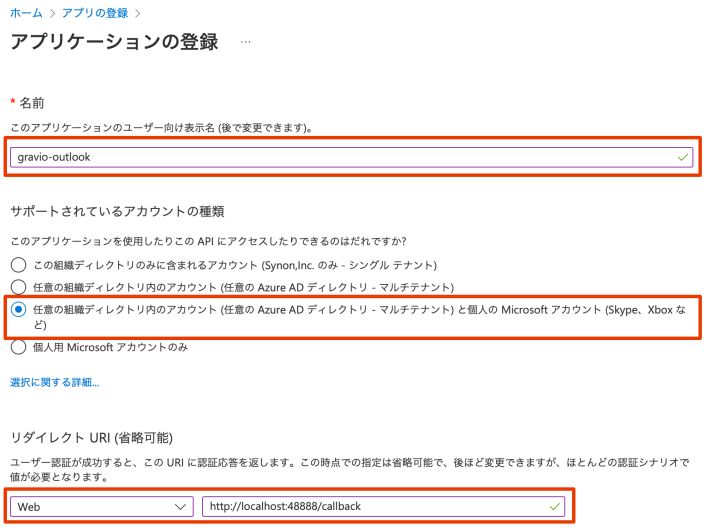
2-1. Register the app in the Azure service.



Click the Create App button.



Click the New Registration button.

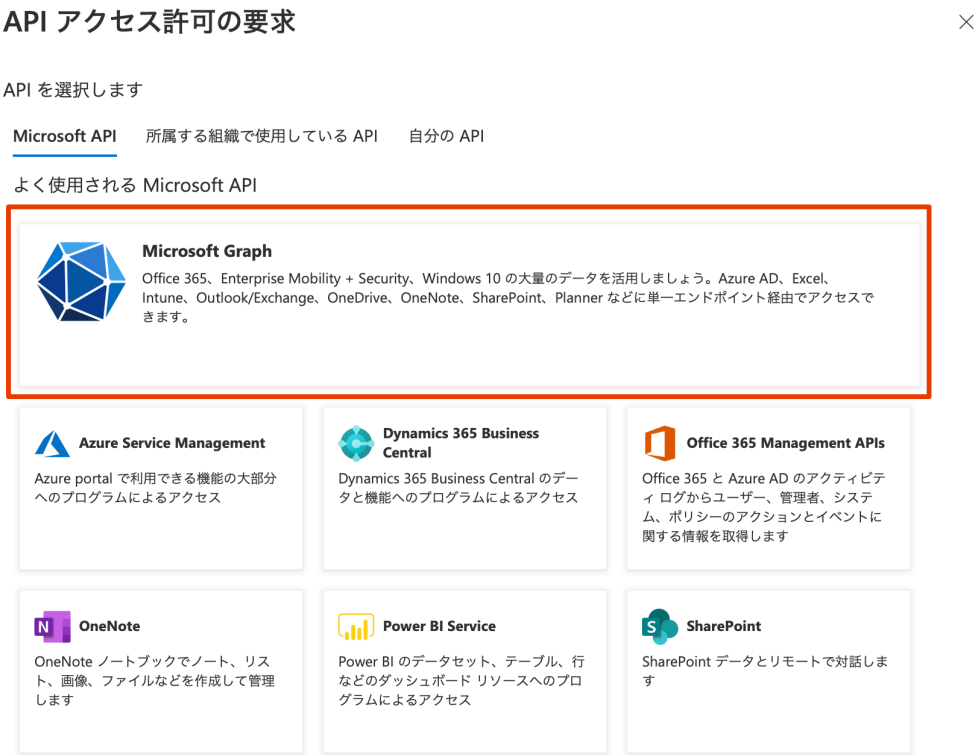


Specify the name of the application, supported account information, and redirect URI. Specify “http://localhost:48888/callback” for the Redirect URI.

2-2. When the app is registered, basic information is displayed in the summary.

2-3. Click Add Permission from the API Permissions Menu.

2-4. Select the Microsoft Graph of the Microsoft API and check Mail.Send and offline_access to add them.



API アクセス許可の要求

❮ すべての API

Microsoft Graph
https://graph.microsoft.com/ ドキュメント

アプリケーションに必要なアクセス許可の種類

委任されたアクセス許可
アプリケーションは、サインインしたユーザーとして API にアクセスする必要があります。

アプリケーションの許可
アプリケーションは、サインインしたユーザーなしで、バック グラウンド サービスまたはデーモンとして実行されます。

アクセス許可を選択する　すべて展開

mail

"管理者の同意が必要" 列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この列には、ご自分の組織や、このアプリが使用される組織の値が反映されていない場合があります。 詳細情報

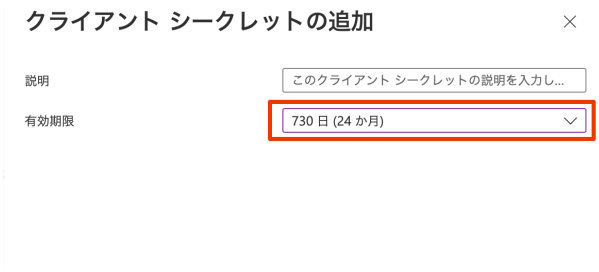
アクセス許可	管理者の同意が必要
OpenId アクセス許可	
email View users' email address	いいえ
MailboxSettings	
Mail (1)	
Mail.Read Read user mail	いいえ
Mail.Read.Shared Read user and shared mail	いいえ
Mail.ReadBasic Read user basic mail	いいえ
Mail.ReadBasic.Shared Read user and shared basic mail	いいえ
Mail.ReadWrite Read and write access to user mail	いいえ
Mail.ReadWrite.Shared Read and write user and shared mail	いいえ
Mail.Send Send mail as a user	いいえ
Mail.Send.Shared Send mail on behalf of others	いいえ

2-5. Once saved, it will be displayed in Configured Permissions. Click Grant admin consent for your organization to give consent.

2-6. Note the client ID and redirect URI.

2-7. Click New Client Secret in the Certificates and Secrets menu to create it.

Create a client secret.



You can specify an expiration date for the client secret.

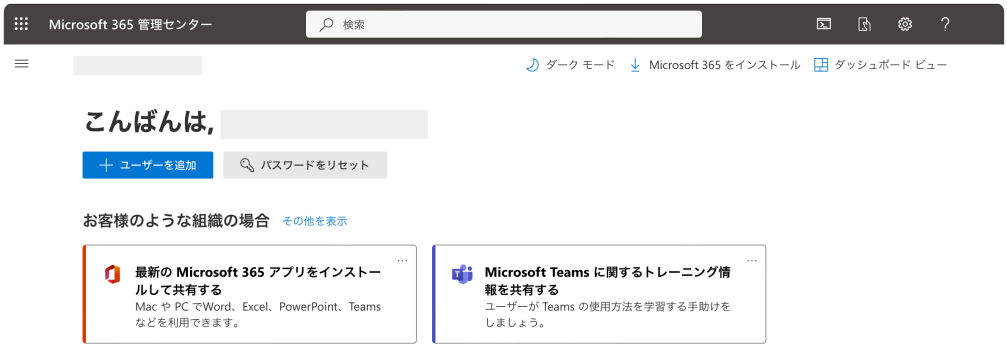
2-8. Display the client secret.

The value is the client secret, so please note it.

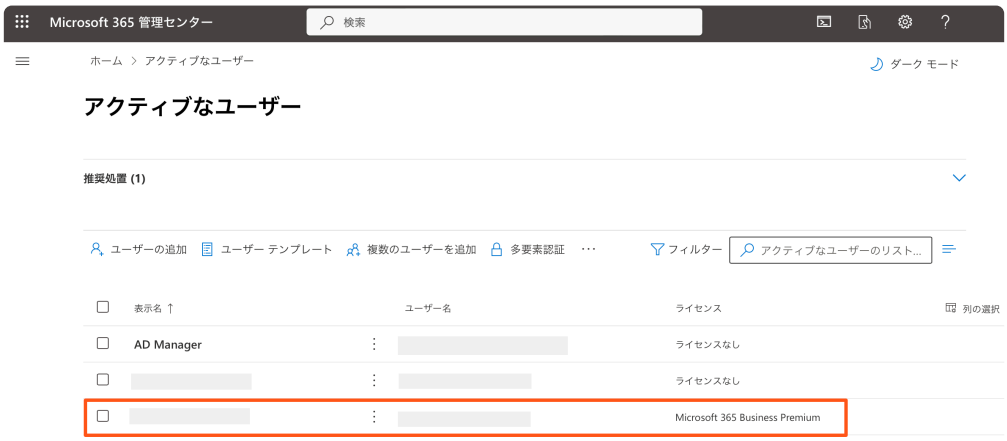
Now that we have the Client ID, Client Secret, and Redirect URI required for OAuth 2.0, the preliminary preparation is complete.

2-9. Enable authenticated SMTP settings in the Microsoft 365 management center.

Open [Microsoft 365 Management Center](#).



Select Active Users from the left menu.



When you select the target user, a panel will open on the right.

Select Manage Mail Apps in the Mail tab.

Please enable authenticated SMTP and save it. The set contents may not be reflected immediately, so if the same error continues to occur in sending mail, please try again after a while. If the error continues to occur, please check the settings again.

