HIKVISION

SOLUTIONS

[Thermal Camera Integration Solution]

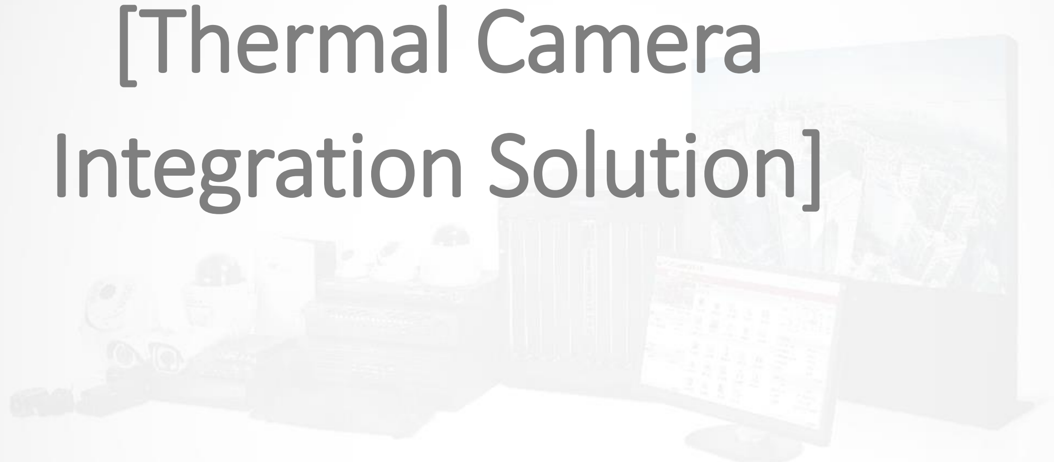


Table of Contents

1 Integration Summary

1.1 Common functions

- Live view
- Playback
- Download
- Event

1.2 Special functions

- Dynamic fire detection
- Temperature alarm
- Temperature difference alarm
- Real-time temperature measurement (Metadata)
- Fire source detection (Metadata)
- Real-time temperature measurement
- Manual temperature measurement

2 Integration Architecture

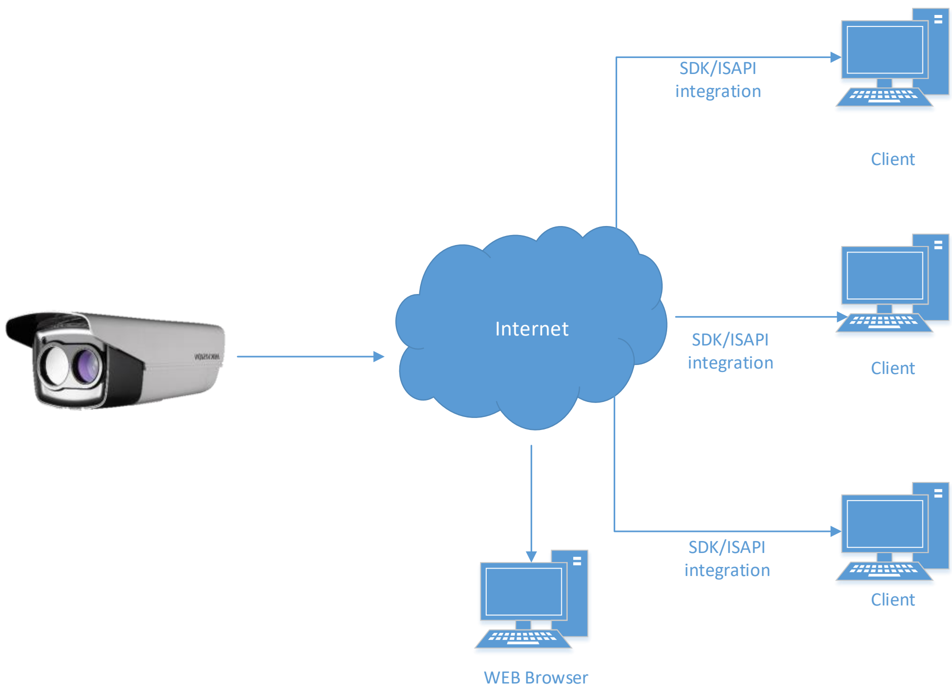


Figure 1-1 Integration Architecture

3 Integration Method

3.1 SDK

The device network SDK is developed based on private network communication protocol, and it is designed for the remote connection and configuration of embedded DVR, Encoder, IPC and the other IP devices.

3.2 ISAPI

ISAPI is a kind of Hikvision protocol, based on HTTP. It is available to all kinds of platforms, such as console application, web application and so on.

