

# Wisenet Open Platform v4.01

# Programming Guide

v4.01

2021-05-31

## Copyright

©2021 Hanwha Techwin Co., Ltd. All rights reserved.

## Trademark

 is the logo of Hanwha Techwin Co., Ltd.

All other trademarks and trade names presented in this document are the property of their respective holders.

## Restriction

Copyright 2020 © Hanwha Techwin Co., Ltd. All rights reserved. Do not copy, distribute, or reproduce any part of this document without written approval from Hanwha Techwin Co., Ltd.

## Disclaimer

Hanwha Techwin Co., Ltd. has made every effort to ensure the completeness and accuracy of this document, but makes no guarantees regarding the information contained herein. All responsibility for proper and safe use of the information in this document lies with users. Hanwha Techwin Co., Ltd. may revise or update this document without prior notice.

## Contact Information

HANWHA TECHWIN Co., LTD.

Hanwhatechwin R&D Center, 701, Sampyeong-dong, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea, 463-400

TEL: +82-70-7147-8740~60 FAX: +82-31-8018-3745

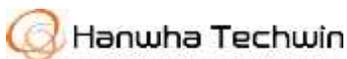
<http://step.hanwha-security.com>

HANWHA TECHWIN AMERICA Inc.

100 Challenger Road Ridgefield Park, New Jersey, 07660 U.S.A.

HANWHA TECHWIN EUROPE LTD.

2nd Floor, No. 5 The Heights, Brooklands, Weybridge, Surrey, KT13 0NY, U.K.



# Revision History

The table below provides the version information and revision history of this document.

Please refer to 2.0x documents for checking revision history before 3.00.

Version	Date	Description
4.01	2021-05-03	Added CLI Added Ubuntu 20.04 docker Unsupported eclipse plugin
4.00	2020-08-31	Added version 4.00 feature list
3.60	2019-12-20	Added installation guide for plugin 64 bit
3.52	2019-09-02	Added new version number  [XML Editor] Added new platforms(T3000) on image
3.51	2019-06-10	Added new version number
3.50	2018-10-29	[Installing Package] <ul style="list-style-type: none"><li>Modified the version number of the Debian package</li></ul> [Creating a New Project] [XML Editor] <ul style="list-style-type: none"><li>Added new platforms (XF8000, T4000) on image</li></ul> [Web Page Design] <ul style="list-style-type: none"><li>Modified WEB UI on image</li></ul> [Device Preferences] <ul style="list-style-type: none"><li>Deleted content</li></ul>
3.00_2nd	2017-12-08	Added supported Models <ul style="list-style-type: none"><li>Thermal Models</li></ul>
3.00	2017-04-14	New revision

# Table of Contents

<b>1</b>	<b>Introduction to Wisenet Open Platform.....</b>	<b>5</b>
1.1	Overview .....	5
1.2	Main Features of Wisenet Open Platform (WOP).....	5
1.3	What's new in Wisenet Open Platform SDK.....	6
1.4	Requirements .....	9
1.5	Supported Camera Models .....	11
<b>2</b>	<b>Developing an Application .....</b>	<b>12</b>
2.1	Prerequisites .....	12
2.2	How to Run Docker Images.....	12
2.3	CLI Guide .....	13
2.4	Folder and File Structure .....	14
2.5	To Create WOP Application Using CLI .....	16
<b>3</b>	<b>Debugging .....</b>	<b>19</b>
3.1	Usage of Remote Debug Viewer .....	19
<b>4</b>	<b>Managing an Application in Web Viewer.....</b>	<b>20</b>
4.1	Upgrading Camera Software .....	20
4.2	Installing an Application to the IP Camera.....	21
4.3	Executing an Application.....	23
4.4	Checking Application Health .....	24
4.5	Application Settings.....	25
4.6	Application Manager.....	25
4.7	Stopping an Application .....	26
4.8	Uninstalling an Application .....	26
<b>5</b>	<b>Sample Application .....</b>	<b>28</b>

