Programming Guide

Hanwha Vision Open Platform

v5.00

2024-06-14

Contact Information

Hanwha Vision Co., Ltd.

Hanwha Vision 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, Korea

www.hanwhavision.com

Hanwha Vision America

500 Frank W. Burr Blvd. Suite 43 Teaneck, NJ 07666

hanwhavisionamerica.com

Hanwha Vision Europe

Heriot House, Heriot Road, Chertsey, Surrey, KT16 9DT, United Kingdom

hanwhavision.eu

Revision History

The table below provides the version information and revision history of this document.
Refer to 2.0x documents for checking revision history prior to version 3.00.

Version	Date	Description
5.00	2023-12-13	Add the MT8137
4.02	2023-12-16	Added and modified the docker command Modified the Troubleshooting contents (The Trouble Shooting document is newly created.)
4.01	2021-05-03	Added CLI Added Ubuntu 20.04 docker Unsupported Eclipse plugins
4.00	2020-08-31	Added a version 4.00 feature list
3.60	2019-12-20	Added an installation guide for plugin 64 bit
3.52	2019-09-02	Added a new version number [XML Editor] Added new platforms (T3000) on image
3.51	2019-06-10	Added a new version number
3.50	2018-10-29	[Installing Package] • Modified the version number of the Debian package [Creating a New Project] [XML Editor] • Added new platforms (XF8000, T4000) on image [Web Page Design] • Modified WEB UI on image [Device Preferences] • Deleted content
3.00_2nd	2017-12-08	Added supported Models • Thermal Models
3.00	2017-04-14	New revision

Table of Contents

1 Introduction to Hanwha Vision Open Platform

1.1 Overview

The Hanwha Vision Open Platform uses a standardized API to enable you to add additional functions to an IP camera without changing the existing software.

This platform allows you to download and install third-party applications on an IP camera to integrate an IP camera with other systems and use it for various purposes. For example, the end user can install a platform for an access control system, and the IP camera can be easily integrated with the access control system. Alternately, the end user can install an application that provides video analysis functions like people counting, which enables the use of an IP camera to control a system.

1.2 Key Features of Hanwha Vision Open Platform

The Hanwha Vision Open Platform SDK provides five key functions: real-time media streaming, video control, video analytics, recording, and video setup. In addition to these features, the SDK APIs provide several events and network functions. Refer to the SDK API document.

- Real-time media streaming
  - Open Platform SDK supports various media formats: raw video (yuv format), encoded video (MJPEG, H.264, H.265 format), and encoded audio (G.711 format).

    e.g., an application provides video analytics functions by using your unique analysis algorithm.

- Video control
  - By using video profile APIs, you can make an application that can change and control video. You can change the video source to NTSC or PAL format and add or delete a video profile according to their your own purposes.

    e.g., an application connects to VMS (Video Management System) and controls the video stream provided to end users.

- Video analytics

- Open Platform provides basic video analysis that is used by IP cameras: motion detection, face detection, appear/disappear, tampering event, intelligent video, enter/exit, and alarm event.

  e.g., an application triggers an alarm when an event has occurred.

- Recording

  - With the SDK APIs, you can store video streams on SD cards and NAS storage. End users can store the video streams on an SD card without an extra storage system and back them up easily.

    e.g., an application stores a video stream on an SD card when motion is detected.

- Video setup

  - You can set the video according to your preference. SDK APIs provide functions to set or get privacy mask for videos.

    e.g., you can store the video stream with a private mask on a SD card or in storage.

1.3 What's New in Hanwha Vision Open Platform SDK

New Features in v5.00

- Added a MT8137 toolchain

  - Added a toolchain to build a Q dual light camera

New Features in v4.02

- Command to run Docker

  - Modified the command to run Docker

  - Inserted a command for conveniently installing Debian

- Removed unsupported models

  - Removed models related to OP

- Created and registered new documents related to troubleshooting

New Features in v4.01

- Support docker

  - Added a Docker image file to provide a development environment.

- Add a **CLI** command

- Added CLI (Command line interface) to simplify project creation and setup and to disable Eclipse plugins.
· No more support for Eclipse plugins
  - Replaced Eclipse plugins with CLI.

New Features in v4.00

· Application signature
  - Added the application signature feature to prevent security risks, such as application tampering.
· SD card-dedicated use
  - Added an API that enables you to dedicate the usage of their SD cards to open platform applications only.
· Raw image pointer direct access (for WN7 or later)
  - Added a feature that enables you to directly access the raw image pointer so that you can use raw images with up to 4K resolution.

New Features in v3.60

· Combined compilation environment
  - Combined a 32-bit-based platform with a 64-bit-based platform to allow operation in the same environment.