13.21. Gravo File Path Handling

Some component properties like FileWrite or SlackFilesUpload are making references to folders and file paths. On this page, we describe how files can be referenced in Gravio.

The Default Folder

All paths referenced on Gravio refer to the below paths on the respective operating systems:

Windows: `C:\ProgramData\HubKit\action\actmgr\data\`
Mac: `/Library/Application Support/HubKit/action/scripts/actmgr/data/`
Linux: `/var/opt/hubkit/action/scripts/actmgr/data/`
Linux with Docker (e.g. on a Raspberry Pi): `/home/gravio/hubkitrepo4/data/actmgr/data/` which is then mounted on `/var/opt/hubkit/actmgr/data` if you are accessing it from within Gravio.

Therefore you have 3 ways of making references to files:

1. Direct Reference

Mac and Linux:

File path	Description
`sample.txt`	Sample.txt in the `actmgr/data/` folder.
`./sample.txt`	Refers to sample.txt in the `actmgr/data/` folder.

Windows:

File path	Description
`sample.txt`	Refers to sample.txt in the `actmgr\data\` folder.
`.\\sample.txt`	Refers to sample.txt in the `actmgr\data\` folder.

2. Relative References

Mac and Linux:

File path	Description
`../../../../gravio/sample.txt`	Refers to `gravio/sample.txt` 4 levels up from the `actmgr/data/` folder.

Windows:

File path	Description

`..\\..\\..\\..\\gravio\\sample.txt`	Refers to `gravio\\sample.txt`, which is four levels up from the `actmgr\data\` folder.

3. Absolute References

Mac and Linux:

File path	Description
`/home/gravio/sample.txt`	Refers to `/home/gravio/sample.txt` under the root.

Windows:

File path	Description
`c:\\temp\\sample.txt`	Refers to `c:\temp\gravio\sample.txt` under c: drive.

In the example, it is assumed that the sample.txt file is readable by the user under which Gravio is operated.

Log Folders

Windows: `C:\ProgramData\HubKit\logs\`
Mac: `/Library/Logs/HubKit/`
Linux: `/var/log/hubkit/`

13.22. Automatic Update Manager Updates

The Update Manager installed on the Gravio Hub, both the Linux version and the ARM64 version (Raspberry), is set to automatically update every day at 2:00 AM, so no update operation is required. In addition, the Update Manager installed on the Gravio Hub will also automatically update whenever the Gravio Hub is restarted.

13.23. Hardware environment when using software sensors

1. Performance Guidelines When Using Inference Files

For a PC using an Intel i7 CPU, it has been verified to work with 2 ONVIF cameras set to save images at intervals of more than 10 seconds. This is not a guarantee that it will always work in your environment due to the amount of memory installed, whether the external storage device is an SSD, or I/O processing, but please use it as a guideline. Also, for PCs using the ATOM chipset, it has been verified to work with one camera set to save images at intervals of more than 20 seconds.

ONVIF compatible cameras are limited to devices that support **ONVIF Profile T**.

CPU/Chipset	Image save setting when using inference files with ONVIF camera (Guideline)
i7	2 cameras with an interval of more than 10 seconds
ATOM	1 camera with an interval of more than 20 seconds

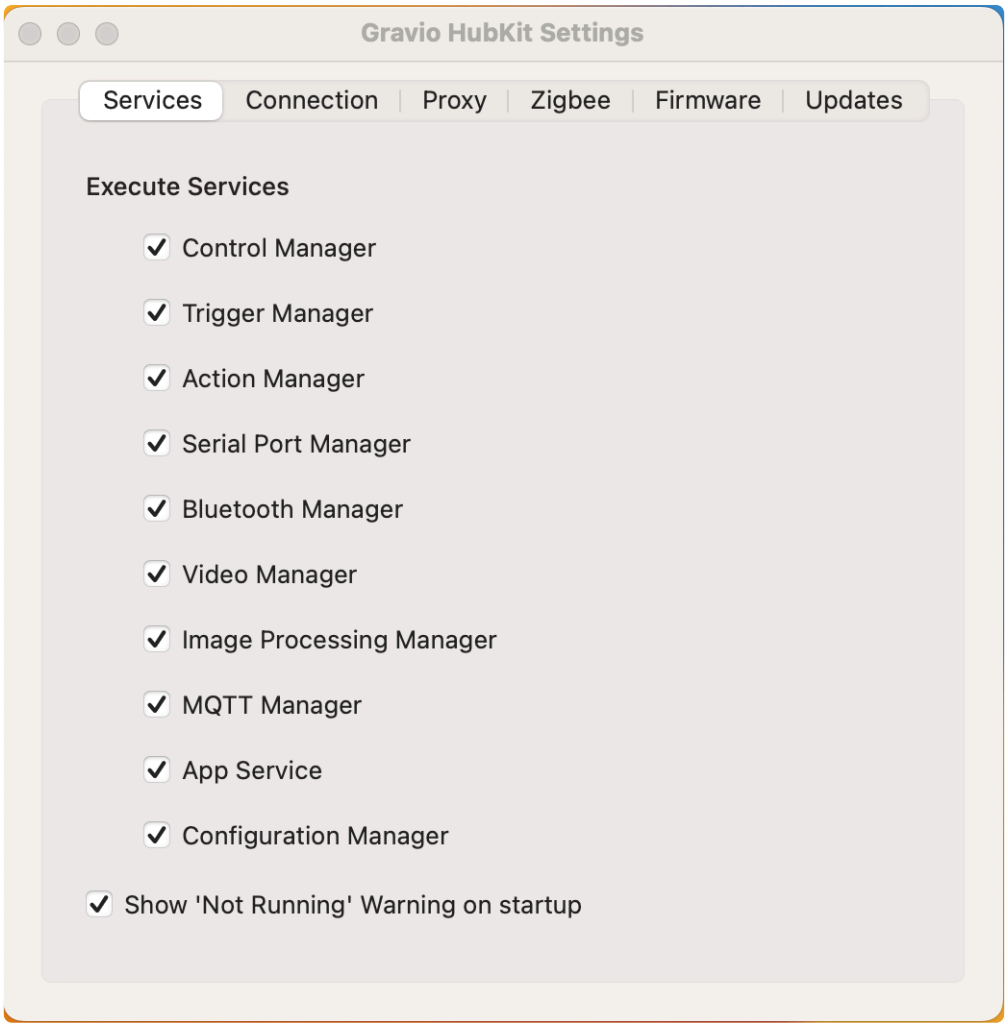
13.24. HubKit Settings

HubKit Configuration Guide

1. If you are using Mac

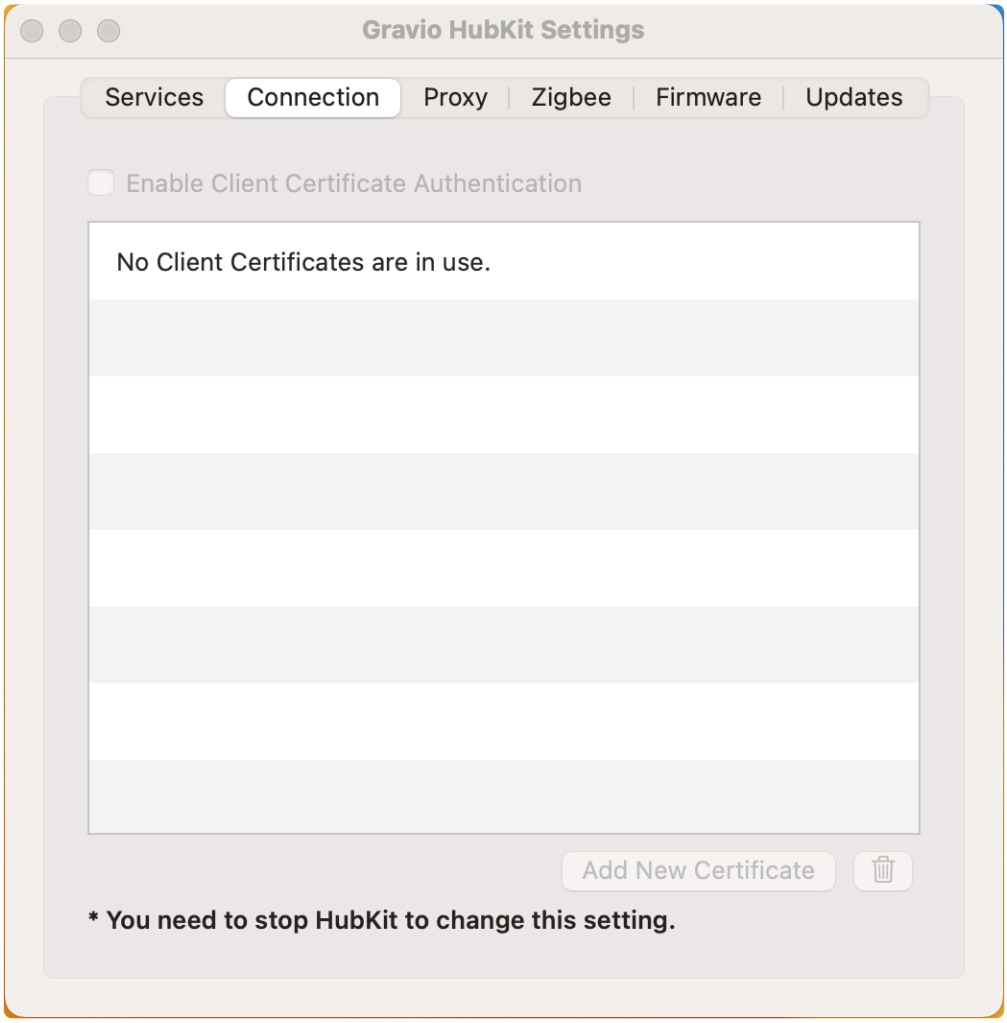
Select the ‘Settings’ menu from the HubKit icon in the top menu bar of macOS.

