# Web Page Development Guide

Hanwha Vision Open Platform
v1.0.0
2025-01-02



**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

Hanwha Vision

# Table of Contents

# 1 Prerequisites

To create an application using the SDK, prepare the following software and hardware:

- Linux, or Windows
- Hanwha Vision Open Platform SDK
- Hanwha IP camera that supports Hanwha Vision Open Platform

# 2 Functions

These functions are defined in the camera to request information or control third-party applications. Download home/js/SDKApi.js for details on these functions.

## 2.1 RequestAjaxMsg

**Define**

RequestAjaxMsg function

| Method | Parameters |
|---|---|
| RequestAjaxMsg | msg, alertMsg, reqUrl, command, asyncVal |

**Description**

This basic function is used for requesting or controlling the functions of the third-party application. This function is defined in the main server of the camera, so it can be used only for starting or stopping the application.

**Parameters**

| Members | Description |
|---|---|
| Msg | The string to start or stop the application* |
| alertMsg | To alert when the function call is failed<br><br>You can decide whether to use this message. |
| reqUrl | The URL processes all request message |
| Command | SDK/SDK_APP/SDK_APP_DATA<br><br>See 3. Commands |
| asyncVal | For most cases, use “undefined” |

**Return**

This function returns no string when the request is processed successfully. If not, it returns the “alertMsg”.

> **Note**
>
> The format of “msg” is xml. Refer to the following chapter.

## 2.2 startApplication

**Define**

startApplication function

| Method | Parameters |
|---|---|
| startApplication | paramJSON |

**Description**

This function is used for starting the application. This function is the same as when the **SDK** command is used.

**Parameters**

| Members | Description |
|---|---|
| paramJSON | JSON string that contains the following information:<br><br>appname: Name of the application.<br><br>success: Callback function when the application starts successfully.<br><br>error: Callback function when the application doesn’t start. |

**Return**

The *paramJSON* parameter is updated without returning anything.

## 2.3 stopApplication

**Define**

stopApplication function

| Method | Parameters |
|---|---|
| stopApplication | paramJSON |

**Description**

This function is used for stopping the application. This function is the same as when the **SDK** command is used.

**Parameters**

| Members | Description |
| --- | --- |
| paramJSON | JSON string that contains the following information:<br><br>appname: Name of the application.<br><br>success: Callback function when the application stops successfully.<br><br>error: Callback function when the application doesn't stop. |

**Return**

The *paramJSON* parameter is updated without returning anything.

# 2.4 getApplicationStatus

**Define**

getApplicationStatus function

| Method | Parameters |
| --- | --- |
| getApplicationStatus | paramJSON |

**Description**

This function is used for fetching application status information. This function returns the status of an application, such as running, stopped, etc.

**Parameters**

| Members | Description |
| --- | --- |
| paramJSON | JSON string that contains the following information:<br><br>appname: Name of the application.<br><br>success: Callback function when the application status information is fetched successfully. |

**Return**

The *paramJSON* parameter is updated without returning anything.

# 2.5 getApplicationSettings

**Define**

getApplicationSettings function

| Method | Parameters |
| --- | --- |
| getApplicationSettings | paramJSON |

**Description**

This function is used for fetching application settings information. This function is the same as when the **SDK_APP** command is used.

**Parameters**

| Members | Description |
| --- | --- |
| paramJSON | JSON string that contains the following information:<br><br>appname: Name of the application.<br><br>success: Callback function that gets the fetched settings information. |

**Return**

The *paramJSON* parameter is updated without returning anything.

## 2.6 updateApplicationSettings

**Define**

updateApplicationSettings function

| Method | Parameters |
| --- | --- |
| updateApplicationSettings | paramJSON |

**Description**

This function is used for updating the application settings information. This function is the same as when the **SDK_APP** command is used.

**Parameters**

| Members | Description |
| --- | --- |
| paramJSON | JSON string that contains following information:<br><br>appname: Name of the application.<br><br>appconfig: Application configuration data.<br><br>success: Callback function called when settings information is updated successfully.<br><br>error: Callback function when there is an error. |

**Return**

The *paramJSON* parameter is updated without returning anything.

# 2.7 sendCommandToServer

**Define**

sendCommandToServer function

| Method | Parameters |
| --- | --- |
| sendCommandToServer | paramJSON |

**Description**

This function is used for sending requests to the server. This function is the same as when the **SDK_APP_DATA** command is used.

**Parameters**

| Members | Description |
| --- | --- |
| paramJSON | JSON string that contains the following information :<br>appname: Name of the application.<br>requestMessage: message String containing request.<br><GetSDK_APP_DATA><br><AppName>application_name</AppName><br><Data>...</Data><br></GetSDK_APP_DATA><br>success: Callback function called with response from the server.<br>error: Callback function in case of error. |

**Return**

The *paramJSON* parameter is updated without returning anything.

> **Note**
>
> - If an error occurs when you use the **sendCommandToServer** function, change <Data>...</Data> to <data>...</data>.
> - Do not use the terms “data” or “Data” in the <Data> ... </Data>, <data> ... </data>. They may cause an xml parsing malfunction.