4 Live view

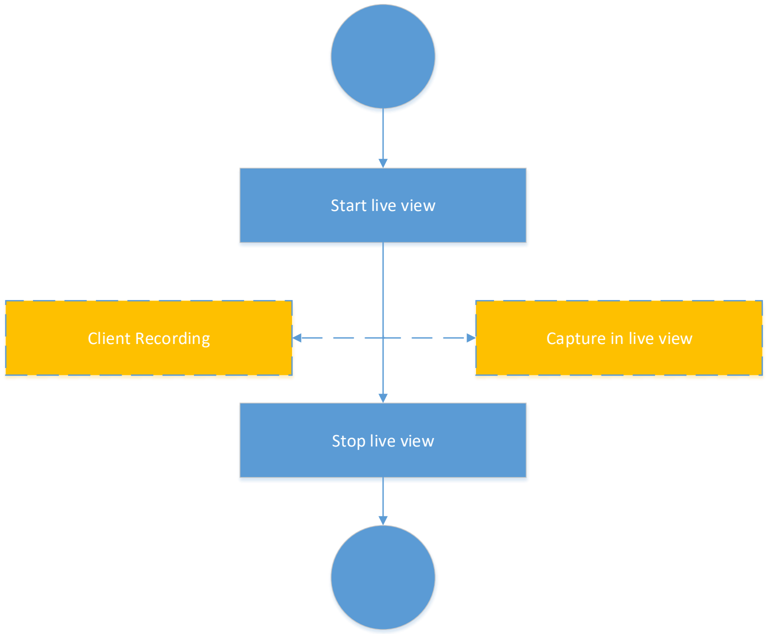
4.1 Description

Live view is common function for the device. It is easy for you to get live view stream from the device and display on your platform. During the live view process, you can also realize other function, such as recoding, capture, voice control, etc. Recording and capture are frequently used, and I will show you the integration method below.

4.2 WEB interface

Figure 1-2 Live view interface

4.3 Integration method for key functions



4.3.1 Live view

1) SDK Integration

SDK interface	Remark
NET_DVR_RealPlay_V40	Start live view
NET_DVR_StopRealPlay	Stop live view

2) ISAPI integration:

Please use RTSP URL below to get live view stream.

URL:rtsp://<address>:<port>/Streaming/Channels/<id>(?parm1=value1&parm2-=value2…)

Remark: There are two channels for the thermal positioning system and thermal stable positioning system, the first channel is visible light channel and the second channel is thermal imaging channel. But there is only one thermal imaging channel for thermal bullet camera.

4.3.2 Recording

1) SDK integration

SDK interface	Remark
NET_DVR_SaveRealData	Video file in PS package format
NET_DVR_SaveRealData_V30	PS package format or 3GPP package format is selectable

2) ISAPI integration

ISAPI doesn't support recording function.

4.3.3 Capture

Start the live view first, and then the player will capture in the decoding stream of the live view. The resolutions of capture and stream are same and the captured picture in BMP and JPEG format are both supported.

1) SDK integration

SDK interface	Remark
NET_DVR_SetCapturePictureMode	Set capture mode, BMP or JPEG
NET_DVR_CapturePictureBlock_New	Blocking capture and the data will be stored in the cache
NET_DVR_CapturePictureBlock	Blocking capture and directly saved as file
NET_DVR_CapturePicture	non-blocking capture and directly saved as file

2) ISAPI integration

Call URL below to get a snapshot from a specified channel.

ISAPI command: /ISAPI/Streaming/channels/<ID>/picture

5 Playback

5.1 Description

This function is used to view the remotely recorded video files stored in the network disks (NAS) or memory cards.

5.2 WEB interface

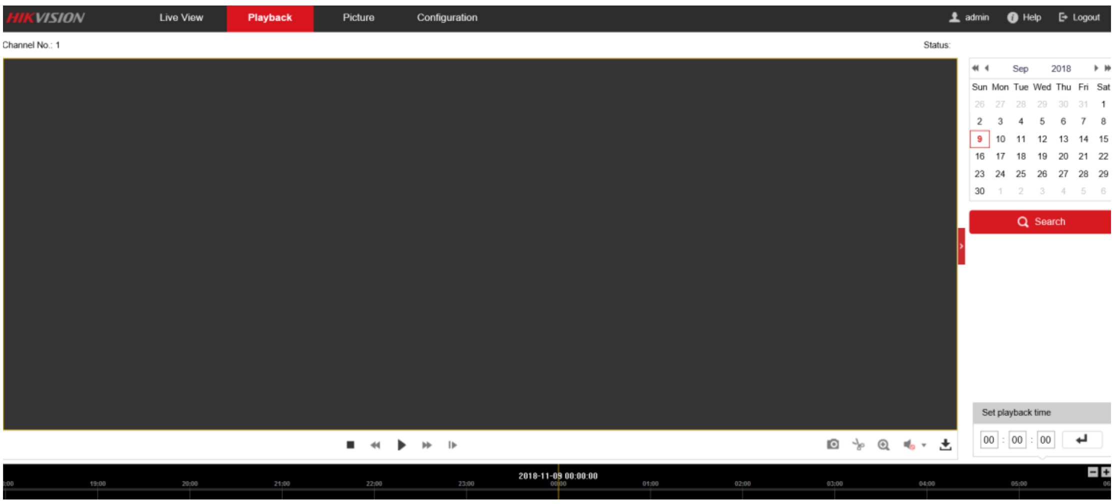
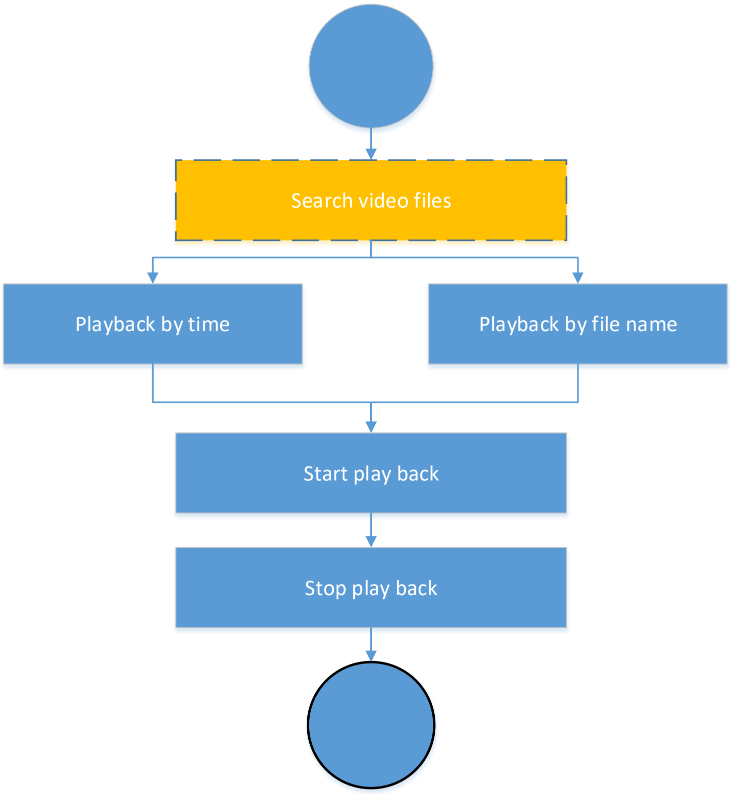