<b>6</b>	<b>Troubleshooting</b> .....	<b>29</b>
6.1	Storing Files on a Flash Memory.....	29
6.2	Storing Files on a SD Card .....	29
6.3	Implementing FTP/Email/JPEG Encoding Function.....	30
6.4	Preserving Applications after a Factory Default .....	30
<b>7</b>	<b>Technical Support</b> .....	<b>31</b>

# 1 Introduction to Wisenet Open Platform

## 1.1 Overview

---

The Wisenet Open Platform (WOP) uses a standardized API to enable users to add additional functions to an IP camera without changing existing software.

This platform supports the downloading/installation of third-party applications to 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 Main Features of Wisenet Open Platform (WOP)

---

The Wisenet Open Platform SDK provides five main functions: real-time media streaming, video control, video analytics, recording, and video setup. In addition to these functions, the SDK APIs provide several events and network functions. Please 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 using a developer's unique analysis algorithm.

- **Video control**

- Using video profile APIs, a developer can make an application that can change and control video. The developer can change the video source to NTSC or PAL format and add/delete a video profile according to their 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 which is used by IP camera: motion detection, face detection, appear/disappear, tampering event, intelligent video, enter/exit, and alarm event.

e.g., An application provides an alarm for users when an event has occurred.

- **Recording**

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

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

- **Video setup**

- Developer can set the video according to their preference. SDK APIs provides functions to set/get privacy mask to video.

e.g., User can store the video stream with private mask to SD card or storage.

## 1.3 What's new in Wisenet Open Platform SDK

---

### New Features in Version 4.01

- **Support docker**

- Added docker image file to provide development environment.

- **Add CLI command**

- Added CLI (Command line interface) to simplify project creation and setup and disable eclipse plug-ins.

- **Unsupported eclipse plugin**

- Replace eclipse plugins with CLI.

## New Features in Version 4.00

- **Application signature**
  - Added the Application signature feature to protect from security risks, such as application tampering.
- **SD card-dedicated use**
  - Added an API that enables users 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 lets users directly access the raw image pointer so that they can use raw images with up to 4K resolution.

## New Features in Version 3.60

- **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 Version 3.52

- **Dynamic event**
  - Application can send events for regular templates.
  - Refer to SDKAPI document
- **Nand flash space**
  - Application can use nand flash space.
  - Refer to 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 Version 3.51

- **Internal modification**
  - Synchronized with X series camera firmware

## New Features in Version 3.50

- **SUNAPI API**
  - Added API for using SUNAPI on the application
- **RS-485 event**
  - Added event to transfer RS-485 data by using the terminal
- **Proxy API**
  - Added 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
  - Event will come when system total usage is over 80% (Max 3 times)
- **Fixed defective issue**
  - Included improved performance caused by VOC and defective issues
- **Upgrade Debian**
  - Added sample applications on Debian (/opt/opensdk/opensdk-3.50/SampleApplication)
  - Upgraded to latest version of libraries

## New Features in Version 3.00

- **Raw audio**
  - Open Platform SDK supports raw audio
    - e.g., An application provides audio analytics functions using a developer's unique analysis algorithm.
- **Application filename**

- Application filename can be changed after compiling
- OriginalAppName\_v{version}.cap  
  - e.g., ServerPushMJPEG.cap → ServerPushMJPEG\_v170414.cap
- **JAVA 8**
  - Updated development environment
- **Eclipse Neon2**
  - Updated development environment

## 1.4 Requirements

---

### 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
<b>CPU</b>	3040 DMIPS	System total up to 80%
<b>RAM</b>	352 MB	System total up to 80%
<b>Flash</b>	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
<b>CPU</b>	2040 DMIPS	System total up to 80%
<b>RAM</b>	225 MB	System total up to 80%
<b>Flash</b>	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
<b>CPU</b>	9273.6 DMIPS	System total up to 80%
<b>RAM</b>	4 GB	System total up to 80%
<b>Flash</b>	512 MB	50 MB