New Features in v3.52

· Dynamic event
  - The application can send events for regular templates.
  - Refer to the SDKAPI document
· NAND flash space
  - Application can use NAND flash space.
  - Refer to the SDKAPI document
· Bug fixes
  - Fixed web functions
  - Fixed Eclipse plugin bugs (Install to Camera: port, file size)
· Internal modification
  - Synchronized with X series camera firmware

New Features in v3.51

- Internal modification

  - Synchronized with X series camera firmware

New Features in v3.50

- SUNAPI API

  - Added an API for using SUNAPI in the application

- RS-485 event

  - Added an event to transfer RS-485 data by using the terminal

- Proxy API

  - Added an API for setting proxy configuration

- Log event

  - Added output events to write the event log on camera

- CPU/Memory limit

  - Modified CPU/Memory usage limit

  - An event occurs when the system total usage is over 80% (Max 3 times).

- Fixed defective issues

  - Included improved performance caused by VOC and defective issues

- Upgraded Debian

  - Added sample applications on Debian (/opt/opensdk/opensdk-3.50/SampleApplication)

  - Upgraded to the latest version of libraries

New Features in v3.00

- Raw audio

  - Open Platform SDK supports raw audio

    e.g., an application provides audio analytics functions using your unique analysis algorithm.

- Application filename

  - The application filename can be changed after compiling

  - OriginalAppName_v{version}.cap

    e.g., ServerPushMJPEG.cap → ServerPushMJPEG_v170414.cap

- JAVA 8

  - Updated the development environment

- Eclipse Neon2

- Updated the development environment

1.4 Prerequisites

Open Platform Resources of WN5 Series

X series 2 mega (XNB-6000 and so on), 5 mega (XNB-8000 and so on)

	System TOTAL	Open Platform Usability Range
CPU	3040 DMIPS	Up to 80% of the system
RAM	352 MB	Up to 80% of the system
Flash	256 MB	20 MB

Open Platform Resources of S3L Series

Q series 2 mega (QNO-6082R and so on), 5 mega (QNO-8080R and so on)

	System TOTAL	Open Platform Usability Range
CPU	2040 DMIPS	Up to 80% of the system
RAM	225 MB	Up to 80% of the system
Flash	256 MB	8 MB

Open Platform Resources of CV2x Series

P series AI camera (PNO-A9081R and so on)

	System TOTAL	Open Platform Usability Range
CPU	9273.6 DMIPS	Up to 80% of the system
RAM	4 GB	Up to 80% of the system
Flash	512 MB	50 MB

Open Platform Resources of CV22S66 Series

LPR camera (TNO-7180R and so on)

	System TOTAL	Open Platform Usability Range
CPU	9273.6 DMIPS	Up to 80% of the system
RAM	2 GB	Up to 80% of the system
Flash	512 MByte	183 Mbyte

Open Platform Resources of WN7 Series

X series 4K camera (XNV-9082R and so on)

	System TOTAL	Open Platform Usability Range
CPU	3420 DMIPS	Up to 80% of the system
RAM	4 GB	Up to 80% of the system
Flash	512 MB	50 MB

Open Platform Resources of CV2x 2M Series

P series AI 2M camera (PNO-A6081R and so on)

	System TOTAL	Open Platform Usability Range
CPU	9273.6 DMIPS	Up to 80% of the system
RAM	4 GB	Up to 80% of the system
Flash	512 MB	150 MB

Most SDK Hanwha Vision cameras support Hanwha Vision Open Platform.

Refer to the Hanwha_opensdk_camera.pdf file on our STEP site to check the SOC information, the supported SDK version, and the firmware version by camera models.

- https://step.hanwhavision.com

Open Platform Resources of MT8137 Series

Q dual light series (QNE-C9013RL and so on)

	System TOTAL	Open Platform Usability Range
CPU	3420 DMIPS	Up to 90% of the system
RAM	2 GB	Up to 90% of the system
Flash	1 G**B**	50 MB

Most SDK Hanwha Vision cameras support Hanwha Vision Open Platform.

Refer to the Hanwha_opensdk_camera.pdf file on our STEP site to check the SOC information, the supported SDK version, and the firmware version by camera models.

2 Developing an Application

2.1 Prerequisites

To develop a Hanwha Vision Open Platform application, prepare the following:

- Operating system: Linux or Windows
- Docker
- Code development editor

Note

Hanwha Vision Open Platform version 4.01 or higher supports application development using Docker. For information on versions earlier than version 4.01, visit our STEP site at https://step.hanwhavision.com.

2.2 Installing the Open Platform SDK

Running Docker Images (Linux, Mac)

Note

To run Docker images, install Docker first. For installing Docker, visit the site at https://docs.docker.com.

Follow the steps below:

Step 1.   Access the Linux server with the sudo privilege (administrator privilege).

```
<user_id>@ubuntu# sudo su
```

Step 2.   Unzip the Opensdk_Docker_Image_5.00_R#_byJenkins.tar.gz. in the workspace. And load the compressed file as a Docker image.

```
root@ubuntu:/home/<user_id># sudo tar zxvf
Opensdk_Docker_Image_5.00_R#_byJenkins.tar.gz
root@ubuntu:/home/<user_id># docker load -i wop_<version>.tar
```

Step 3.   Check the loaded image.

```
root@ubuntu:/home/<user_id># docker images
```

Step 4.   Run the image as a container to access the Hanwha Vision Open Platform5.00 environment. If the shared folder locates "/mnt", run the command below:

```
root@ubuntu:/home/<user_id># docker run --rm -it --volume="$PWD:/mnt" wop:5.00
/bin/bash


If you use Window Terminal, run the command below:
C:Opensdk_Docker_Image_5.00_R15_byJenkins> docker run --rm -it --
volume="$(pwd):/mnt" wop:5.00 /bin/bash
```

Step 5.   Exit the container.

```
root@ubuntu:/home/<user_id># exit
```

Running Docker Images (Windows)

> **Note**
>
> To run Docker images, install Docker Desktop first.