1. Services



Use this when you need to start or stop the services that make up the HubKit individually. If you want to display a warning when a service is stopped at the time of launching HubKit, select ‘Show Service Stop Warning at Startup’.

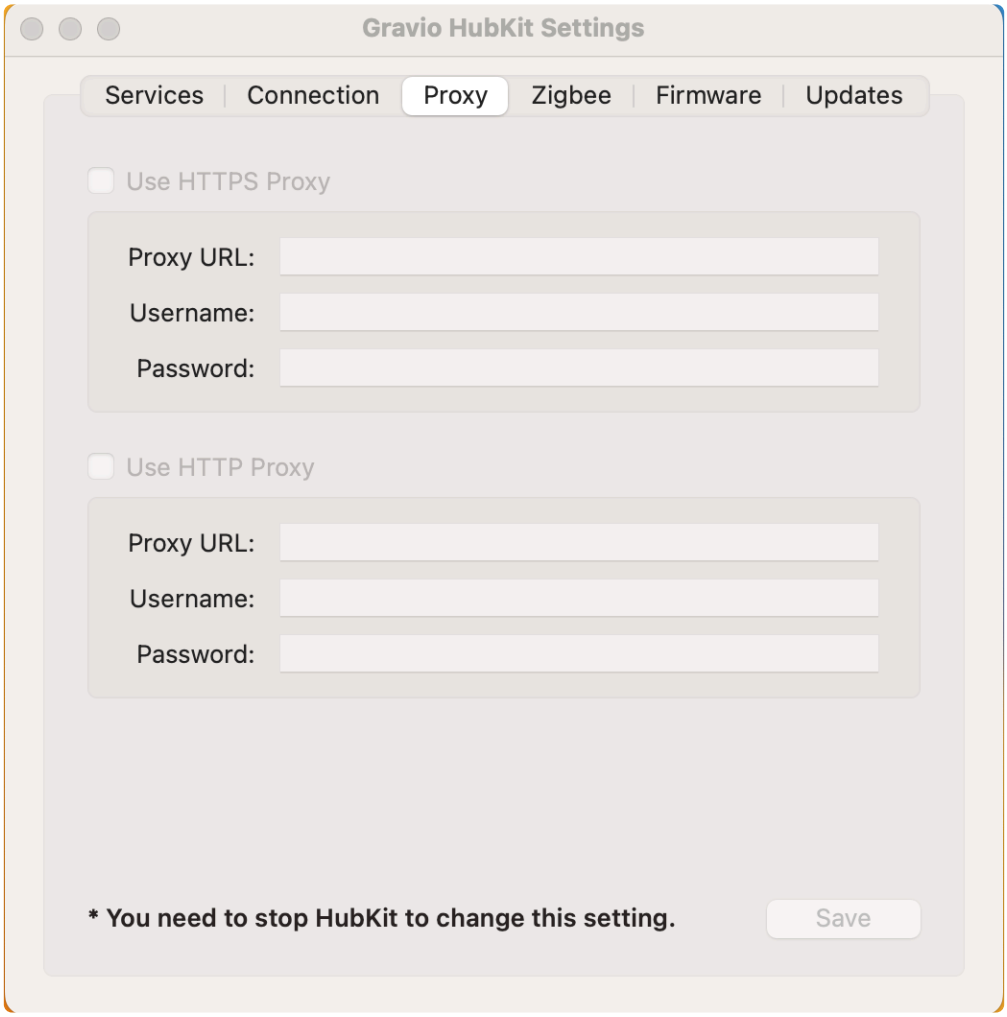
2. Connection



Select ‘Use Client Authentication’ if you want to restrict the Gravio Studio that connects to the HubKit. For instructions on how to install the certificate, refer to [If you want to restrict access from Gravio Studio to the hub](#).

*Please stop HubKit if it is running.

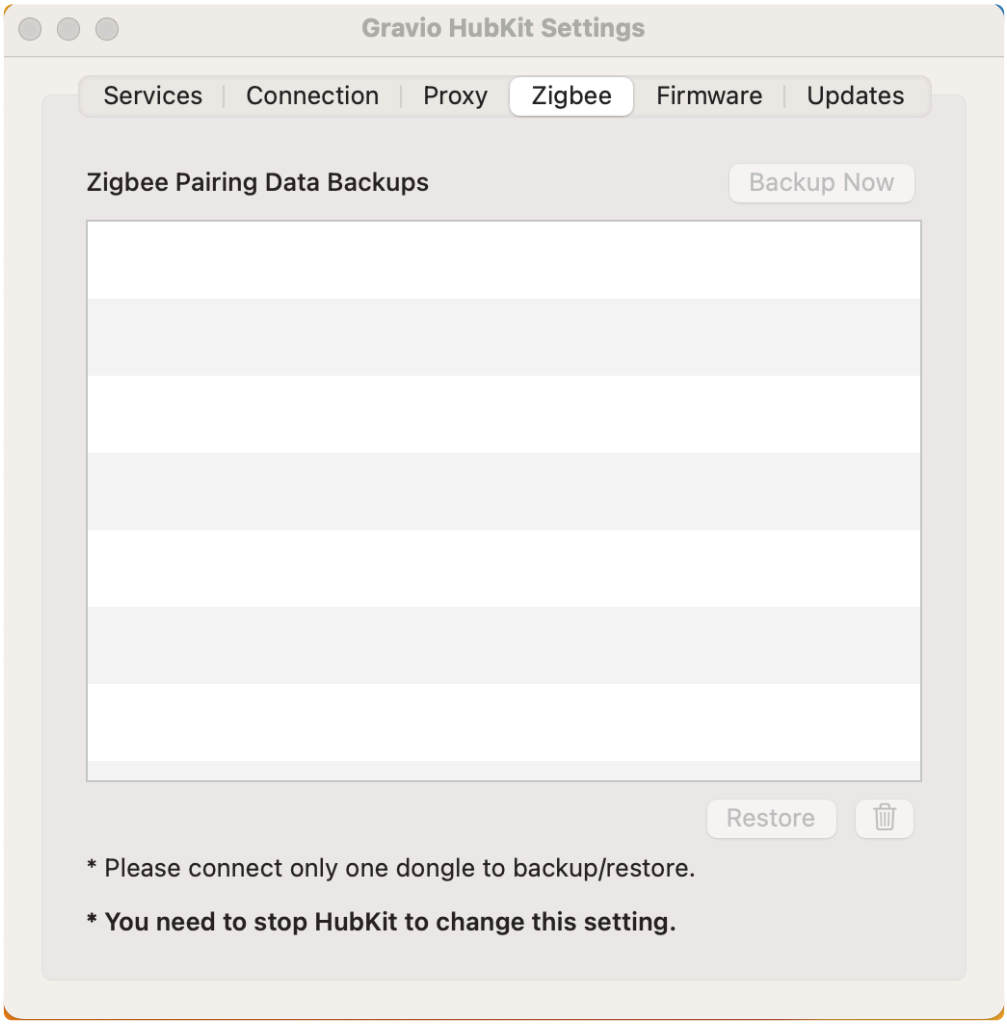
3. Proxy



Set this if you need to route HubKit through a proxy server when using internet services. Enter the account information (proxy URL, username, password) for the Proxy server to be used for each of the HTTPS and HTTP protocols.

*HubKit must be stopped to configure this setting.