Figure 1-3 Playback interface

5.3 Integration methods for key functions



5.3.1Playback by time

1) SDK integration

Step1: Call **NET_DVR_PlayBackByTime_V40** to start playback according to the time period with videos in the specified time range

Step2: Call **NET_DVR_PlayBackControl_V40** (command: NET_DVR_PLAYSTART) to control the start of playback.

2) ISAPI integration

URL: rtsp://<address>:<port>/Streaming/tracks/<id>(?parm1=value1&parm2-=value2…)

Remark: When playback by time, the user can specify the start time and end time without calling API to search video files. The progress between 0 to 100 is not supported, and progress configuration is also not supported.

5.3.2 Playback by file name

1) SDK Integration

Step1: Call related APIs to obtain file name.

Step2: Call **NET_DVR_PlayBackByName** to start playback by file name.

Step3: Call **NET_DVR_PlayBackControl_V40** (command: NET_DVR_PLAYSTART) to control the start of playback.

2) ISAPI integration

Call URL below for the playback function.

URL: rtsp://<address>:<port>/Streaming/tracks/<id>(?parm1=value1&parm2-=value2…)

6 Download

6.1 Description

After the configuration for the storage in the camera, the video will stored in the specified HDD. Then you can search the video files and download if it is in need.

6.2 WEB interface

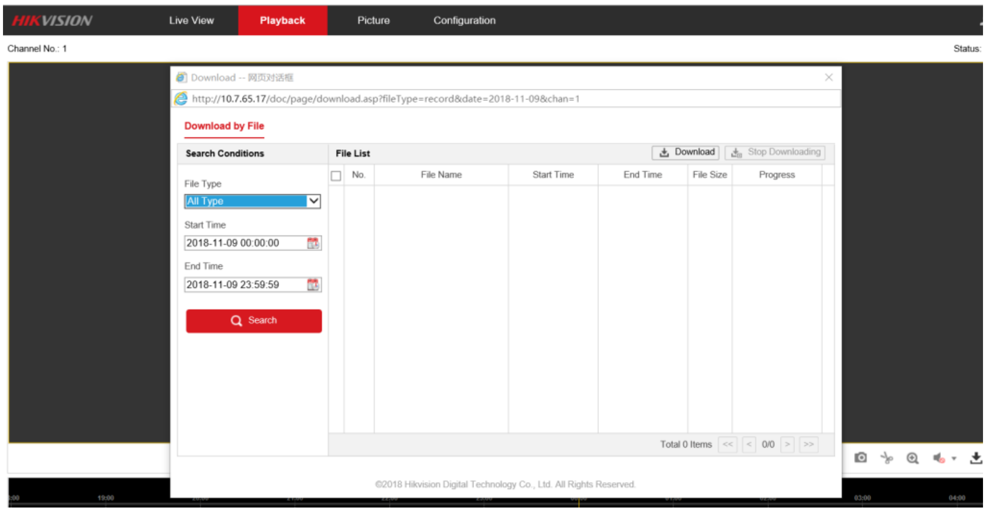
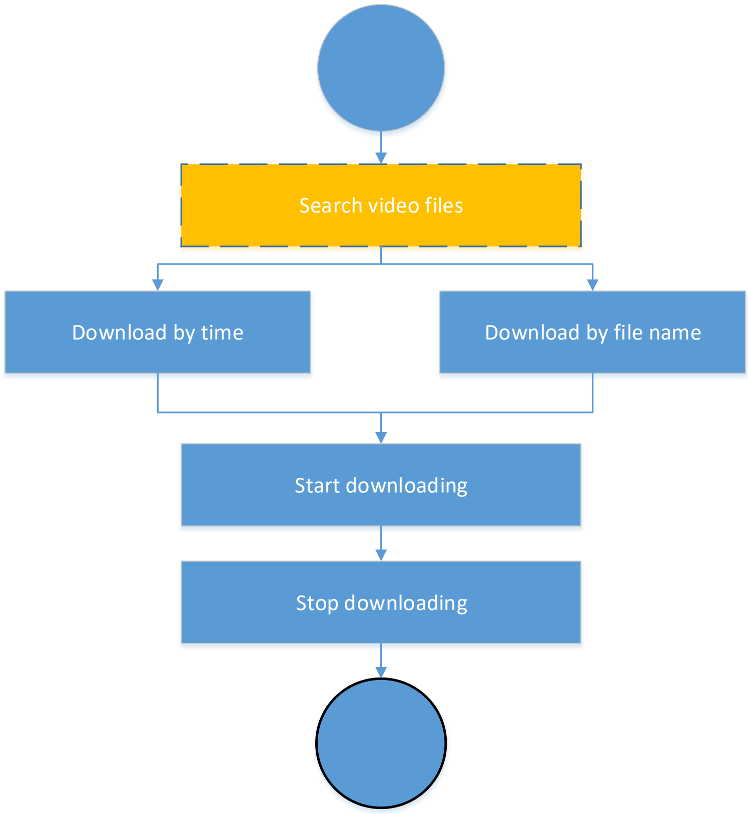


Figure 1-4 Download interface

6.3 Integration methods for key functions



6.3.1Download by time

1) SDK integration

Step1: Call **NET_DVR_GetFileByTime_V40** to start downloading according to the time period with videos in the specified time range.