Follow the steps below:

Step 1.   Run Docker Desktop Installer.exe to install Docker Desktop.

Step 2.   Make sure you don't select [Use the WSL 2-based engine] in order not to use WSL2.

Step 3.   After running Docker Desktop, run Windows PowerShell as an administrator.

Step 4.   Follow steps 2 to 5 of "Running Docker Images (Linux, Mac)".

Installing the Debian Package (Linux)

Follow the steps below:

Step 1.   Get the SDK package "wisenet_opensdk-5.00.deb".

```
 <user_id>@ubuntu:/home/<user_id>#  sudo    -s
root@ubuntu:/home/<user_id>#  apt-get update
root@ubuntu:/home/<user_id>#  apt-get    install    curl
root@ubuntu:/home/<user_id>#  apt-get    install    libssl-dev
root@ubuntu:/home/<user_id>#  apt-get    install    libtar-dev
root@ubuntu:/home/<user_id>#  apt-get    install    libssl-ocaml
root@ubuntu:/home/<user_id>#  apt-get    install    libc6-dev
root@ubuntu:/home/<user_id>#  apt-get    install  build-essential
root@ubuntu:/home/<user_id>#  apt-get    install  libcurl4-gnutls-dev
```

Step 2.   Install third-party packages as required by "wisenet_opensdk-5.00.deb". Go to the directory where the SDK package resides.

Step 3.   Install the SDK package by running the command below:

```
root@ubuntu:/home/<user_id># dpkg  -i   wisenet_opensdk-<version>.deb
```

Step 4.   Log out and log in again.

2.3 CLI Guide

opensdk_new_project

Create a new template code.

```
opensdk_new_project –n ${app_name} –v ${app_version} –s ${sdk_version} –p
${platform}
```

Example:

```
~# opensdk_new_project -n WOP -v 1.0 -s 5.00 -p WN7
```

If you want to add multiple platforms, add as many -p options as you want.

Example:

```
~# opensdk_new_project -n WOP -v 1.0 -s 5.00 -p WN7 –p CV2X
```

opensdk_install

Install the application on the camera.

```
opensdk_install -a ${app_name} -i ${camera_ip} -u ${camera_id} -w
${camera_password}
```

Example:

```
~# opensdk_install -a WOP -i 192.168.1.1 -u admin -w password123!
```

opensdk_check_platform

Check the camera's platform.

```
opensdk_check_platform -m ${camera_model_name}
```

Example:

```
~# opensdk_check_platform -m XNV-9082R
```

opensdk_packager

Compress and encrypt the application files.

```
opensdk_packager
```

Example:

```
~# make clean; make; opensdk_packager
```

2.4 Folder and File Structure

The folder and file structures of the application are as follows:

```
├──Includes
├──bin
├──html
│   ├──css
│   ├──fonts
│   ├──images
│   ├──styles
│   └──index.xml
├──inc
├──res
│   └──images
├──src
│   └──[App_name].cpp
├──IpCameraManifest.xml
│   ├──Application Overview
│   │         ├──Application Details
│   │         ├──Supported Platforms
│   │         └──Application Configuration
│   └──Application Settings
│             ├──Input Events
│             └──Permission
├──Makefile
├──libs
└──App_name.cap
```

Includes

Contains the paths to header files and toolchains required to build a project.

bin

Contains bin files generated when building a project.

html

Contains images, css, fonts, styles, and index.html files to configure the web pages.

inc

Contains header files.

res

Contains image files used in applications.

src

[App_name].cpp – defines application operations, receives data and events, and sends responses accordingly.

IpCameraManifest.xml

・ Application overview

- Application details: Overall information of the application (application name, application location, version, vendor, description, target SDK, min SDK required, and max SDK)

- Supported platforms: SOC information of cameras to be installed in the application (you can make changes)

- Application configuration: Key and Value information (you can make changes)

・ Application settings

- Input events: Determines whether to receive information such as encoded video, raw video, raw audio, and VA events.

- Permission: Determines whether to allow PTZ, Device, and SD card access.

libs

Contains library files.

Makefile

Builds codes to create binary files.

App_name.cap

The executable file to be installed on the camera created via the **opensdk_packager** command.

2.5 Creating an Application Using CLI

The following is an example of creating an application named "Wisenet", building it, and installing it on an XNV-9082R camera.