## Open Platform Resources of CV22S66 Series

LPR Camera (TNO-7180R and so on)

	System TOTAL	Open Platform Usability Range
<b>CPU</b>	9273.6 DMIPS	System total up to 80%
<b>RAM</b>	2 GB	System total up to 80%
<b>Flash</b>	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
<b>CPU</b>	3420 DMIPS	System total up to 80%
<b>RAM</b>	4 GB	System total up to 80%
<b>Flash</b>	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
<b>CPU</b>	9273.6 DMIPS	System total up to 80%
<b>RAM</b>	4 GB	System total up to 80%
<b>Flash</b>	512 MB	150 MB

## Open Platform Resources of CV2x OP Series

P series AI 2M camera (PNO-A9081ROP and so on)

	System TOTAL	Open Platform Usability Range
<b>CPU</b>	9273.6 DMIPS	System total up to 80%
<b>RAM</b>	4 GB	System total up to 80%
<b>Flash</b>	512 MB	150 MB

## 1.5 Supported Camera Models

---

Most SDK Wisenet cameras support Wisenet 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.hanwha-security.com>

# 2 Developing an Application

## 2.1 Prerequisites

---

To develop a Wisenet Open Platform (WOP) application, prepare the following:

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

### Note

Wisenet Open Platform version 4.01 or later supports the application development using Docker. For information on versions earlier than version 4.01, refer to our STEP site (<https://step.hanwha-security.com>).

## 2.2 How to Run Docker Images

---

### Note

To run Docker images, you have to install Docker first. For Docker installation, refer to the following steps:

<https://docs.docker.com>

<https://hub.docker.com/editions/community/docker-ce-desktop-windows/> (windows)

### Procedure

---

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

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

**Step 2.** Load the compressed file as a Docker image.

```
root@ubuntu:/home/<user_id># docker load -i opensdk_docker<version>.tar.gz
```

**Step 3.** Check the loaded image.

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

**Step 4.** Run the image as a container to access the Wisenet Open Platform 4.01 environment.

```
root@ubuntu:/home/<user_id># docker run --rm -it opensdk_docker<version> /bin/bash
```

**Step 5.** Exit the container.

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

## 2.3 CLI Guide

---

### opensdk\_new\_project

**Description** 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 4.01 -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 4.01 -p WN7 -p CV2X
```

### opensdk\_install

Install the application on the camera.

#### Description

```
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

#### Description

Check the camera's platform.

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

#### Example

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

## opensdk\_packager

### Description

Compress and encrypt the application files.

```
opensdk_packager
```

#### Example

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

## 2.4 Folder and File Structure

---

The folder and file structure of the application is as follows:



**Includes**

Contains paths of 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 (user can make changes)
  - Application Configuration: Key and Value information (user 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 To Create WOP 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

You have to check the Wisenet Open Platform (WOP) version supported by the camera and use the SDK of the applicable version. The CGI command to check the WOP version is as follows:

```
http://<camera IP>/stw-cgi/system.cgi?submenu=deviceinfo&action=view
```

### Procedure

---

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

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

You will receive the following response: It says you should use WOP SDK 4.01 for application development.

```
Model=XNV-9082R
...
OpenSDKVersion=4.01_210216
```

- Step 2.** From the Linux editor, check the platform of the XNV-9082R camera model.

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

You will receive the following response: It says the "WN7" platform is used.

```
XNV-9082R is in WN7 platform
```

- Step 3.** Create a template code to develop the version 1.0 application named `wisenet` under the SDK version 4.01. The `wisenet` application will be used on the camera developed under the WN7 platform.