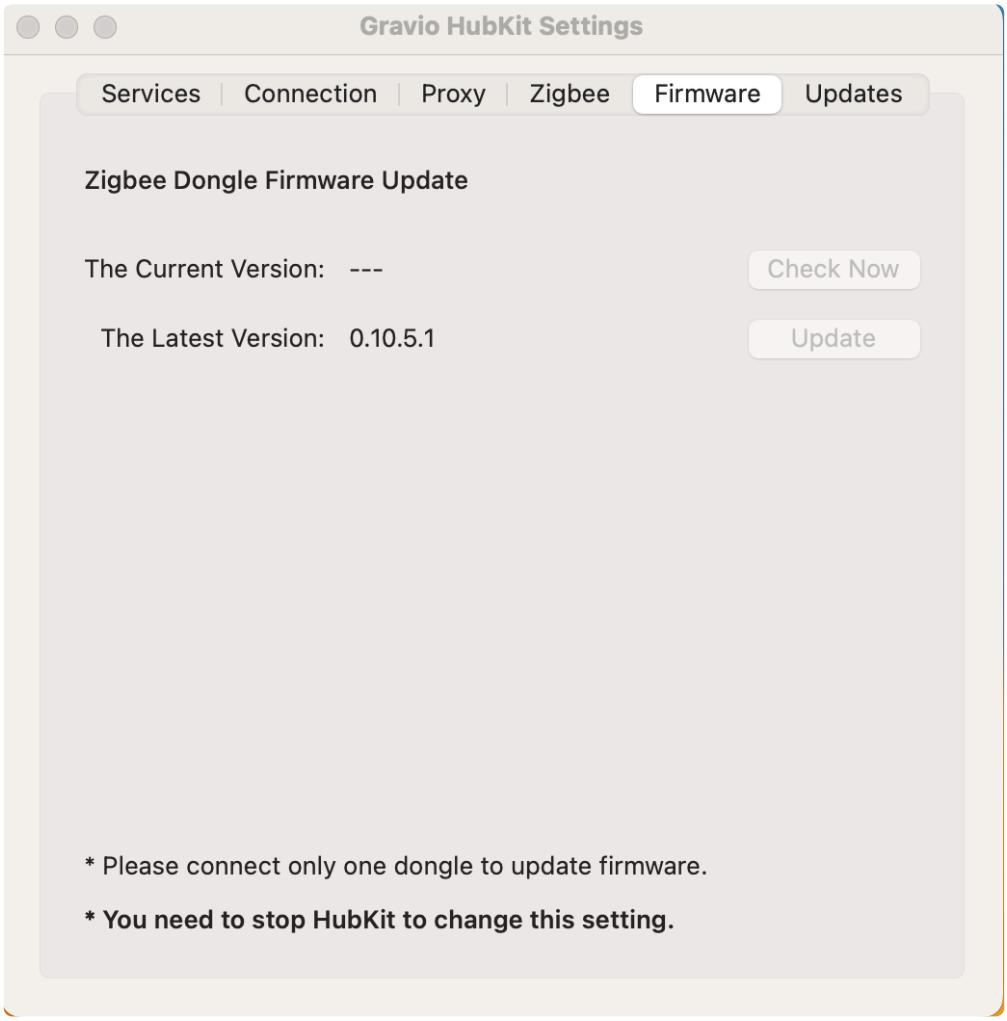
4. Zigbee



Press the ‘Backup Now’ button to backup pairing information from the Zigbee dongle’s sensor when you want to backup or transfer to another dongle. If you want to restore the backup pairing information, select the backup content and press the ‘Restore’ button.

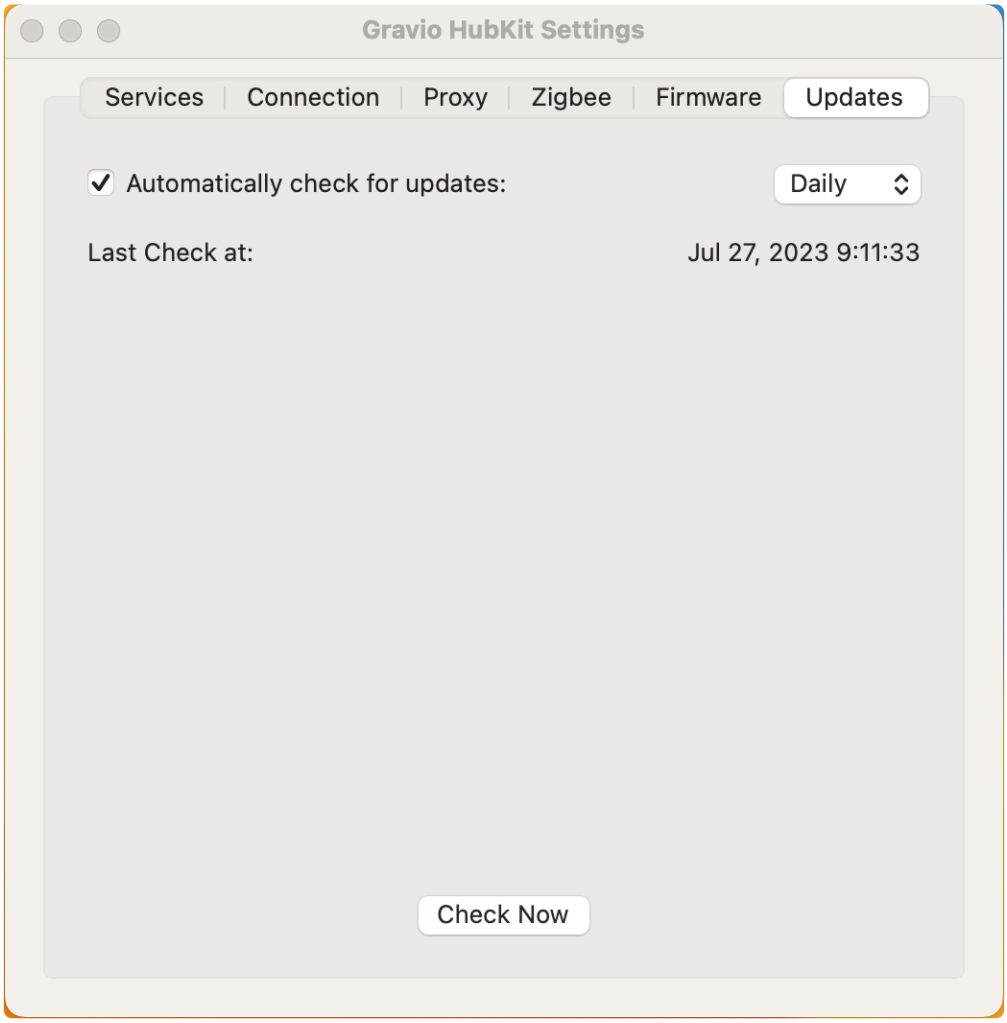
*Please stop HubKit if it is running.

5. Firmware



Refer to How to update Zigbee firmware for firmware updates.

6. Update



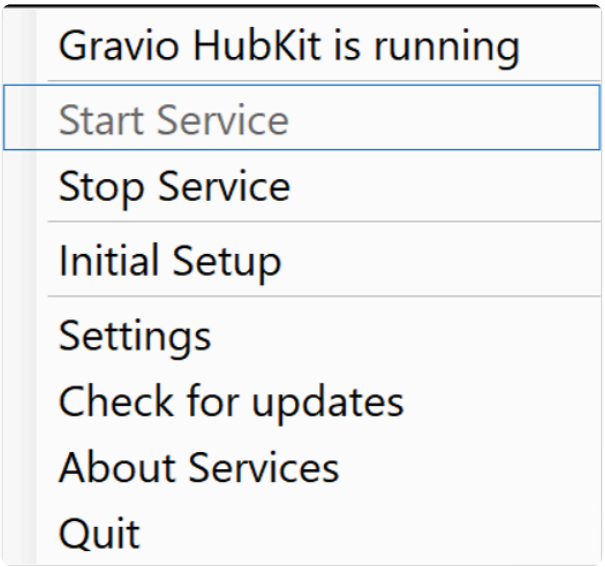
Set the timing to regularly check if there is a newer version of HubKit. Press the ‘Check Now’ button to check immediately for the latest version.

A confirmation dialog will be displayed if there is a newer version. Follow the menu to install if you choose to update.

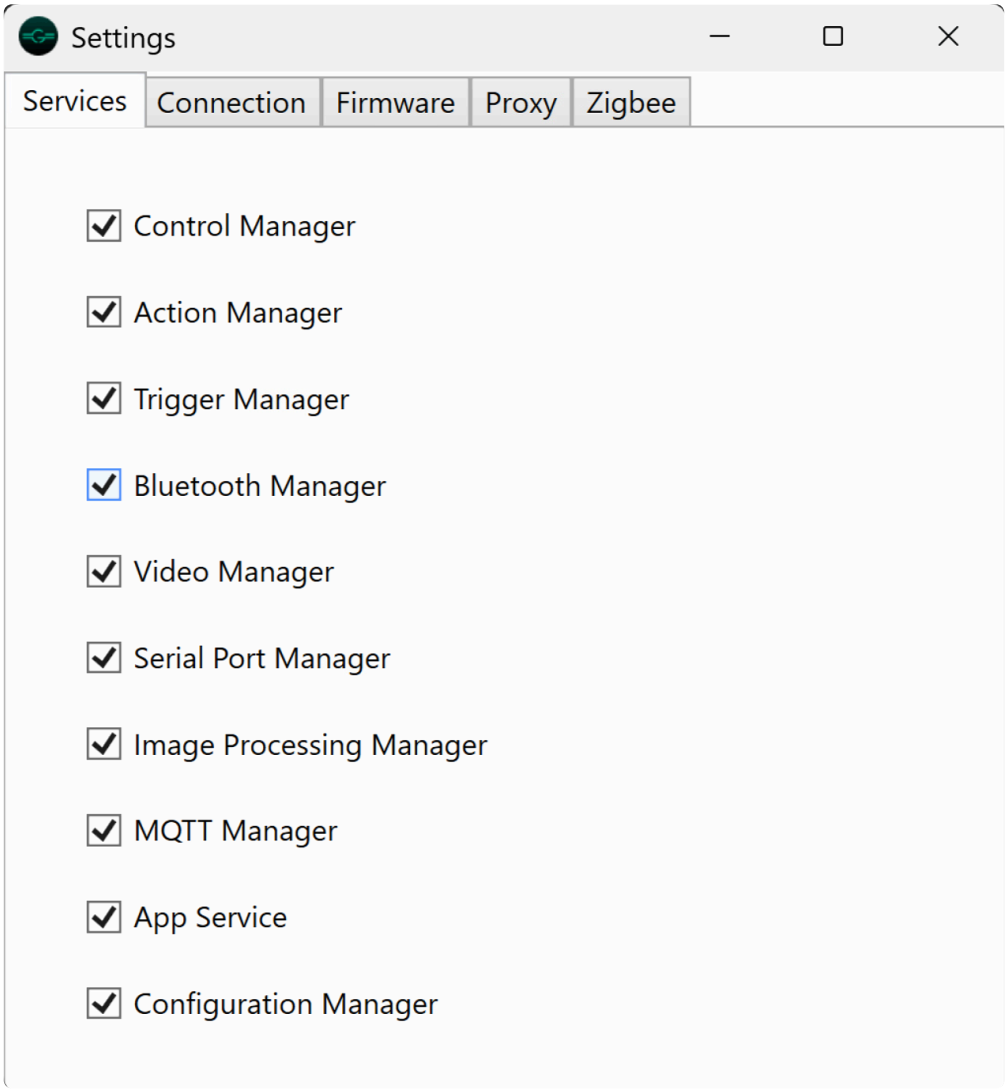
2. If you are using Windows

Execute ‘GravioTools4’ from the ‘GravioHubKit’ menu in the Windows10 start menu.