> **Note**
>
> Check the Hanwha Vision Open Platform version supported by the camera and use the SDK of the applicable version. The **CGI** command to check the Hanwha Vision Open Platform version is as follows:
>
> http://<camera IP>/stw-cgi/system.cgi?msubmenu=deviceinfo&action=view

Follow the steps below:

Step 1. Run the **CGI** command to check the Hanwha Vision Open Platform version supported by the XNV-9082R camera model.

```
http://192.168.1.100/stw-cgi/system.cgi?msubmenu=deviceinfo&action=view
```

You will receive the following response saying: "You should use Hanwha Vision Open Platform SDK 5.00 for application development."

```
Model=TNM-C4940TDR
...
OpenSDKVersion=5.00_230629
```

Step 2. From the Linux editor, check the platform of the TNM-C4940TDRcamera model in the document (Hanwha_opensdk_camera_XXXXXXX.pdf)

Step 3. Create a template code to develop the version 1.0 application named "Wisenet" under the SDK version 5.00. The Wisenet application will be used on the camera developed under the WN7 platform.

```
# opensdk_new_project -n wisenet -v 1.0 -s 5.00 -p WN7
```

Step 4. Go to the folder that contains the application.

Step 5. Build, encrypt, and package the **# cd Wisenet** application.

```
wisenet# make clean;make;opensdk_packager
```

Folders and their structures are as follows: For a detailed description of each folder and file, refer to "2.4 Folder and File Structure."

```
├──Includes
├──bin
├──html
│    ├──css
│    ├──fonts
│    ├──images
│    ├──styles
│    └──index.xml
├──inc
```

```
├──res
│    └──images
├──src
│    └──[App_name].cpp
├──IpCameraManifest.xml
│    ├──Application Overview
│    │              ├──Application Details
│    │              ├──Supported Platforms
│    │              └──Application Configuration
│    └──Application Settings
│                   ├──Input Events
│                   └──Permission
├──Makefile
└──App_name.cap
```

Caution

For a platform that requires a certificate, if it is not located under "/opt/opensdk/signature," the application won't be installed on the camera correctly.

Issuing a certificate

- Download a test key and a test certificate (valid for 3 months) from our STEP site during the application development phase.
- After registering an application you've developed on our STEP site, request a signature key and a certificate for it.
    A. Request a signature key and certificate on our STEP site. You will receive them via email.
    B. Save the signature key and certificate received from Hanwha Vision under the following directory on your development PC:
        "/opt/opensdk/signature"
    C. Execute opensdk_packager, and then an electronically signed package file will be created.
    D. When installing an open platform application on a camera, the camera software checks the signature and certificate of the application and only installs applications free of forgery.

Step 6.   Install the Wisenet application on the camera with the IP address of 192.168.1.100, provided that the camera access ID and password are admin and test123456 respectively.
For how to install an application from the camera web viewer, refer to "4.2 Installing an Application to the IP Camera."

```
wisenet# opensdk_install -a wisenet -i 192.168.1.100 -u admin -w test123456
```

3 Debugging

A Remote Debug viewer is a command-line application used to view logs from a third-party application. Execute this application, included in the SDK package, from the Linux command prompt.

RemoteDebugViewer is located in /opt/opensdk/opensdk-<version>/common/bin.

Or, you can find it easily by running the command below:

```
# which RemoteDebugViewer
```

3.1 Usage of Remote Debug Viewer

Use the following API in a third-party application to get logs from the Remote Debug viewer:

```
void debug_message(const char *fmt, ...);
```

Input arguments for the Remote Debug viewer:

`-i` IP address of IP camera

`-p` HTTP port number on IP camera

`-u` User name of IP camera

`-w` Password of IP camera

`-d` Viewer port number

`-a` Application name

Example:

```
# ./RemoteDebugViewer  -i  192.168.1.100   -p   80   -u   admin   -w  4321   -
d   8080   -a   ServerPushMJPEG
```

4 Managing an Application in Web Viewer

4.1 Upgrading Camera Software

To properly manage the application on the camera, use the latest firmware.

Follow the steps below:

Step 1.   Download the latest firmware file from Hanwha Vision.

Step 2.   Log in to the IP camera.

Step 3.   Go to [Setup] > [System] **>** [Upgrade/Restart].
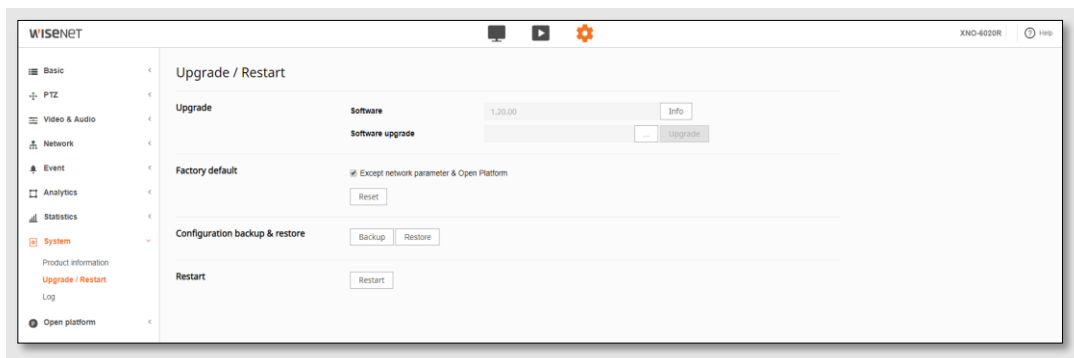
Step 4.   Click the [**…**] button and select the firmware file. Click [Upgrade].