Step2: Call **NET_DVR_PlayBackControl_V40** (command: NET_DVR_PLAYSTART) to control the start of download.

2) ISAPI integration

ISAPI doesn't support download by time

Remark: When download by time, the user can specify the start time and end time without calling API to search video files. The progress between 0 to 100 is not supported, and progress configuration is also not supported

6.3.2 Download by file name

1) SDK integration

Step1: Call related APIs to obtain file name.

Step2: Call **NET_DVR_GetFileByName** to start downloading by file name.

Step3: Call **NET_DVR_PlayBackControl_V40** (command: NET_DVR_PLAYSTART) to control the start of download.

2) ISAPI integration

Please use URL below to download file by name.

URL: /ISAPI/ContentMgmt/download

7 Event

After the configuration via WEB interface， you can use SDK or ISAPI to receive alarm from the device. Please note that you can receive alarm only in arming mode.

7.1 Common events

7.1.1 Arming mode

1) Description

For the thermal camera, there are many basic events you can receive after the relevant parameters are configured, such as Motion Detection, Video Tampering, Alarm Input, etc. I will take Motion Detection for example.

2) WEB interface configuration

Please see the below configuration for Motion Detection.

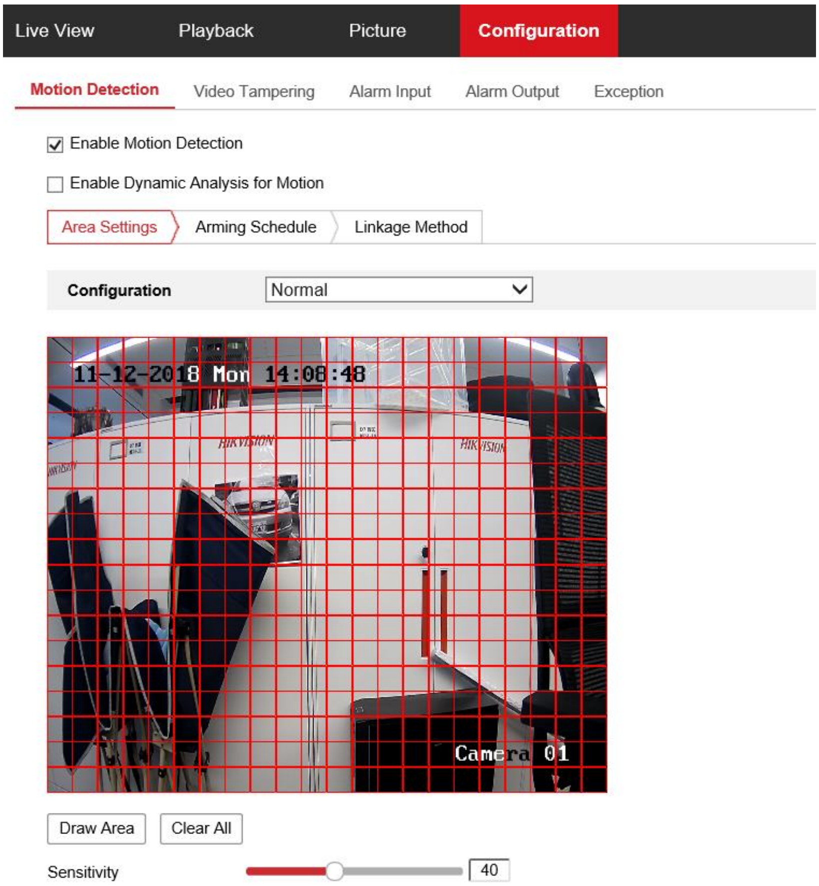


Figure 1-5 Motion detection configuration

3) Integration method

SDK integration

Call **NET_DVR_SetDVRMessageCallBack_V31** to set alarm callback function for receiving and handling alarm information.

Command: COMM_ALARM_V30

Main Structure: NET_DVR_ALARMINFO_V30

Main field

Main field	Remark
dwAlarmType	Alarm type: 0- semaphore alarm, 1- hard disk full, 2- video lost, 3- motion detection, etc
dwAlarmInputNumber	Alarm input port, valid when alarm type is 0 or 23
byAlarmOutputNumber	The triggered alarm output port
byAlarmRelateChannel	Recording channel triggered by alarm: 0- not triggered, 1- triggered.
byChannel	Alarming channel, it is valid when alarm type is 2, 3, 6, 9, 10 or 11
byDiskNumber	Alarming hard disk, it is valid when alarm type is 1, 4, or 5

ISAPI integration

URL: GET /ISAPI/Event/notification/alertStream

Note: A connection is established with the device when this URL is called, and stays open to constantly receive event notifications. If Heartbeat/Network timed out, platform should call this URL to reconnect.