# 3 Commands

The following commands provide a way to communicate with the application and main server.

**Parameters**

Web communication commands

| Command | Description |
| --- | --- |
| SDK | To control the application, such as start/stop |
| SDK_APP | To set or get the setting of the application |
| SDK_APP_DATA | To communicate with the application, use this command |

## 3.1 SDK Command

This command is used to start or stop the application using the third-party application’s own web page.

Syntax

| Usage | Syntax |
| --- | --- |
| Start the app | <StartSDK><br><AppName>application_name</AppName><br></StartSDK> |
| Stop the app | <StopSDK><br><AppName>application_name</AppName><br></StopSDK> |

**Description**

Syntax is the message used as a parameter("msg") for the **RequestAjaxMsg** function. Type the full application name for "application_name" as defined in an IPCameraManifest.xml file.

**Return**

After the **RequesAjaxMsg** function is called, the web page is reloaded regardless of success. If the function fails to start or stop the application, the web page shows the alert message("alertMsg") as a popup.

# 3.2 SDK_APP Command

This command is used to set or get the settings of the application, especially the setting values defined in IPCameraManifest.xml.

Syntax

| Usage | Syntax |
| --- | --- |
| Set the settings | <SetSDK_APP><br><AppName>application_name</AppName><br><VideoEncoding>...</VideoEncoding><br><rawVideo>...</rawVideo><br><appConfigData>...</appConfigData><br><vaEvent>...</vaEvent><br><rawAudio>...</rawAudio><br></SetSDK_APP> |

| Usage | Syntax |
| --- | --- |
| Get the settings | <GetSDK_APP><br><AppName>application_name</AppName><br></GetSDK_APP> |

**Description**

Syntax is the message used as a parameter for the request function (request function will be defined). Type the full application name in “application_name” as defined in an IPCameraManifest.xml file.

**Function Define**

Request Function

| Method | Parameters |
| --- | --- |
| Any method allowed | msg, alertMsg, reqUrl, command, asyncVal |

**Function Description**

This is the function to set or get the basic information defined in an IPCameraManifest.xml file. This function is defined based on the **RequestAjaxMsg** function.

**Function Parameters**

| Members | Description |
| --- | --- |
| Msg | The string to set or get the IPCameraManifest.xml |
| alertMsg | To alert when the function call is failed<br>You can decide whether to use this message. |
| reqUrl | The URL processes all request messages |
| Command | SDK_APP |
| asyncVal | In most cases, use “undefined” |

**Function Return**

| Usage | Syntax |
| --- | --- |
| Success to set | <Results><br><Status>OK</Status><br></Results> |
| Success to get | IPCameraManifest.xml in xml format |

| Usage | Syntax |
| --- | --- |
| Fail to set or get | <Error><br><br><ErrorString>ERROR: Configuration file not found<br><br></ErrorString><br><br></Error> |

**Function Example**

```
function GetSettings()
{
    var msg = “”;
    msg = “<GetSDK_APP>”;
    msg += “<AppName>ServerPushMJPEG</AppName>”;
    msg += “</GetSDK_APP>”;
    SDK_APPGetSettingsAjaxMsg(msg, “”, URL, ‘SDK_APP’, ‘undefined’);
}

function SDK_APPGetSettingsAjaxMsg(msg, alertMsg, reqUrl, command,
asyncVal)
{
     if(typeof command == “undefined”) command = “”;
     if(typeof asyncVal == “undefined”) asyncVal = false;

     $.ajax({
          type: “POST”,
          async: asyncVal,
          cache : false,
          url : reqUrl,
          dataType: “xml”,
          data: encodeURI(msg),
          success: OnSuccessGetSettings
});
}

function OnSuccessGetSettings(ack)
{
     if($(ack).find(“ErrorString”).text())
     // Request is failed, process error.
     if($(ack).find(“model”).text())
```

```
    // Request is success
}
```

> **Note**
>
> See the example of “ServerPushMJPEG”.