Step 5.   The camera will shut down while upgrading, and then it will reboot.
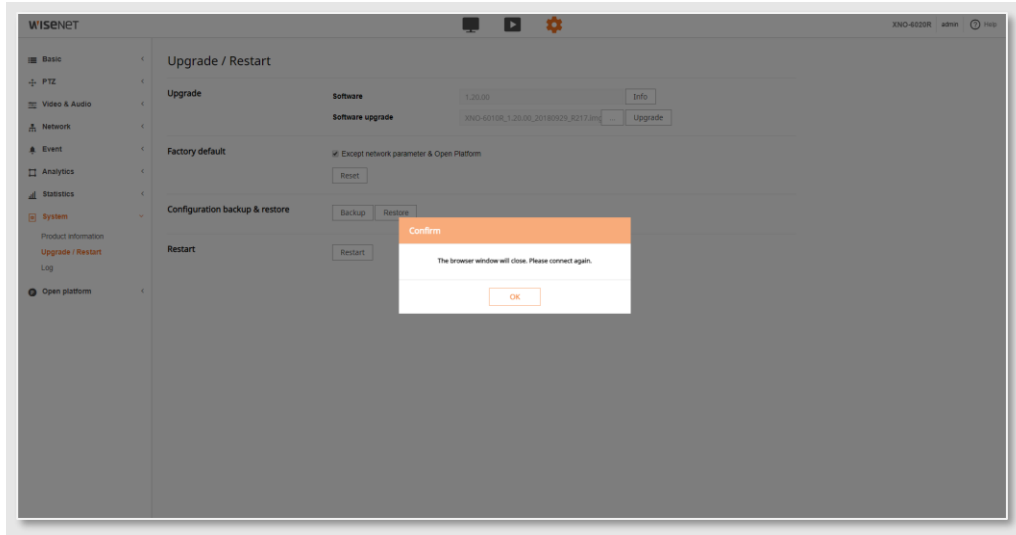
4.2 Installing an Application to the IP Camera

Follow the steps below:

Step 1.   Get a .cap file of the application to install.

Step 2.   Log in to the camera.