Main field in the returned data

Main field	Remark
channleID	Channel number
dateTime	The event time
eventType	VMD
eventDescription	Motion alarm

A snapshot of Motion Detection

```
Content-Type: application/xml; charset="UTF-8"
Content-Length: 551

<EventNotificationAlert version="2.0"
    xmlns="http://www.hikvision.com/ver20/XMLSchema">
    <ipAddress>10.0.1.64</ipAddress>
    <portNo>80</portNo>
    <protocol>HTTP</protocol>
    <macAddress>94:e1:ac:71:4f:32</macAddress>
    <channelID>1</channelID>
    <dateTime>2018-11-04T08:33:34-8:00</dateTime>
    <activePostCount>16</activePostCount>
    <eventType>VMD</eventType>
    <eventState>active</eventState>
    <eventDescription>Motion alarm</eventDescription>
    <DetectionRegionList>
    </DetectionRegionList>
    <channelName>Camera 01</channelName>
</EventNotificationAlert>
-boundary
```


Remark: Thermal camera doesn't support **listening mode**.

7.2 Special events

7.2.1Dynamic Fire Detection

1) Description

After setting the detection parameters and linkage actions, the device will start scanning and detecting automatically via built-in algorithms. When the fire source is detected, the configured PTZ linkage, recording or alarm will be triggered. SDK can get the uploaded alarm information by arming mode, while the linkage action should be selected as "Upload to Center".

2) WEB interface configuration

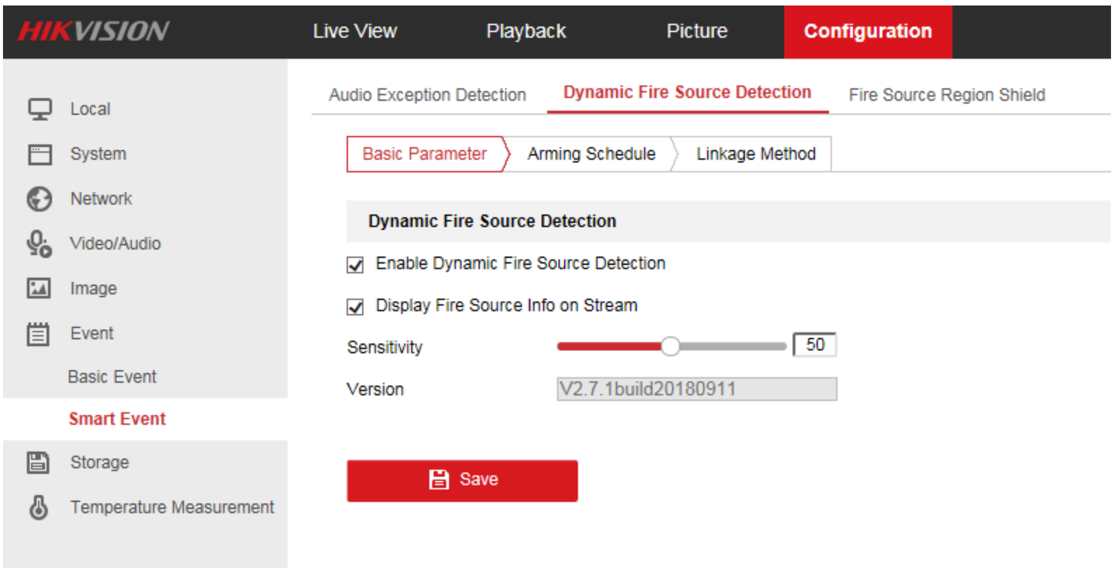


Figure 1-6 Fire detection configuration

3) Integration method

SDK integration

Call **NET_DVR_SetDVRMessageCallBack_V31** to set alarm callback function for receiving and handling alarm information.

Command: COMM_FIREDETECTION_ALARM

Main Structure : NET_DVR_FIREDETECTION_ALARM

Main field

Main field	Remark
wFireMaxTemperature	Fire source highest temperature, value range: [300,4000], unit: degree Centigrade
wTargetDistance	Fire source target distance, value range: [100,10000], Unit: m
byStrategyType	Strategy type: 0- fire or smoke detection, 1- fire and smoke detection, 2- double confirm, 3- specified fire source, 4- specified smoke
byAlarmSubType	Alarm sub type: 0- fire source detection alarm, 1- smoke detection alarm, 2- fire and smoke detection

ISAPI integration

URL: GET /ISAPI/Event/notification/alertStream

Note: If Heartbeat/Network timed out, platform should call this URL to reconnect.

Main field in the returned data

Main field	Remark
dateTime	The event time
eventType	fireDetection
eventDescription	fireDetection alarm
temperatureUnit	Eg.celsius
fireMaxTemperature	The fire max temperature.
targetDistance	The target distance

7.2.2Temperature Alarm

1) Description

After configuring parameters and linkage action, the device will start real-time temperature measurement automatically. When the temperature is higher than the configured alarm threshold, the configured PTZ linkage, recording or alarm will be triggered. SDK will get the alarm information uploaded by device via arming mode, while the linkage action should be set as "Upload to Center".