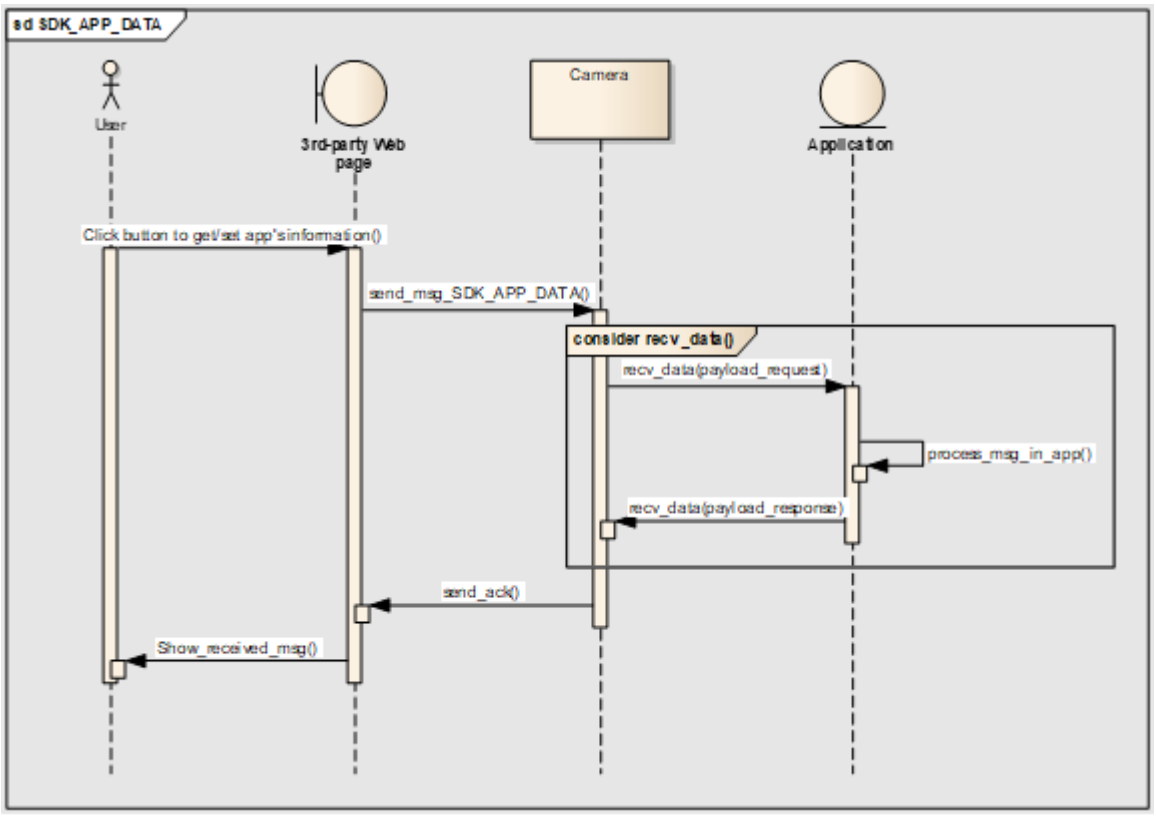
## 3.3 SDK_APP_DATA Command

This command is used to communicate with the application. The sequence diagram of this command is as shown in the picture below.

When the camera receives “SDK_APP_DATA” messages from the web, it sends the message to the application, especially “recv_data(void*, void*)”.

The function (recv_data()) has 2 parameters to receive and send messages: the first parameter (void *payload_request) is to receive the message from the camera (from the web) and the second (void *payload_response) is to send the message to the camera (to the web).

See the main source code when you make a project in Docker.




Syntax

| Usage | Syntax |
| --- | --- |
| Send the message to the application | <SetSDK_APP_DATA><br><AppName>application_name</AppName><br><Data>...</Data><br></SetSDK_APP_DATA><br><GetSDK_APP_DATA><br><AppName>application_name</AppName><br><Data>...</Data><br></GetSDK_APP_DATA> |

**Description**

Syntax is the message used as a parameter for the request function (see the example below). Type the full application name in “application_name” as defined in an IPCameraManifest.xml file.

> **Note**
>
> If an error occurred when running the **SDK_APP_DATA** command, change <Data>...</Data> to <data>...</data>.

**Function Example**

```
function GetSettings()
{
    var msg = “”;
    msg = “<GetSDK_APP>”;
    msg += “<AppName>ServerPushMJPEG</AppName>”;
    msg += “<Data>GetInitData</Data>”
    msg += “</GetSDK_APP>”;
    SDK_APP_DataGetAjaxMsg(msg, “”, URL, ‘SDK_APP_DATA’, ‘undefined’);
}

function SDK_APP_DATAGetMsg(msg, alertMsg, reqUrl, command, asyncVal)
{
     if(typeof command == “undefined”) command = “”;
     if(typeof asyncVal == “undefined”) asyncVal = false;

     $.ajax({
          type: “POST”,
```

```
          async: asyncVal,
          cache : false,
          url : reqUrl,
          dataType: “xml”,
          data: encodeURI(msg),
          success: OnSuccessGet
});
}

function OnSuccessGet(ack)
{
if($(ack).find(“Success”).text())
     // Request is success, the message is from the application
}
```

```
OPENSDK_ERR_CODE recv_data(void *payload_request, void *payload_response)
{
    OPENSDK_PAYLOAD_REQUEST *req =
    (OPENSDK_PAYLOAD_REQUEST *)payload_request;
    OPENSDK_PAYLOAD_REQUEST *rsp =
    (OPENSDK_PAYLOAD_REQUEST *)payload_response;
    if(strstr(req_payload->pBuff, “GetInitData”) != NULL)
    {
        strcpy(res_payload->pBuff, “Success”);
    }
    Return OPENSDK_APP_OK;
}
```

All messages to get or set data (**SDK_APP_DATA** command) will be sent to recv_data(void *, void *) in the application.

# 4 References

Refer to ServerPushMJPEGApp/html/index.html for examples of the use of functions.

# 5 Limitations

It is recommended to use the functions of 3.50 and later.