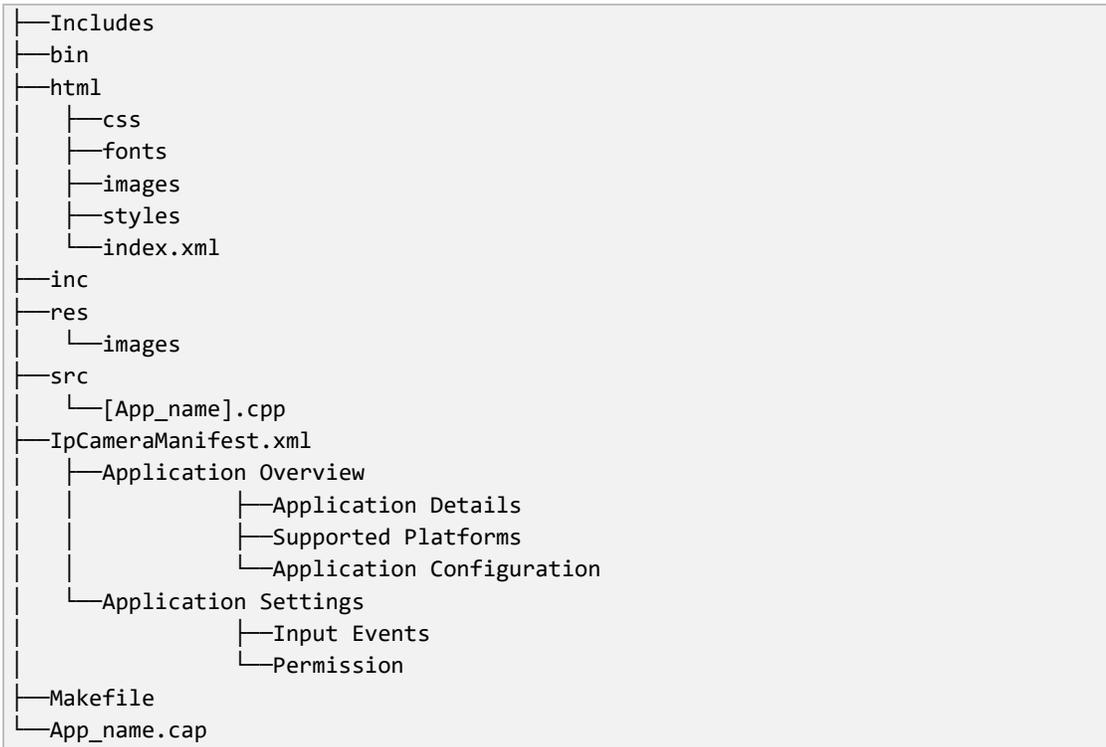
```
# opensdk_new_project -n wisenet -v 1.0 -s 4.01 -p WN7
```

- Step 4.** Go to the folder which 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 "Folder and File Structure (Page 14)".



For a platform in which a certificate is required, if it is not placed under "/opt/opensdk/signature," the application won't be installed on the camera correctly.

#### **How to issue a certificate**

- During the application development phase, you may download a test key and a test certificate from our STEP site. (They are valid for 3 months.)
- When you register an application you've developed on our STEP site, you should 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 Techwin under the following directory in your development PC:  
"/opt/opensdk/signature"
  - C. Execute opensdk\_packager, and 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 "Installing an Application to the IP Camera (Page 21)".

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

# 3 Debugging

Remote debug viewer is a command line application used to view logs from a 3<sup>rd</sup> party application. This application will come with the SDK package, and needs to be executed from the Linux command prompt.

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

Or, you can find it easily using 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 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, it is recommended to have the latest firmware.

To upgrade to the firmware, follow the steps below:

### Procedure

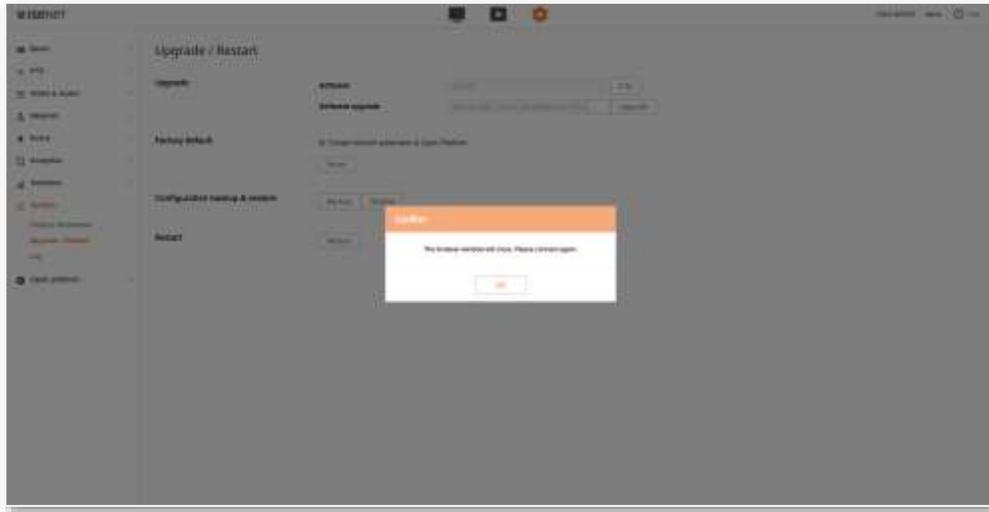
- Step 1.** Get a firmware file from Hanwha Techwin.
- Step 2.** Login to IP camera.
- Step 3.** Go to **Setup>System>Upgrade/Restart**.



- Step 4.** Click the ... button and select the firmware file. Click **Upgrade** button.



- Step 5.** The camera will shut down while upgrading and will then reboot.

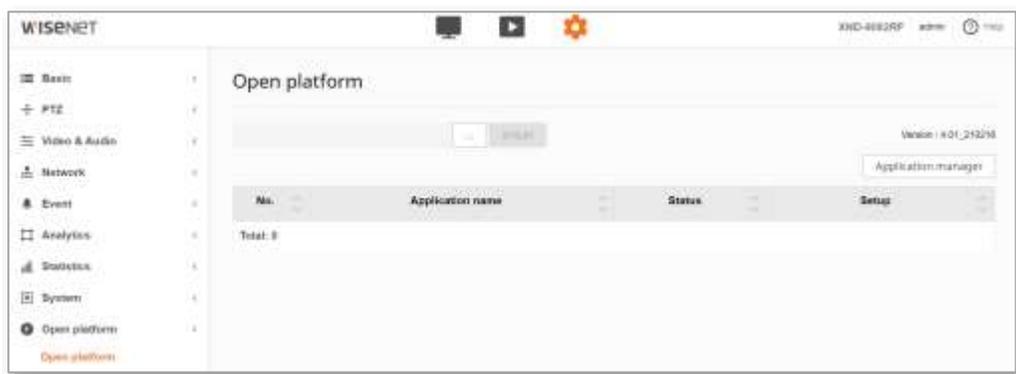


## 4.2 Installing an Application to the IP Camera

### Procedure

- Step 1. Get ".cap" file of the application to install.
- Step 2. Login to camera.

Go to **Setup>Open Platform**. You can check the current Open Platform version that the camera supports. (You may see the version, 4.01\_210216, on the below figure, for example.)