When GravioTools4 starts, an icon is displayed in the task bar. Right-click to display the menu and select the ‘Settings’ menu.

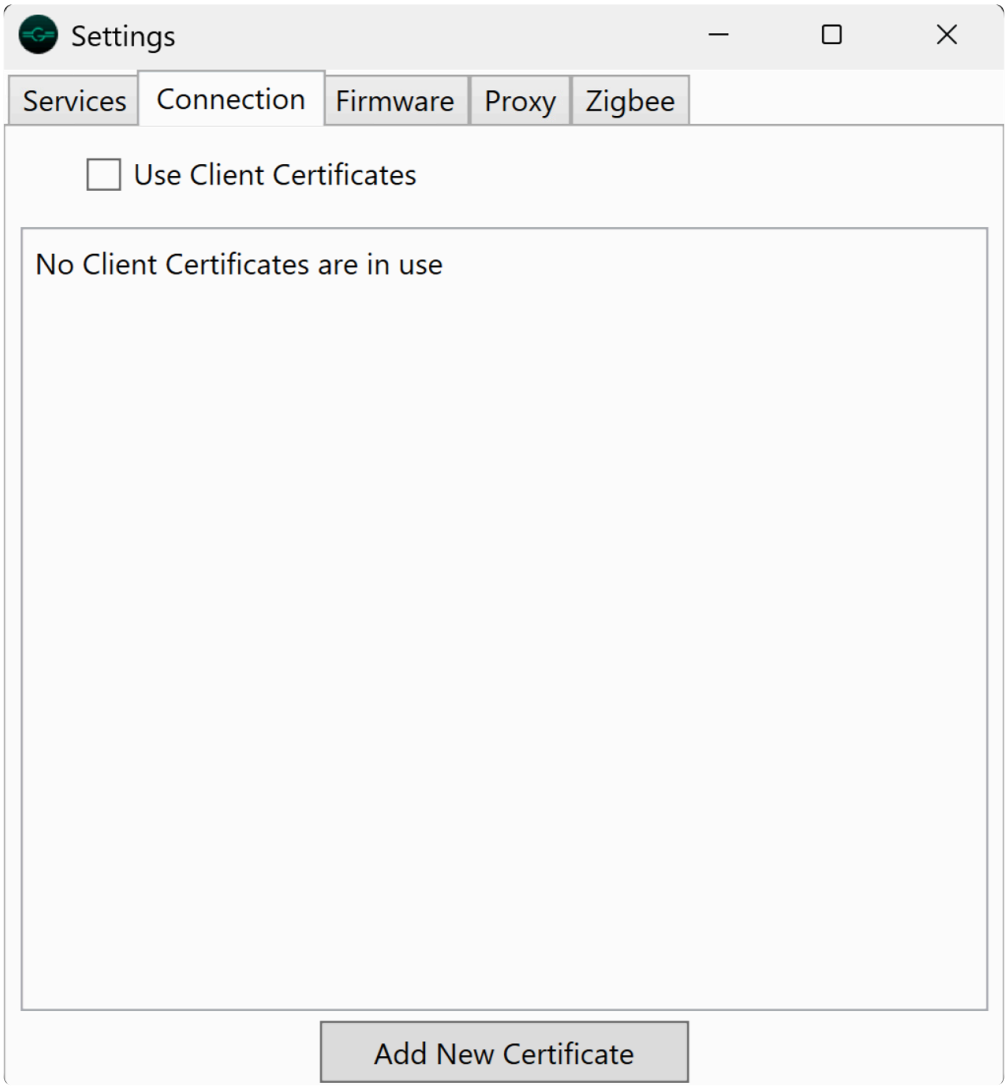


1. Services



Use this when you need to start or stop the services that make up the HubKit individually. If you want to display a warning when a service is stopped at the time of launching HubKit, select ‘Show Service Stop Warning at Startup’.

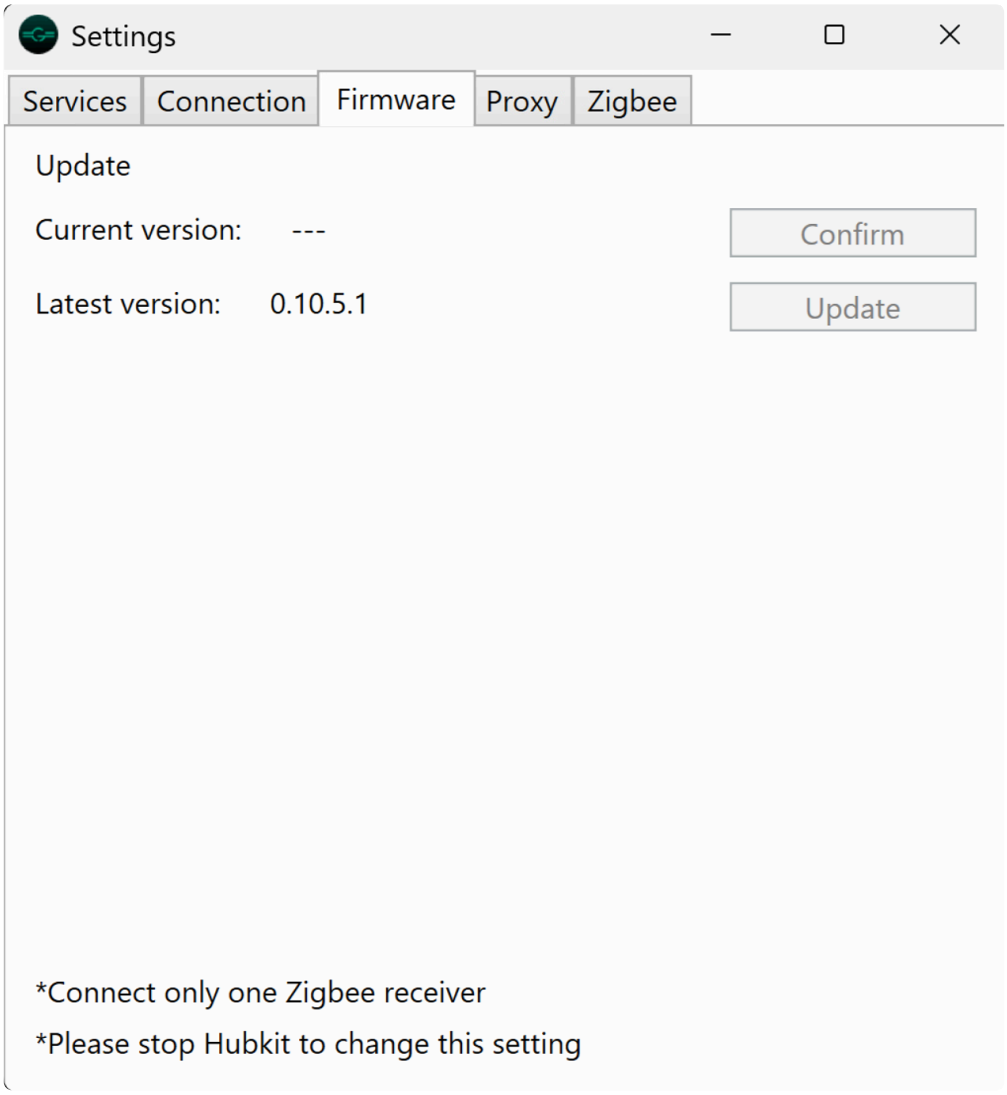
2. Connection



Select ‘Use Client Authentication’ if you want to restrict the Gravio Studio that connects to the HubKit. For instructions on how to install the certificate, refer to [If you want to restrict access from Gravio Studio to the hub](#).