2) WEB interface configuration

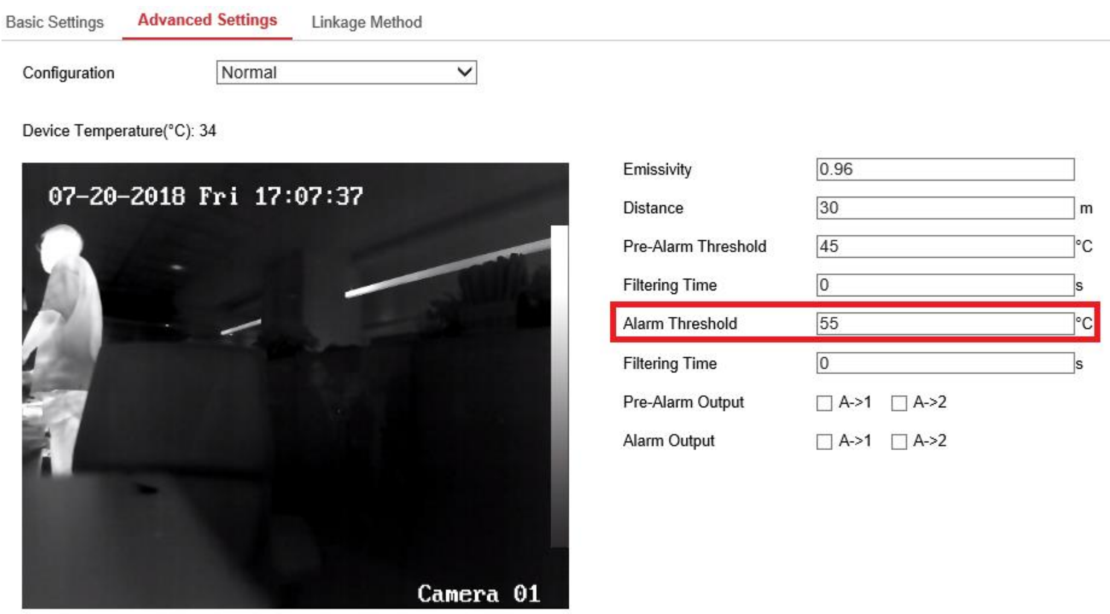


Figure 1-7 Temperature alarm configuration

3) Integration method

SDK integration

Call **NET_DVR_SetDVRMessageCallBack_V31** to set alarm callback function for receiving and handling alarm information.

Command: COMM_THERMOMETRY_ALARM

Main Structure : NET_DVR_THERMOMETRY_ALARM

Main field

Main field	Remark
byThermometryUnit	Temperature Unit :0- Degrees Centigrade, 1- Degrees Fahrenheit, 2- Degree Kelvin
byAlarmLevel	Alarm Level :0- Pre-Alarm, 1- Alarm
byAlarmType	Alarm Type :0- Highest Temperature, 1- Lowest Temperature, 2- Average Temperature
fCurrTemperature	Current Temperature :Corrects to one decimal place, range from -40 to 1000

ISAPI integration

URL: GET /ISAPI/Event/notification/alertStream

Note: If Heartbeat/Network timed out, platform should call this URL to reconnect.

Main field in the returned data

Main field	Remark
dateTime	The event time
eventType	TMA,TMPA
eventDescription	Temperature Measurement Alarm
thermometryUnit	Eg.celsius
ruleTemperature	Configured temperature
currTemperature	Current temperature

ruleCalibType	Calib type .eg. region
ruleType	Eg.highest temp is higher than

7.2.3Temperature Difference Alarm

1) Description

After configuring parameters and linkage action, the device will start real-time temperature measurement automatically. When the temperature difference is higher than the configured alarm threshold, the configured PTZ linkage, recording or alarm will be triggered. SDK will get the alarm difference information uploaded by device via arming mode, while the linkage action should be set as "Upload to Center".