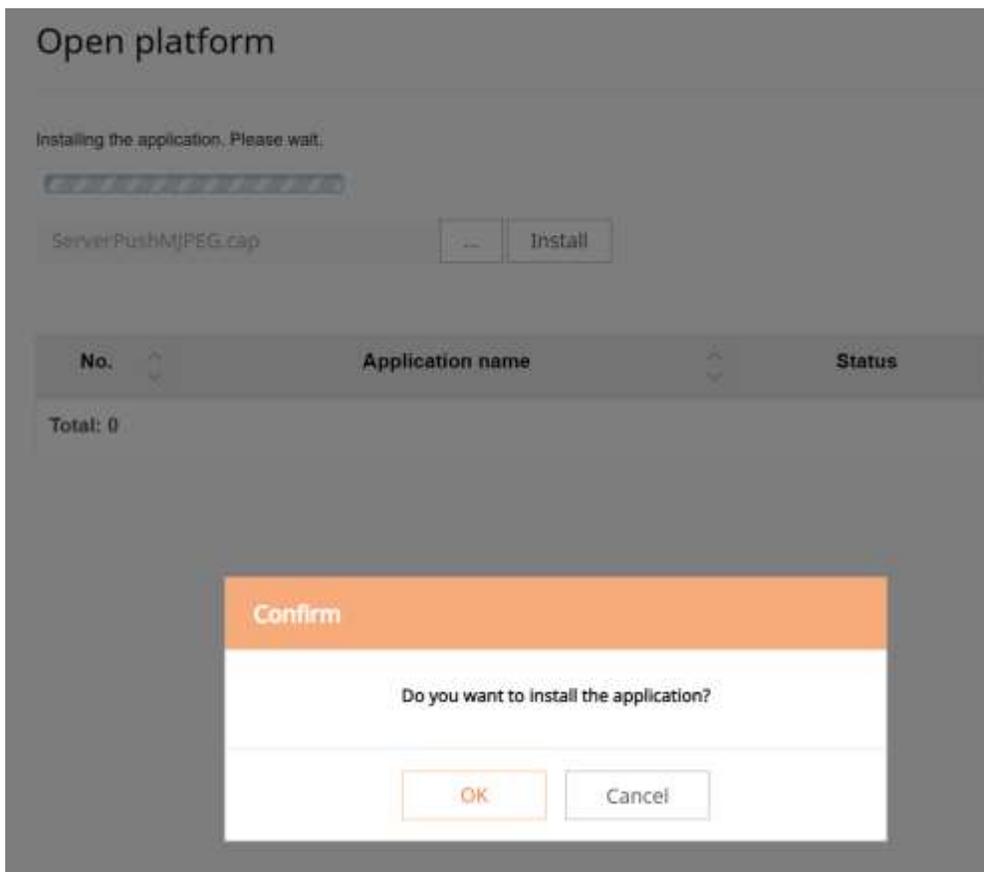


- Step 3. Click ... on the top and locate the installation package (.cap file).



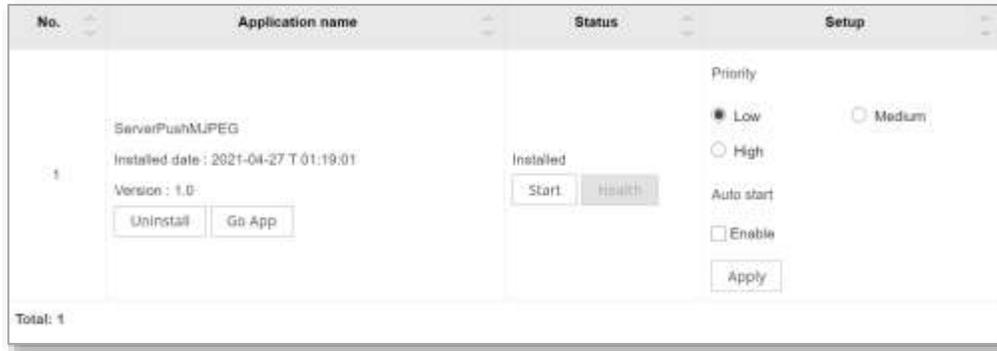
**Step 4.** Click **Install**. The application will be uploaded, and a window pops up and displays the permissions required for the application.

**Step 5.** Click **OK** to allow the permissions and install. 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 be shown.

**Step 7.** It will appear in the list of installed applications. The status shown will be **Installed**.



## 4.3 Executing an Application

---

**Step 1.** Click the **Start** button. The application will start and the status will be shown as **Running**.



**Step 2.** To see the running application, click the **Go App** button.



## 4.4 Checking Application Health

---

The health of any running application can be checked.

### Procedure

---

- Step 1.** Click the **Health** button.
- Step 2.** A window pops up and displays CPU usage, memory usage, thread count, and duration of the application. Click **OK** to close the window.