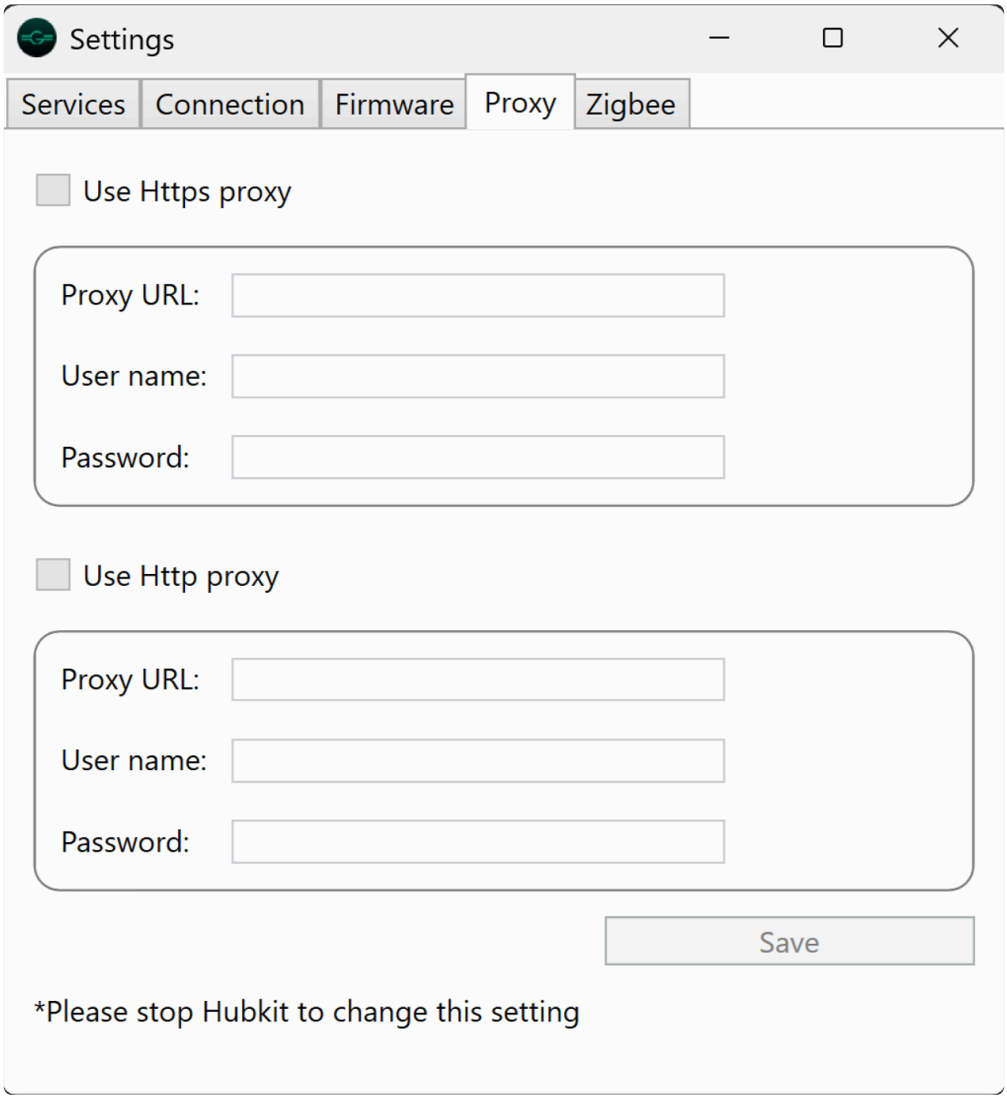
*Please stop HubKit if it is running.

3. Firmware



Refer to How to update Zigbee firmware for firmware updates.

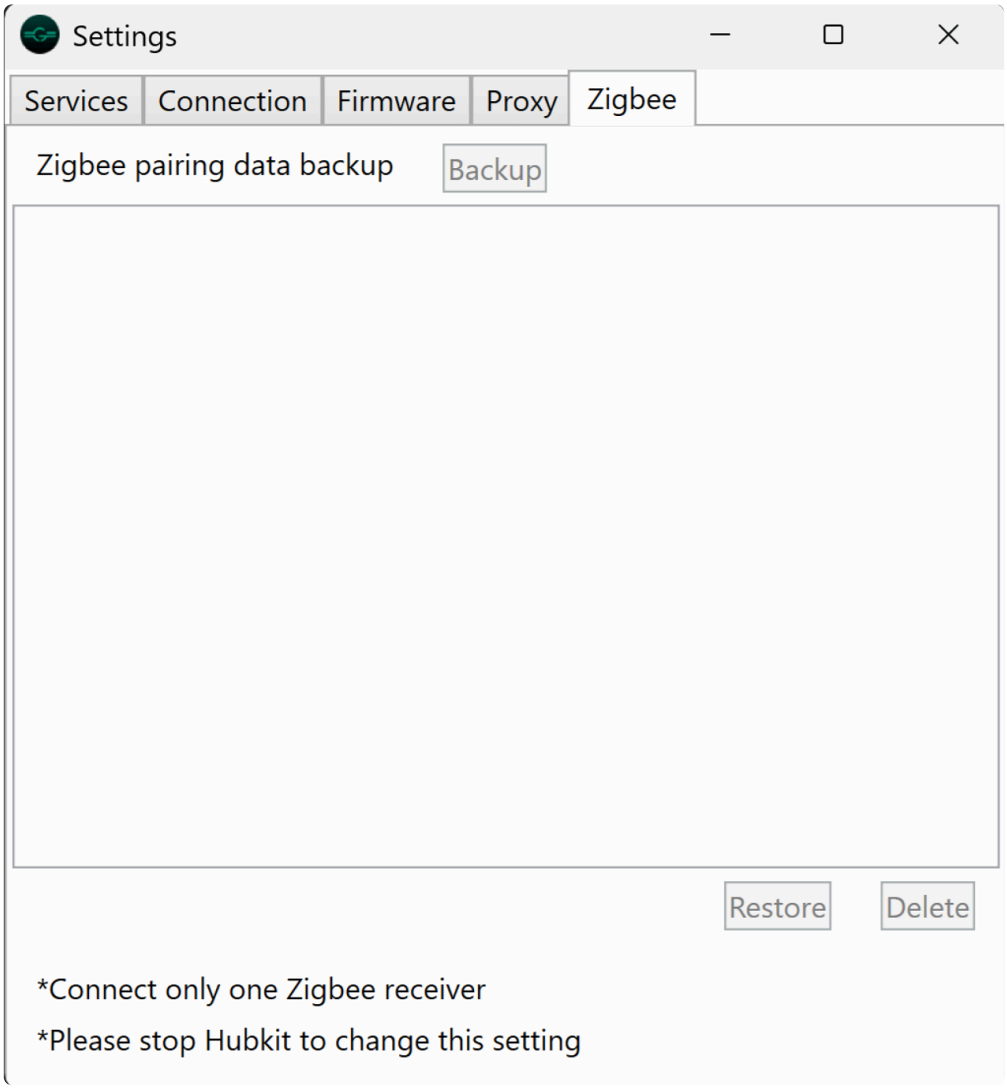
4. Proxy



Set this if you need to route HubKit through a proxy server when using internet services. Enter the account information (proxy URL, username, password) for the Proxy server to be used for each of the HTTPS and HTTP protocols.

*HubKit must be stopped to configure this setting.

5. Zigbee



Press the ‘Backup Now’ button to backup pairing information from the Zigbee dongle’s sensor when you want to backup or transfer to another dongle. If you want to restore the backup pairing information, select the backup content and press the ‘Restore’ button.

*Please stop HubKit if it is running.

6. Update

When GravioTools4 starts, an icon is displayed in the task bar. Right-click to display the menu and select ‘Download the latest version’.

A confirmation dialog will be displayed if there is a newer version. Follow the menu to install if you choose to update.

13.25. Credits and Attributions

Software Libraries Used

Under BSD-2-Clause License

- github.com/godbus/dbus
- github.com/pkg/errors

Under BSD-3-Clause License

- pkg.go.dev/golang.org/x/text
- pkg.go.dev/golang.org/x/net
- pkg.go.dev/golang.org/x/sys
- pkg.go.dev/golang.org/x/crypto
- pkg.go.dev/golang.org/x/image
- pkg.go.dev/golang.org/x/oauth2
- pkg.go.dev/golang.org/x/net
- github.com/muka/ble
- github.com/bugst/go-serial
- github.com/PaesslerAG/gval
- github.com/PaesslerAG/jsonpath
- github.com/generikvault/gvalstrings
- github.com/gorilla/mux

Under Apache 2.0 License

- github.com/coreos/go-systemd
- github.com/aws/aws-sdk-go
- github.com/hajimehoshi/go-mp3
- github.com/hajimehoshi/oto
- github.com/spf13/cobra
- github.com/tensorflow/tensorflow

Under MIT License

- github.com/fatih/structs
- github.com/mattn/go-colorable
- github.com/mattn/go-isatty
- github.com/mgutz/ansi
- github.com/mgutz/logxi
- github.com/sigurn/crc16
- github.com/sigurn/utils
- github.com/StackExchange/wmi
- github.com/go-ole/go-ole

- github.com/hashicorp/mdns
- github.com/rs/cors
- github.com/dgrijalva/jwt-go
- github.com/mattn/go-sqlite3
- github.com/montanaflynn/stats
- github.com/robfig/cron
- github.com/spf13/viper
- github.com/stretchr/testify
- github.com/jmoiron/sqlx
- github.com/mattn/go-pointer
- github.com/mattn/go-tflite
- github.com/satori/go.uuid
- github.com/sirupsen/logrus

Under Other Licenses

- LICENSE: github.com/clbanning/mxj
- LICENSE: github.com/eclipse/paho.mqtt.golang
- LICENSE: gopkg.in/check.v1
- LICENSE: gopkg.in/yaml.v3
- LICENSE: github.com/golang-migrate/migrate/
- LICENSE: github.com/golang/freetype
- LICENSE: github.com/shirou/gopsutil
- ISC LICENSE: github.com/nfnt/resize
- ISC LICENSE: github.com/davecgh/go-spew
- MPL-2.0 License: github.com/go-sql-driver/mysql
- Zlib License: github.com/kardianos/service

Credits for icons, images and other sources.