Go to [Setup] > [Open platform]. You can check the current open platform version that the camera supports. (For example, you can see the version, 5.00_220216, on the image below.)
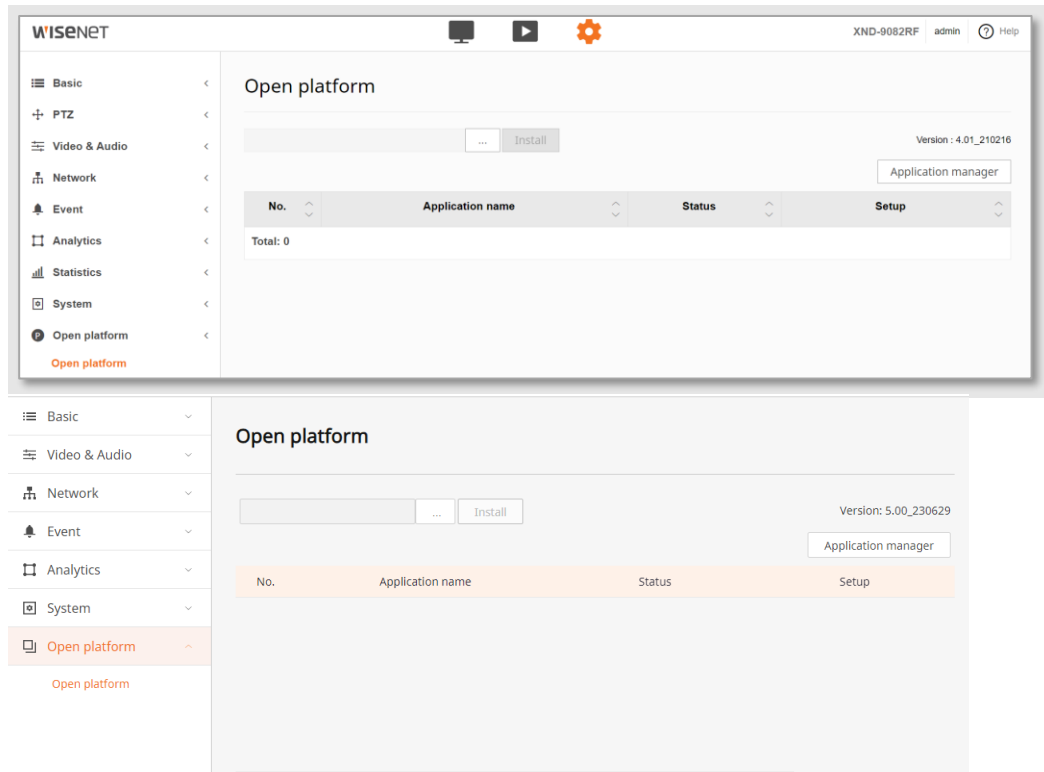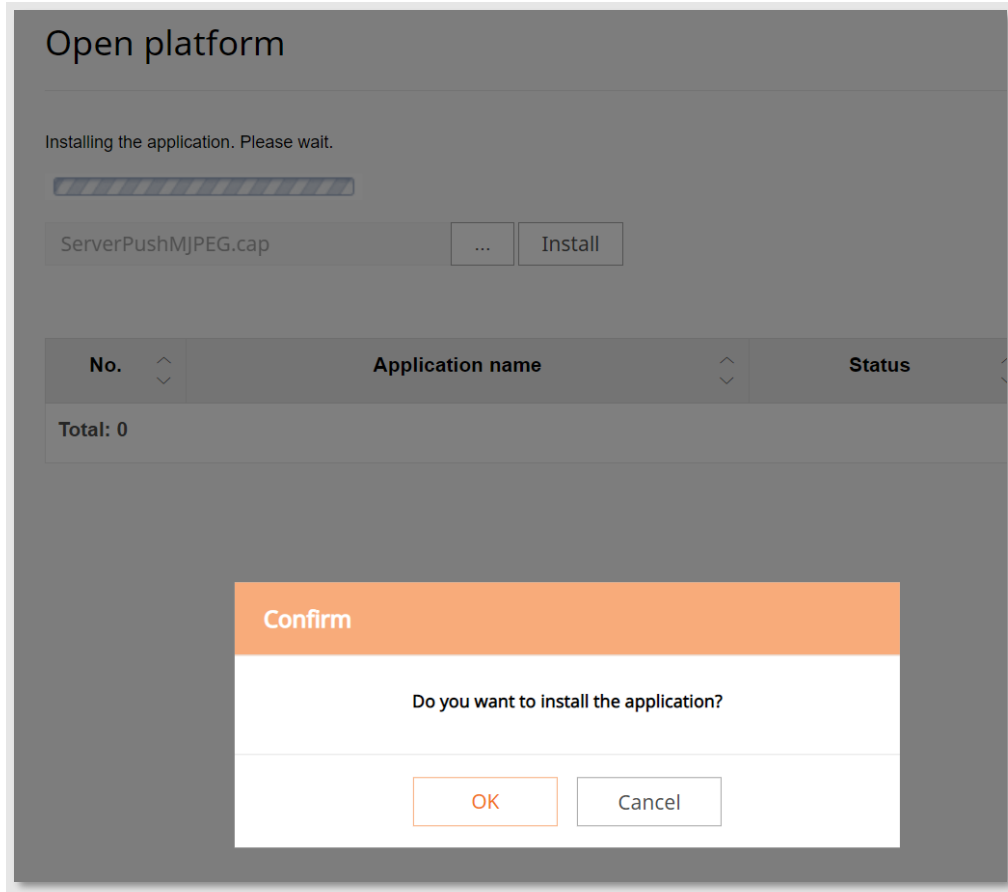
Step 3.   Click the [**...**] button on the top and locate the installation package (.cap file).
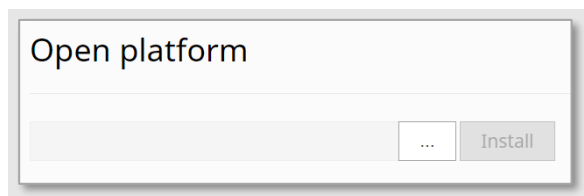
Step 4.   Click [Install]. Then the application will be uploaded, and a window will pop up asking if you want to grant permission to the application.

Step 5.   Click [OK] to grant permissions and install the application. The application will be installed, and a message box will be shown.

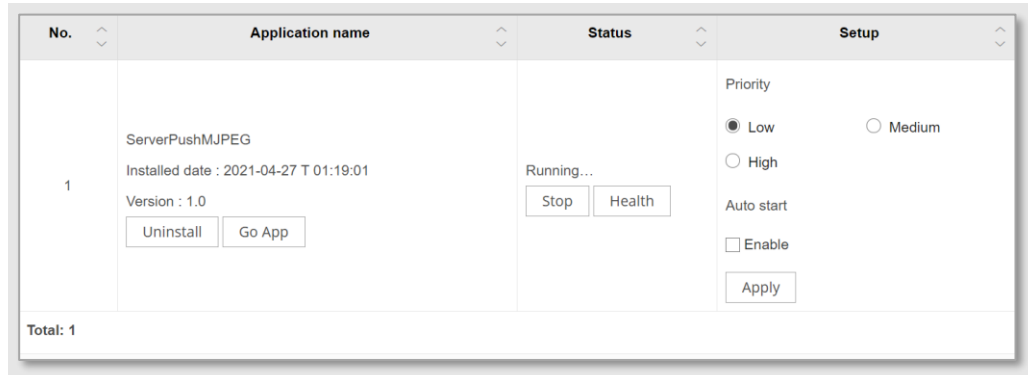**Step 6.** If the cap file is not appropriate, it will not be installed, and an error message will appear.