## 4.5 Application Settings

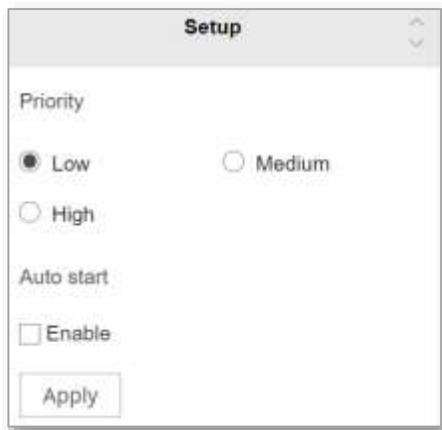
---

Settings allow the user to set the priority of an application and set the auto start option.

### Procedure

---

- 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 applications set as "low priority" will close first.
- Step 2.** Check **Enable** in **Auto start** to execute the application automatically when the camera is powered on and the main task is executed.
- Step 3.** Click **Apply** to set the changes. A success message window pops up.



## 4.6 Application Manager

---

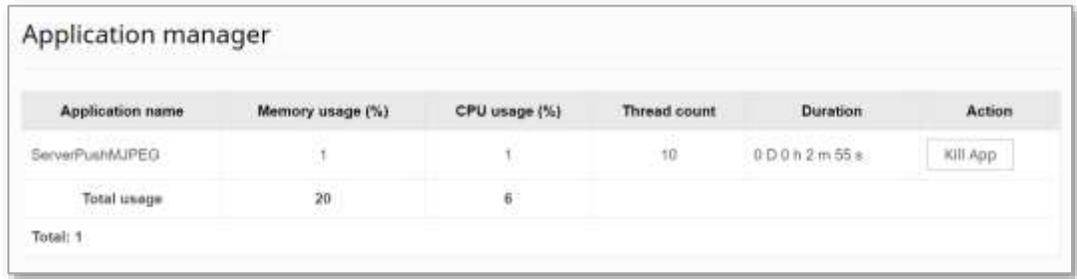
Like the **Health** function, the camera gives information on all running applications when the **Application Manager** button is clicked.

### Procedure

---

- Step 1.** Click the **Application Manager** button.
- Step 2.** A window pops up and displays CPU usage, memory usage, thread count, and duration of all running applications.

**Step 3.** The user can stop an application using the **Kill App** button.



The screenshot shows a window titled "Application manager" containing a table with the following data:

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

## 4.7 Stopping an Application

---

To stop a running application, click the **Stop** button.

The status will be updated as **Stopped**.