2) WEB interface configuration

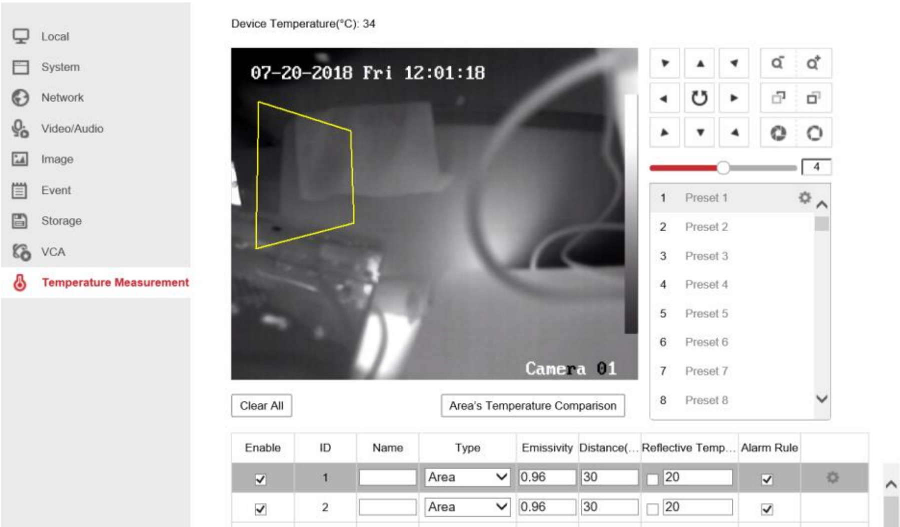


Figure 1-8 Temperature difference alarm configuration 1

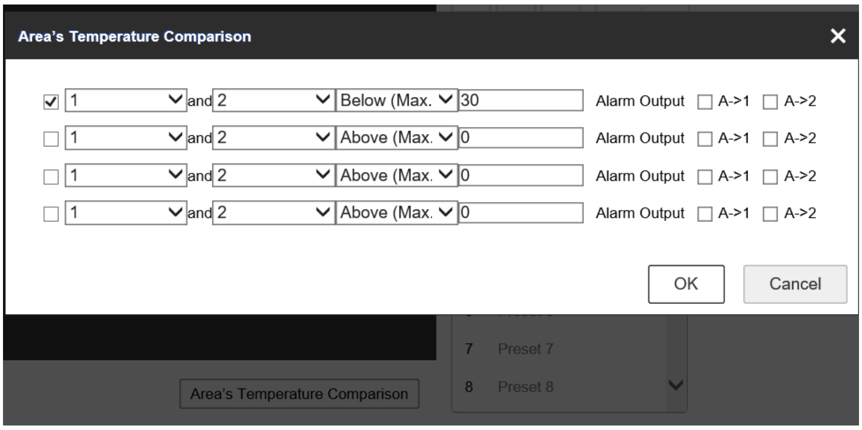


Figure 1-9 Temperature difference alarm configuration 2

3) Integration method

SDK integration

Call **NET_DVR_SetDVRMessageCallBack_V31** to set alarm callback function for receiving and handling alarm information.

Command: COMM_THERMOMETRY_DIFF_ALARM
Main Structure : NET_DVR_THERMOMETRY_DIFF_ALARM
Main fields

Description	Main field	Remark
Alarm1	byAlarmID1	Alarm1
Alarm2	byAlarmID2	Alarm2
Alarm Level	byAlarmLevel	Alarm Level: 0- Pre-Alarm, 1- Alarm
Alarm Type	byAlarmType	Alarm Type: 0- Highest Temperature, 1- Lowest Temperature, 2- Average Temperature
Current temperature difference	fRuleTemperatureDiff	Current temperature difference, corrects to one decimal place, range from -40 to 1000, (floating number +100)

ISAPI integration
URL: GET /ISAPI/Event/notification/alertStream
Note: If Heartbeat/Network timed out, platform should call this URL to reconnect.
Main field in the returned data

Main field	Remark
dateTime	The event time
eventType	TMA,TPMA
eventDescription	Temperature Measurement Precautionary Alarm
thermometryUnit	Eg.celsius
ruleTemperature	Configured temperature
currTemperature	Current temperature
ruleCalibType	Calib type. eg. region
ruleType	Eg. temp diff is higher than

8 Metadata

Metadata is the data that describes other data. In the video monitoring filed, metadata usually contains intelligent structured information (e.g., real-time temperature measurement, fire source detection) which can help users to understand the videos.

8.1 Real-time Temperature Measurement

1) The description of the upload data

Description	value	Remark
tempValue	xs:float	The temperature value.
tempUnit	centigrade	The temperature unit.
	fahrenheit	
	kelvin	
tempProperty	highest	The property of temperature.
	lowest	
	average	

2) The example of real-time temperature measurement in metadata


```
<Metadata version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <type>activityTarget</type>
  <subType>thermometry</subType>
  <time>2017-01-01T00:00:00+08:00</time>
  <DevInfo>
    <ipAddress>192.168.0.64</ipAddress>
    <portNo>554</portNo>
    <macAddress>28:57:be:ee:d1:5a</macAddress>
    <channel>1</channel>
  </DevInfo>
  <TargetDetection>
    <TargetList>
      <Target>
        <ruleID>1</ruleID>
        <RegionList>
          <Region>
            <Point>
              <x>100</x>
              <y>100</y>
             </Point>
             <Point>
               <x>200</x>
               <y>200</y>
             </Point>
          </Region>
        </RegionList>
        <PropertyList>
          <Property>
            <description>tempValue</description>
            <value>30</value>
```

```xml
      </Property>
      <Property>
        <description>tempUnit</description>
        <value>centigrade</value>
      </Property>
      <Property>
        <description>tempProperty</description>
        <value>average</value>
      </Property>
    </PropertyList>
  </Target>
  <Target>
    <ruleID>1</ruleID>
    <RegionList>
      <Region>
        <Point>
          <x>100</x>
          <y>100</y>
        </Point>
      </Region>
    </RegionList>
    <PropertyList>
      <Property>
        <description>tempValue</description>
        <value>20</value>
      </Property>
      <Property>
        <description>tempUnit</description>
        <value>centigrade</value>
      </Property>
      <Property>
        <description>tempProperty</description>
        <value>lowest</value>
      </Property>
    </PropertyList>
  </Target>
  <Target>
    <ruleID>1</ruleID>
    <RegionList>
      <Region>
        <Point>
          <x>200</x>
          <y>200</y>
        </Point>
```

```
            </Region>
          </RegionList>
          <PropertyList>
            <Property>
              <description>tempValue</description>
              <value>40</value>
            </Property>
            <Property>
              <description>tempUnit</description>
              <value>centigrade</value>
            </Property>
            <Property>
              <description>tempProperty</description>
              <value>highest</value>
            </Property>
          </PropertyList>
        </Target>
      </TargetList>
    </TargetDetection>
</Metadata>
```


3) Integration method

Key steps

Step1: Configure metadata parameters, including enabling intelligent event and setting message format, via the URL: PUT /ISAPI/Streaming/channels/<ID>/metadata.

Step2: Request for getting stream for live view based on RTSP protocol method via the URL: DESCRIBE rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

Step3: Parse the obtained stream data to judge whether the stream contains the metadata information and get the trackID of metadata if contained.

Step4: Request for getting detailed metadata information by RTSP protocol method via the URL: SETUP rtsp://<ipaddress>/ISAPI/Streaming/channels/<ID>/trackID=MetadataID.

Step5: Start the live view by RTSP protocol method via the URL: PLAY rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

Step6: Judge whether the type value of uploaded RTP packet is 0x6b according to the description of RTP packet or the value of parameter payload type in RTP packet header.