Step 7. It will appear in the list of installed applications. The status will be changed to "Installed".
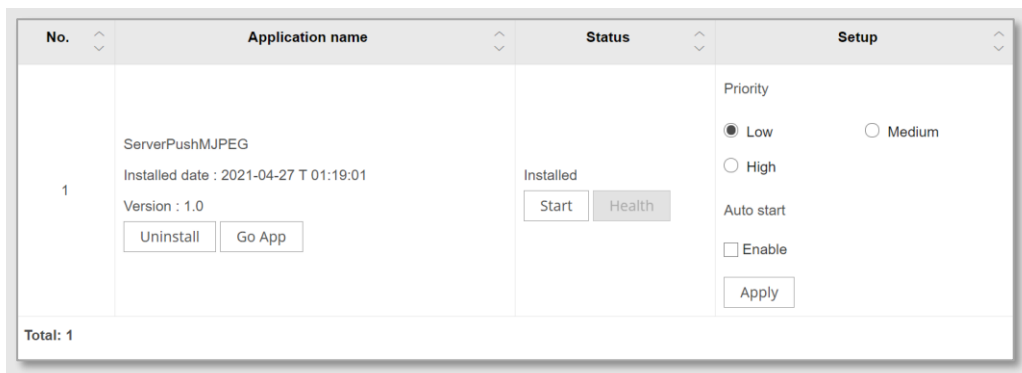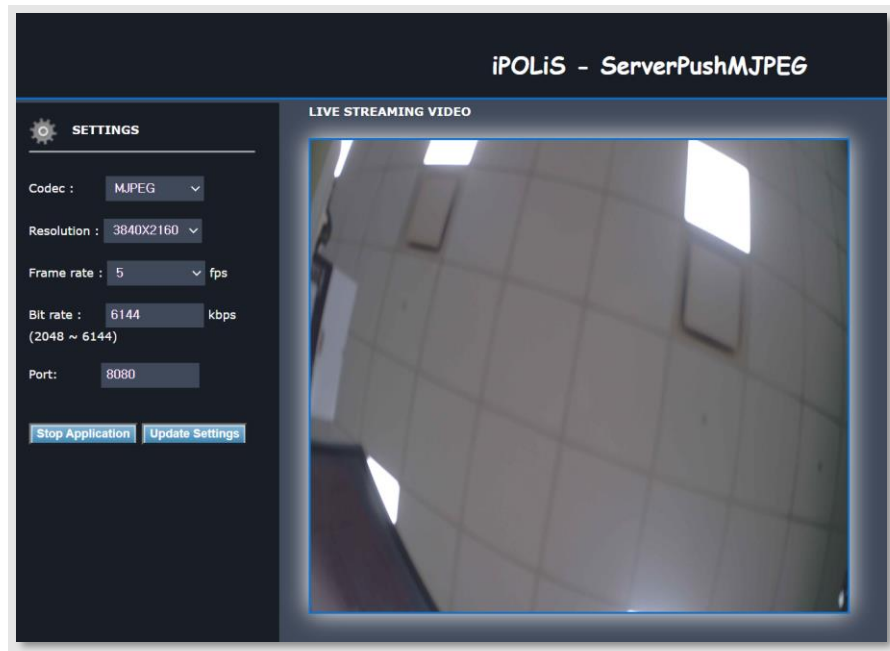

4.3 Executing an Application

Follow the steps below:

Step 1.   Under [Status] click [Start]. The application will start, and the status will be changed to "Running".

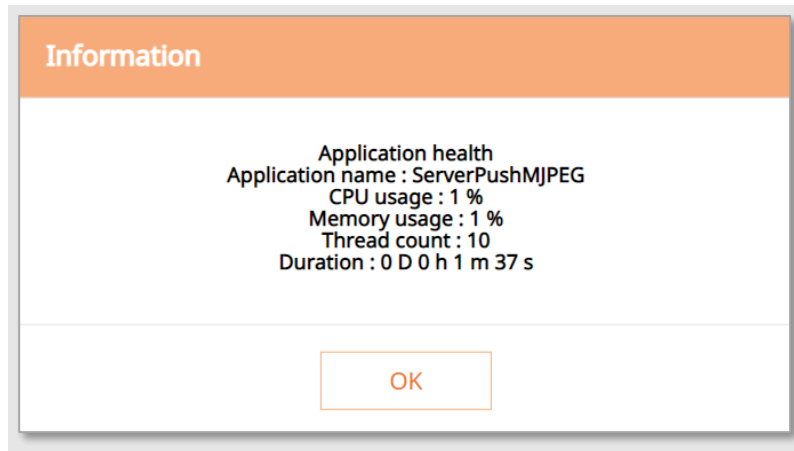Step 2.   To view the running application, click [Go App].

4.4 Checking Application Health

Check the health of any running application.

Follow the steps below:

Step 1.   Under [Status] click [Health]].

Step 2.   A window pops up, showing the CPU usage, memory usage, thread count, and duration of the application. Click [OK] to close the window.

4.5 Application Settings

Set the priority of an application and set the auto-start option.

Follow the steps below:

Step 1.   In [Priority], you can set the priority among running applications. If the resource usage of the whole camera (including the main task of the camera and applications) becomes too high, the application set as "low priority" will close first.