## 4.8 Uninstalling an Application

---

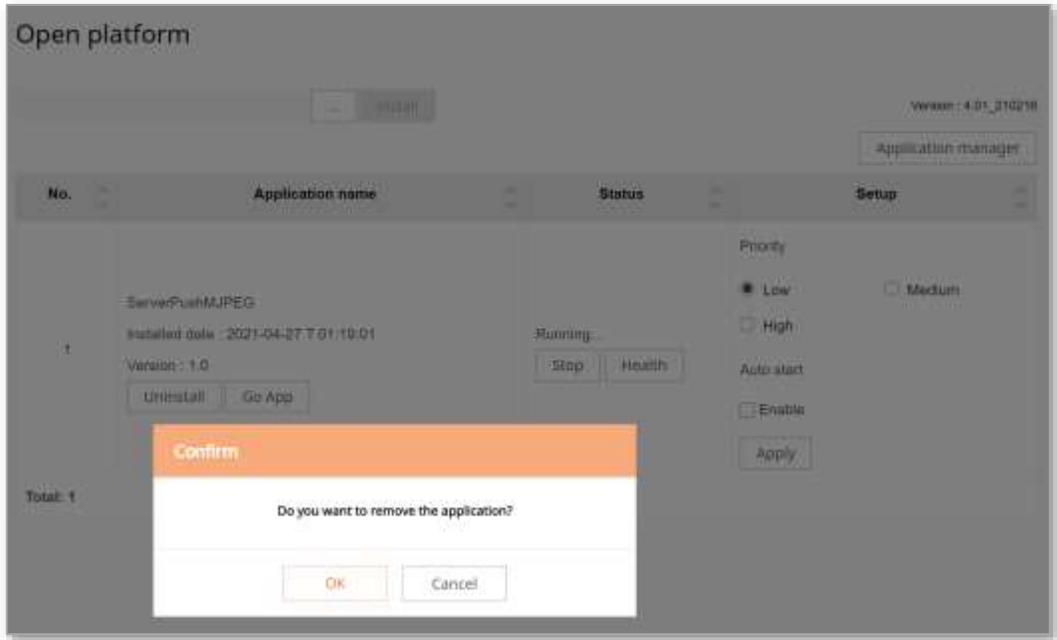
### Procedure

---

**Step 1.** To uninstall an application, click the **Uninstall** button.

**Step 2.** Click **OK** to confirm.

**Step 3.** The application will be removed from the list of installed applications.



# 5 Sample Application

Various sample application files are available from the Wisenet Open Platform SDK.

You may find sample application files under the following path:

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

# 6 Troubleshooting

## 6.1 Storing Files on a Flash Memory

You may read and write files under the following path:

```
/mnt/opensdk/apps/<AppName>
```

If you delete an application, the corresponding information stored in the flash memory will be removed as well. If you reboot the camera only without deleting the application, stored files won't be removed but will be available. For information to be saved even after the application is deleted, refer to "Nand Flash Space" under Chapter 6 Others in the SDK API document.

## 6.2 Storing Files on a SD Card

Use the following API to check the path for the mounted SD card.

```
Const INT8_N *opensdk_getSDcardStoragePath();
```

For any of the following cases, NULL will be returned.

- No SD card is mounted
- No access right available to the SD card in "IPCameraManifest.xml"

If NULL is returned despite not applicable to cases above, check if the SD card is activated. To activate the SD card from the camera web viewer, select on the menu Basic > Event > Storage, select [SD Card] from the [Storage action setup] list, and choose [On] under the [Record] column.

	Device	Record	Free size	Total size	Status	
<input checked="" type="radio"/>	SD Card	On	29.70 GB	29.71 GB	Recording	Format
<input type="radio"/>	NAS	Off	0 MB	0 MB	None	Format

The space is limited to 80% of the size of the SD card. If the SD card recording is enabled, the space will be shared with a network camera main application.

## 6.3 Implementing FTP/Email/JPEG Encoding Function

---

Regarding FTP/Email/JPEG encoding, you should receive the event after calling `send_event()`.

The behavior scenario is as below:

```
Open Platform Application           Apphelper
send_event() -----> processing
rcv_event() <----- processing result
```

Event type is `OPENSdk_EVENT_STATUS`.

`TASK_STATUS.TaskName` will be `OPENSdk_FILE_UPLOAD` for FTP, `OPENSdk_JPEG_ENCODE` for JPEG Encoding.

In the case of JPEG encoding, it takes up to 100 ms. If you request it without waiting for the result notification, it causes memory leaks.

## 6.4 Preserving Applications after a Factory Default

---

During factory default initialization, you may preserve open applications installed and their settings but reset other settings.

If you don't want to remove open applications during factory default initialization, select on the menu `System > Upgrade/Restart` from the web viewer, and select `[Except network parameter & open platform]` under `[Factory default]`.



If you physically press the reset button on the camera, all settings, including network settings and open platform information, will be initialized to the factory default.

# 7 Technical Support

The company operates a membership program for partners who use Wisenet products. Check our STEP site as follows:

- <https://step.hanwha-security.com>

For those whose signed up as an Application or Technology Partner at the initial subscription on our STEP site, the following services are available for the Open Platform SDK.

- Download by 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 developed apps on cameras.

For any inquires on Wisenet Open Platform application development, please visit Help Desk > Q&A on our STEP site.