- Icons made by Freepik from www.flaticon.com is licensed by CC 3.0 BY

13.26. Gravio Apps Framework

Gravio will also provide a framework, that allows third party companies to build native mobile applications that integrate seamlessly into the Gravio Edge infrastructure. APIs provided by Gravio's edge servers can provide data or be triggered by an authenticated application.

Gravio apps open an entire new ecosystem of providers and developers who can build on top of an existing Gravio edge network.

Examples of Gravio Apps:

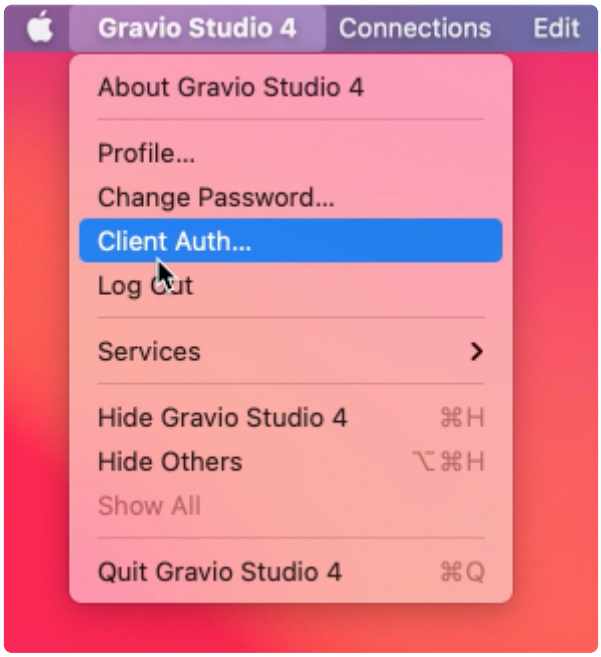
- A branded "ACME Industrial Maintenance" company application that can be used to check all of ACME's plants on site that are being monitored by Gravio Edge Servers and connected sensors.
- A branded "Joe's Restaurant Chain" application that allows its guests to use their phones to check queue lengths or restaurant occupancies in real-time.
- A branded "Farmer Co." application that allows farmers to retrieve sensor information out in the fields using their mobile phones.
- A city authority that can use mobile applications to retrieve information from local traffic monitoring and control systems.

The Kitchen Sink Gravio App will be available soon here. If you wish to be notified when it goes live, please write us an e-mail to support@gravio.com

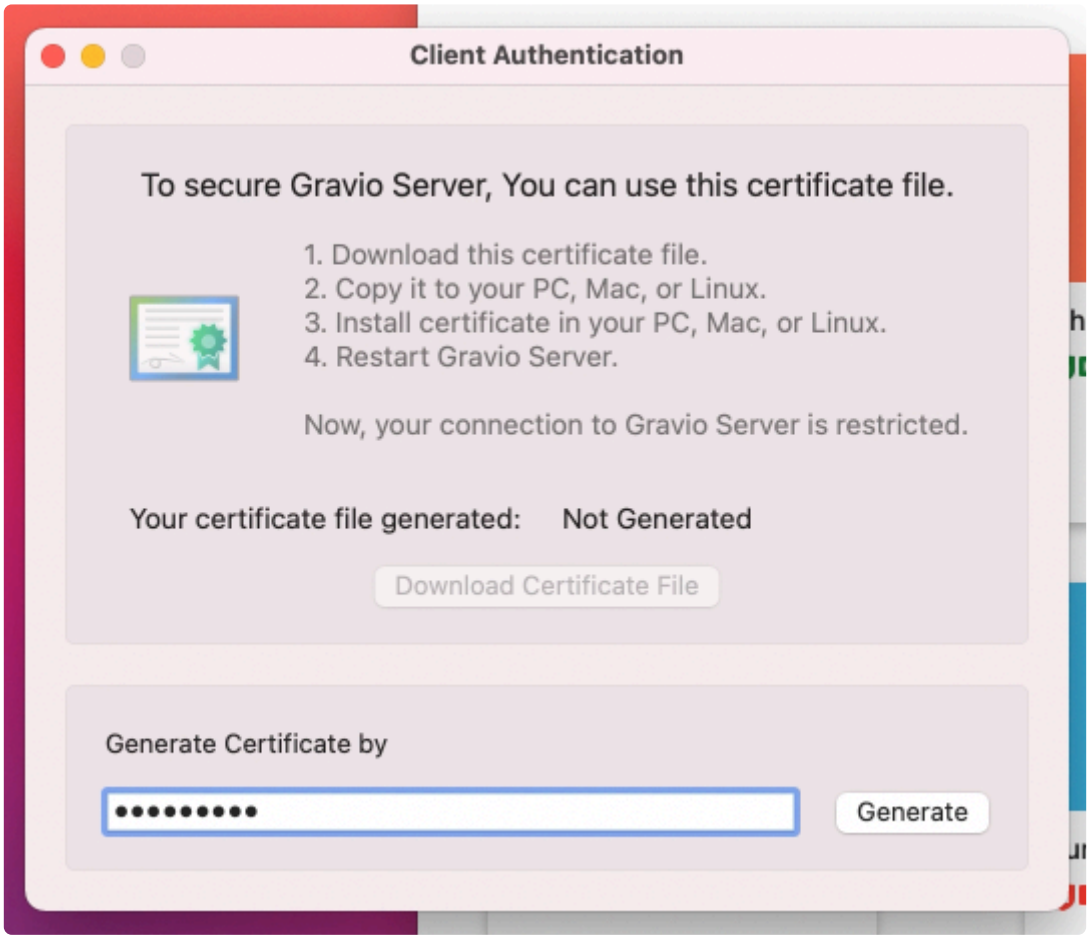
14. Security and Authentication

If you have Gravio Standard or better, you can secure your Gravio HubKit against your Gravio Studio machine using a Client Authentication method. This means you generate an authentication certificate from your Gravio Studio machine, which will be deployed on the Gravio HubKit server. Any Gravio HubKit server that has the client certificate authentication setting enabled will ensure that only Gravio Studios with the respective counterparts are allowed to access it.

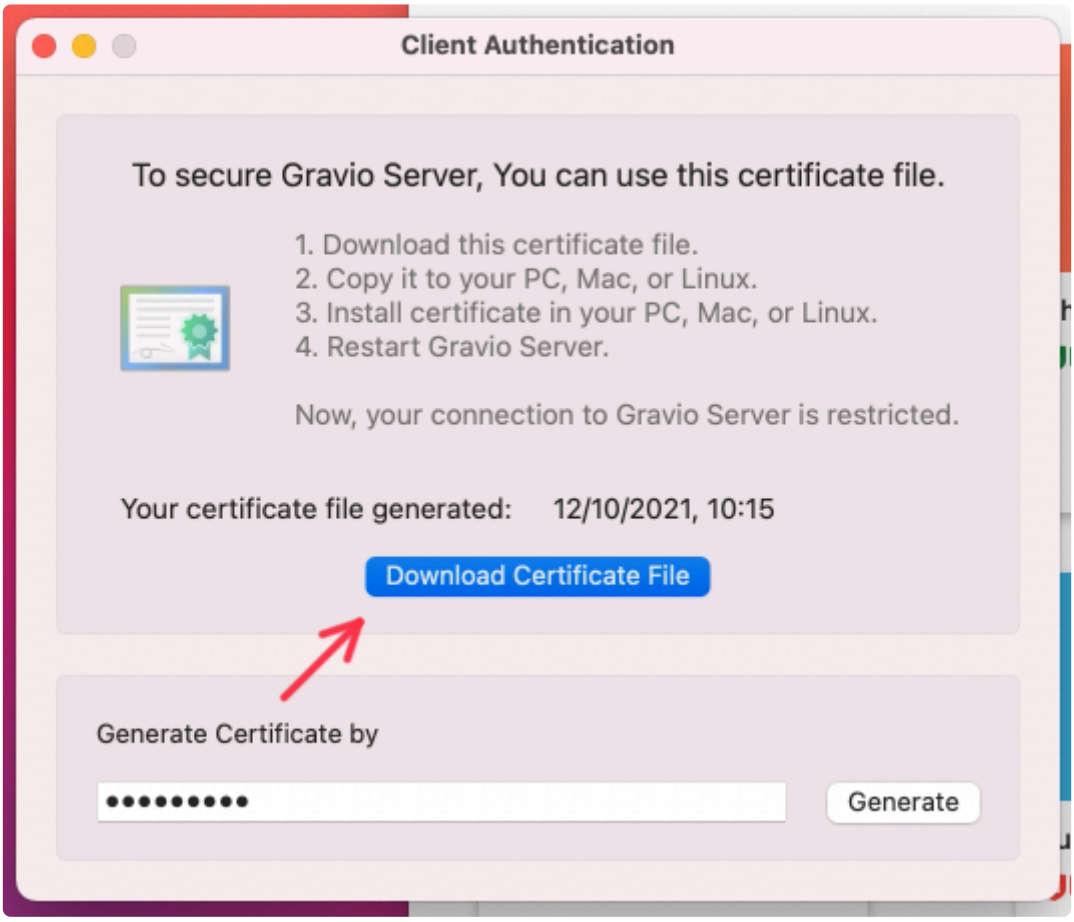
To create the certificate, open your Gravio Studio and generate the from the main menu:



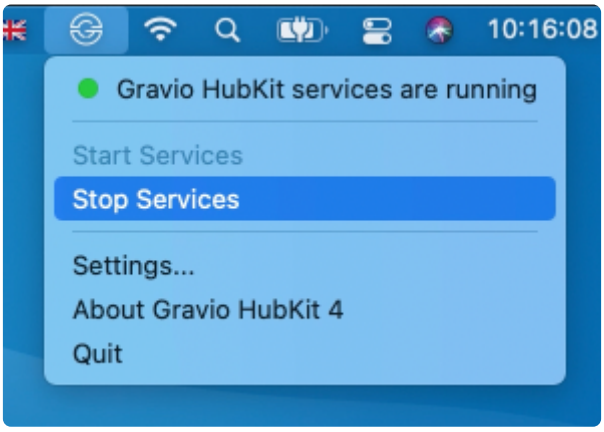
You will have to enter a key, based on which the certificate will be issued.