Step7: Parse the metadata information in the RTP packet and display the intelligent information on the video.

Step8: Stop the live view by RTSP protocol method via the URL: TEARDOW rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

For more details about the metadata integration, please refer to the Metadata Application (ISAPI Protocol) Programming Manual for the integration.

8.2 Fire Source Detection

1) The description of the upload data

Description	Value	Remark
distance	xs:integer	The distance from fire source.Unit:m
tempValue	xs:float	The temperature value
tempUnit	centigrade	The temperature unit.
	fahrenheit	
	kelvin	
tempProperty	highest	Currently only support highest temperature

2) The example of fire source detection in metadata


```xml
<Metadata version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <type>activityTarget</type>
  <subType>fireDetection</subType>
  <time>2017-01-01T00:00:00+08:00</time>
  <DevInfo>
    <ipAddress>192.168.0.64</ipAddress>
    <portNo>554</portNo>
    <macAddress>28:57:be:ee:d1:5a</macAddress>
    <channel>1</channel>
  </DevInfo>
  <TargetDetection>
    <TargetList>
      <Target>
        <RegionList>
          <Region>
            <Point>
              <x>0</x>
              <y>0</y>
            </Point>
            <Point>
              <x>600</x>
              <y>0</y>
            </Point>
            <Point>
              <x>600</x>
              <y>600</y>
```

```
            </Point>
            <Point>
              <x>0</x>
              <y>600</y>
            </Point>
          </Region>
        </RegionList>
        <PropertyList>
          <Property>
            <description>distance</description>
            <value>1000</value>
          </Property>
        </PropertyList>
      </Target>
      <Target>
        <RegionList>
          <Region>
            <Point>
              <x>300</x>
              <y>300</y>
            </Point>
          </Region>
        </RegionList>
        <PropertyList>
          <Property>
            <description>tempValue</description>
            <value>100</value>
          </Property>
          <Property>
            <description>tempUnit</description>
            <value>centigrade</value>
          </Property>
          <Property>
            <description>tempProperty</description>
            <value>highest</value>
          </Property>
        </PropertyList>
      </Target>
    </TargetList>
  </TargetDetection>
</Metadata>
```


3) Integration method

Key steps

Step1: Configure metadata parameters, including enabling intelligent event and setting message format, via the URL: PUT /ISAPI/Streaming/channels/<ID>/metadata.

Step2: Request for getting stream for live view based on RTSP protocol method via the URL: DESCRIBE rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

Step3: Parse the obtained stream data to judge whether the stream contains the metadata information and get the trackID of metadata if contained.

Step4: Request for getting detailed metadata information by RTSP protocol method via the URL: SETUP rtsp://<ipaddress>/ISAPI/Streaming/channels/<ID>/trackID=MetadataID.

Step5: Start the live view by RTSP protocol method via the URL: PLAY rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

Step6: Judge whether the type value of uploaded RTP packet is 0x6b according to the description of RTP packet or the value of parameter payload type in RTP packet header.

Step7: Parse the metadata information in the RTP packet and display the intelligent information on the video.

Step8: Stop the live view by RTSP protocol method via the URL: TEARDOW rtsp://<ipaddress>/ISAPI/Streaming/channels/ID.

For more details about the metadata, please refer to the Metadata Application (ISAPI Protocol) Programming Manual for the integration.

9 Temperature Measurement

9.1 Real-time Temperature Measurement

1) Description

After you configure the temperature measurement parameters in normal mode or expert mode, you can call relevant API to receive real-time temperature information in callback function.

2) Integration method

SDK integration

Call NET_DVR_StartRemoteConfig to establish long connection, the detected real-time temperature information will be returned in callback function configured by long connection API.

Command: NET_DVR_GET_REALTIME_THERMOMETRY

Main structure: NET_DVR_THERMOMETRY_UPLOAD

Main fields

Main field	Remark
szRuleName	Rule name
byRuleCalibType	Rule calibration type: 0- Point, 1- Frame, 2- Line
byThermometryUnit	Unit: 0- Degree Centigrade, 1- Degree Fahrenheit, 2- Kelvin

byDataType	Data status type: 0- Measuring, 1- Started, 2-Ended
bySpecialPointThermType	Whether supports measuring temperature by special point
fCenterPointTemperature	Medium Temperature, which is accurate to one decimal place.
fHighestPointTemperature	Highest Temperature, which is accurate to one decimal place.
fLowestPointTemperature	Lowest Temperature, which is accurate to one decimal place.

9.2 Manual Temperature Measurement

1) Description

After you configure the basic parameter of manual temperature measurement , such as distance ,emissivity and so on , you can call relevant API to get measurement results in the callback function.

2) Integration method

SDK integration

Call **NET_DVR_StartRemoteConfig** to enable manual test and set callback function, get the measurement results from the callback function.

Command: NET_DVR_GET_MANUALTHERM_INFO

Main Structure : NET_SDK_MANUAL_THERMOMETRY

Main fields

Main field	Remark
dwAbsTime	Absolute time mark (read-only)
byThermometryUnit	Temperature measurement unit: 0- degrees Centigrade, 1- degrees Fahrenheit, 2- Kelvin
byDataType	Data status type: 0- detecting, 1- start, 2- end (read-only)
byRuleCalibType	Rule calibration type: 0-point, 1- frame, 2- line
struPointTherm	Point temperature measurement, valid when calibration type is 0
struRegionTherm	Region temperature measurement, valid when calibration type is 1 or 2