Step 2.   In [Auto start] select [Enable] to execute the application automatically when the camera is powered on and the main task is executed.

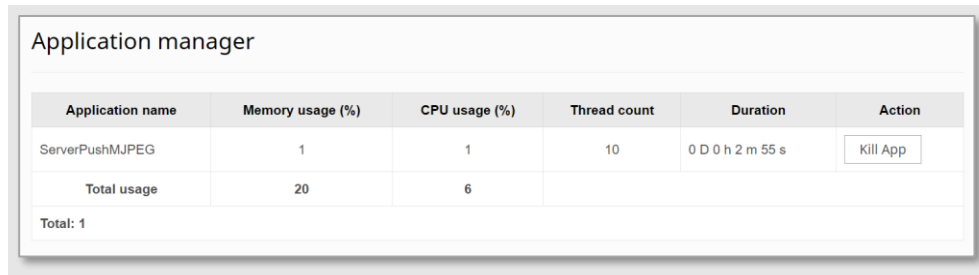Step 3.   Click [Apply] to apply the changes. A success message window pops up.
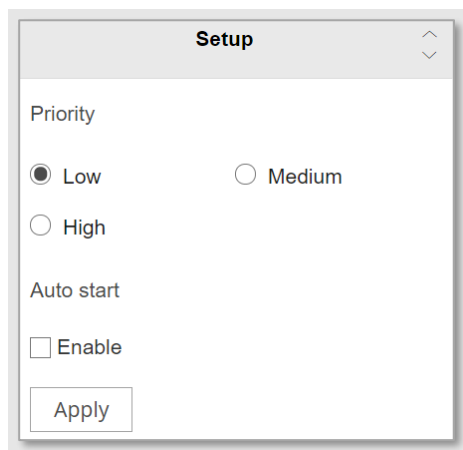
4.6 Application Manager

Like the [Health] function, the camera gives information on all running applications when you click [Application manager].

Follow the steps below:

Step 1.   Click [Application manager].

Step 2.   A window pops up, showing CPU usage, memory usage, thread count, and duration of all running applications.

Step 3.   To close the application, click [Kill App].

Application manager

| Application name	Memory usage (%)	CPU usage (%)	Thread count	Duration	Action
ServerPushMJPEG	1	1	10	0 D 0 h 2 m 55 s	Kill App
Total usage	20	6			

Total: 1

4.7 Stopping an Application

To stop a running application, click [Stop]. Then the status will thereafter be changed to **"**Stopped".

4.8 Uninstalling an Application

Follow the steps below:

Step 1.   To uninstall an application, click [Uninstall].

Step 2.   Click [OK] to confirm the uninstallation.

Step 3.   The application will be removed from the list of installed applications.
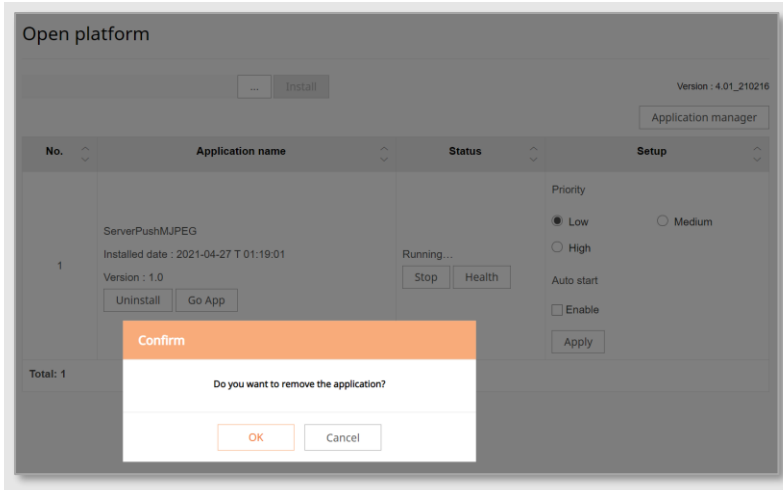
5 Sample Application

Various sample application files are available from the Hanwha Vision Open PlatformSDK.

You may find sample application files under the following path:

```
opt/opensdk/opensdk-${SDK_VERSION}/SampleApplication
```

	Device	Record	Free size	Total size	Status	
●	SD Card	On ▾	29.70 GB	29.71 GB	Recording	Format
○	NAS	Off ▾	0 MB	0 MB	None	Format

Factory default   ☑ Except network parameter & Open Platform

Reset

6 Technical Support

The company operates a membership program for partners who use Hanwha products.
Visit our STEP site at:

- https://step.hanwhavision.com

For those who signed up as an application or technology partner at the initial subscription on out STEP site, the following services are available for the Open Platform SDK.

- Download the SDK versions.
- SDK version and platform information supported by camera models.
- AppTest.key and AppTest.crt are available to be downloaded for the application signing test.
- Personal keys and certificates are issued in order to install the developed apps on cameras.

If you have any inquiries about the development of the Hanwha Vision Open Platformapplication, visit our STEP site and select [Help Desk] > [Q&A].