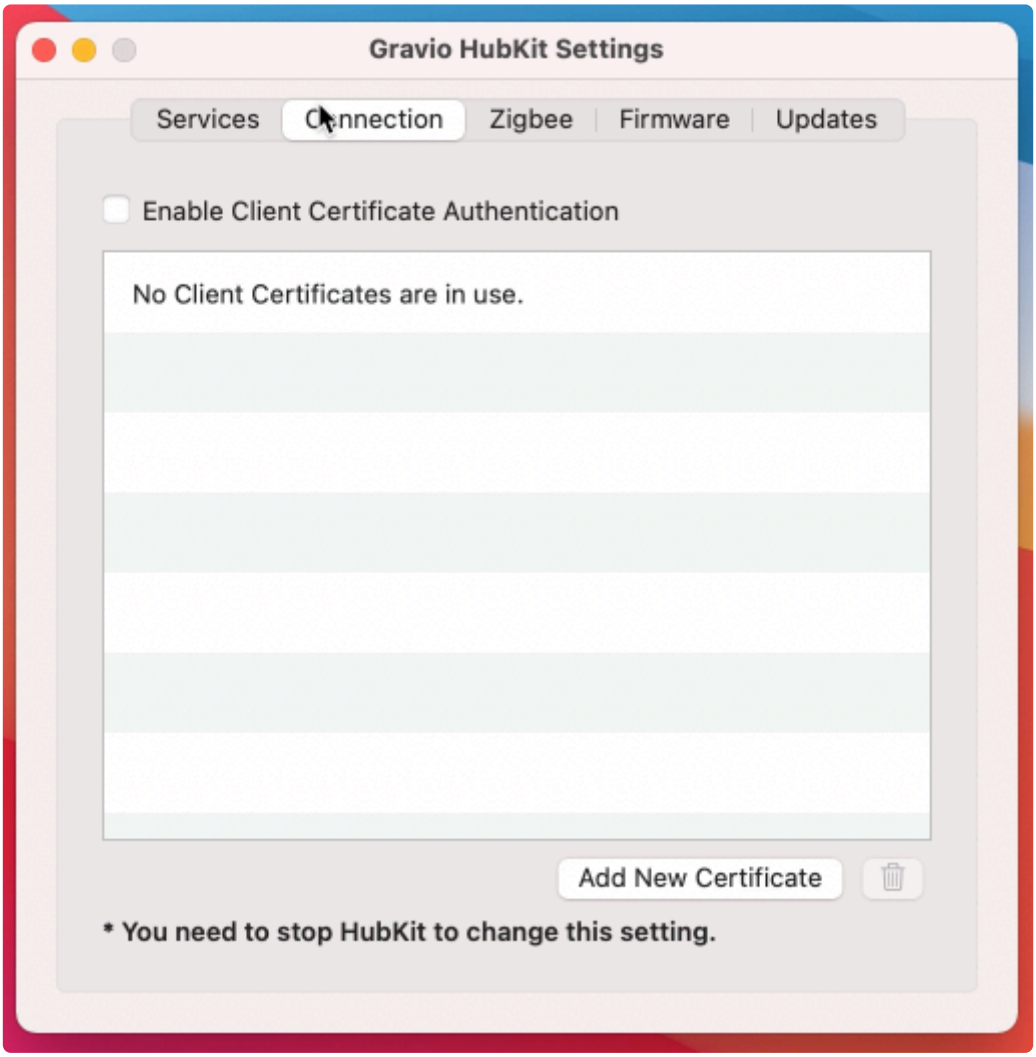
Click the “Generate” button. At this point, the Certificate matching this Gravio Studio is generated. You can download it to your computer, so you can deploy it on the respective Gravio HubKits.



In the “Settings” > “Connection” menu of HubKit, there is an option to upload the certificates. Note that you will have to stop the server in order to do so.

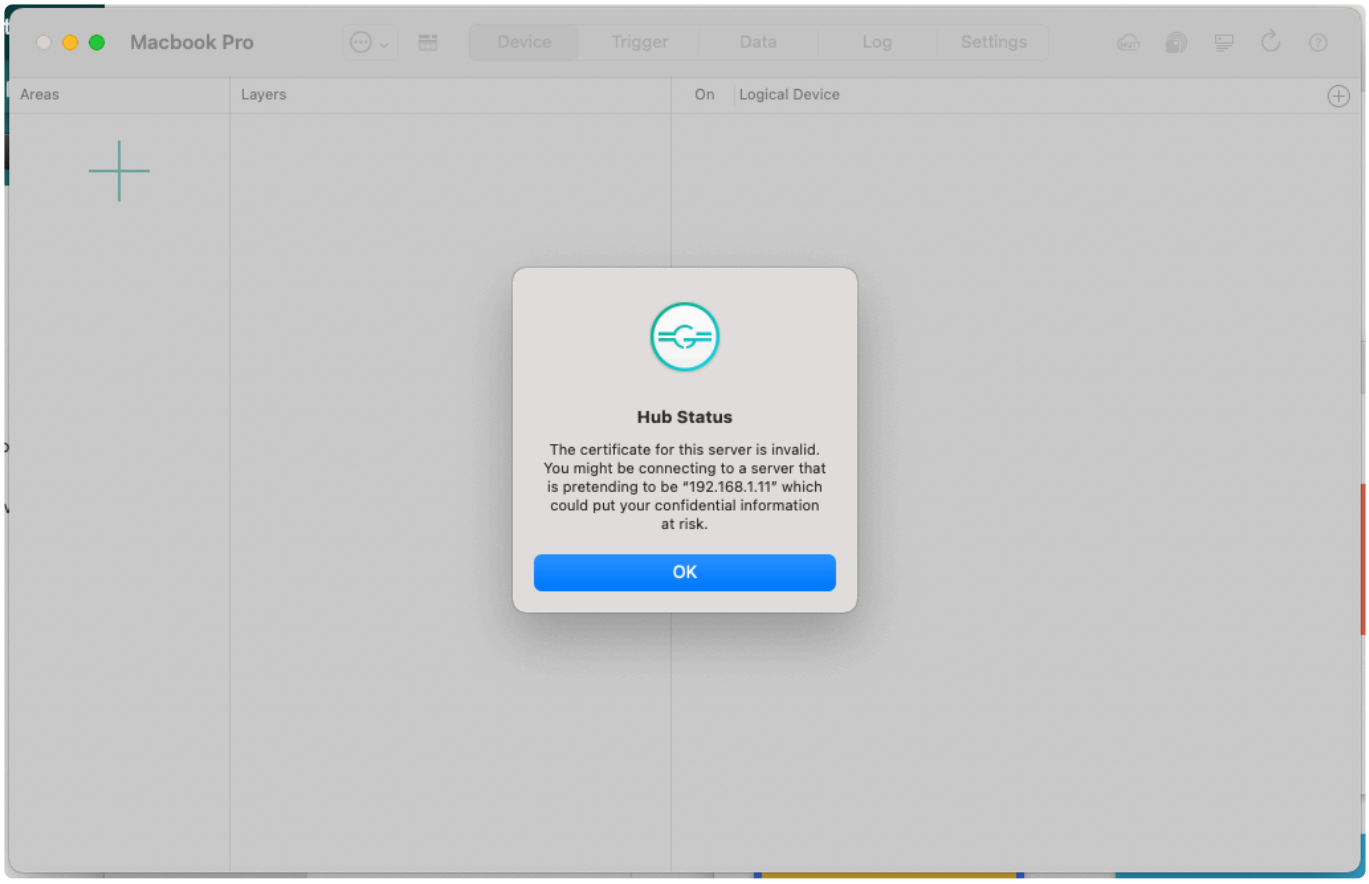


Click on “Settings…” again and enable the certificate authentication in the “Connections” tab. Upload the just downloaded certificate.



Your connection is now authenticated.

Note that if another Gravio Studio tries to access the HubKit now, you will get an error message similar to the following:


Macbook Pro
Device
Trigger
Data
Log
Settings
Areas
Layers
On
Logical Device
Hub Status
The certificate for this server is invalid. You might be connecting to a server that is pretending to be "192.168.1.11" which could put your confidential information at risk.
OK