Device Network SDK (Display and Control)

Developer Guide

Legal Information

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE DOCUMENT IS PROVIDED "AS IS" AND “WITH ALL FAULTS AND ERRORS”. OUR COMPANY MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IN NO EVENT WILL OUR COMPANY BE LIABLE FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES, INCLUDING, AMONG OTHERS, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF DATA, CORRUPTION OF SYSTEMS, OR LOSS OF DOCUMENTATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, IN CONNECTION WITH THE USE OF THE DOCUMENT, EVEN IF OUR COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS.

Contents

Chapter 1 Overview

1.1 Introduction

This document provides multiple integration applications of display and control devices based on Device Network SDK, such as LED screen, video wall controller, decoder, remote control, and so on. The APIs used in the applications are also listed here for reference.

1.2 Update History

Summary of Changes in Version 6.1.4.35_July, 2020

Related Product: Decoder with Model DS-6901UDI, DS-6904UDI, DS-6908UDI, DS-6910UDI, DS-6912UDI, DS-6916UDI in Software Version 2.6.0

1. Extended the video output parameter structure of video wall ***NET_DVR_WALLOUTPUTPARAM*** (related API: ***NET_DVR_GetDeviceConfig***):
   added a background color type "0xff" (custom) to the member **byBackgroundColor**;
   added a member **struBackColor** (background color).
2. Extended the dynamic decoding parameter structure ***NET_DVR_PU_STREAM_CFG_V41*** (related API: ***NET_DVR_MatrixStartDynamic_V41***):
   added two members **byStreamEncrypt** (whether to encrypt the stream) and **sStreamPassword** (stream encryption password).
3. Extended the structure about remote decoding and playback parameters on video wall ***NET_DVR_MATRIX_DEC_REMOTE_PLAY_V50*** (related API: ***NET_DVR_RemoteControl***):
   added two members **byStreamEncrypt** (whether to encrypt the stream) and **sStreamPassword** (stream encryption password).

Summary of Changes in Version 6.1.4.X_July, 2020

Fixed some bugs in the document.

Summary of Changes in Version 5.3.5.5_May, 2018

New document.

Chapter 2 Typical Applications

2.1 Configure Video Wall and Open Roaming Window

Before performing any operations, you must configure the video wall, that is, adding signal sources, adding the video outputs, linking the video outputs to the jointed screens, and adjusting the position of jointed screens on video wall. Windowing is to open a new window on the screen(s). The window can be within a screen or span multiple screens. You can move the window on the valid screens as desired and this function is called roaming.

Before You Start

- Make sure you have called ***NET_DVR_Init*** to initialize the development environment.
- Make sure you have called ***NET_DVR_Login_V40*** to log in to the device.

Steps

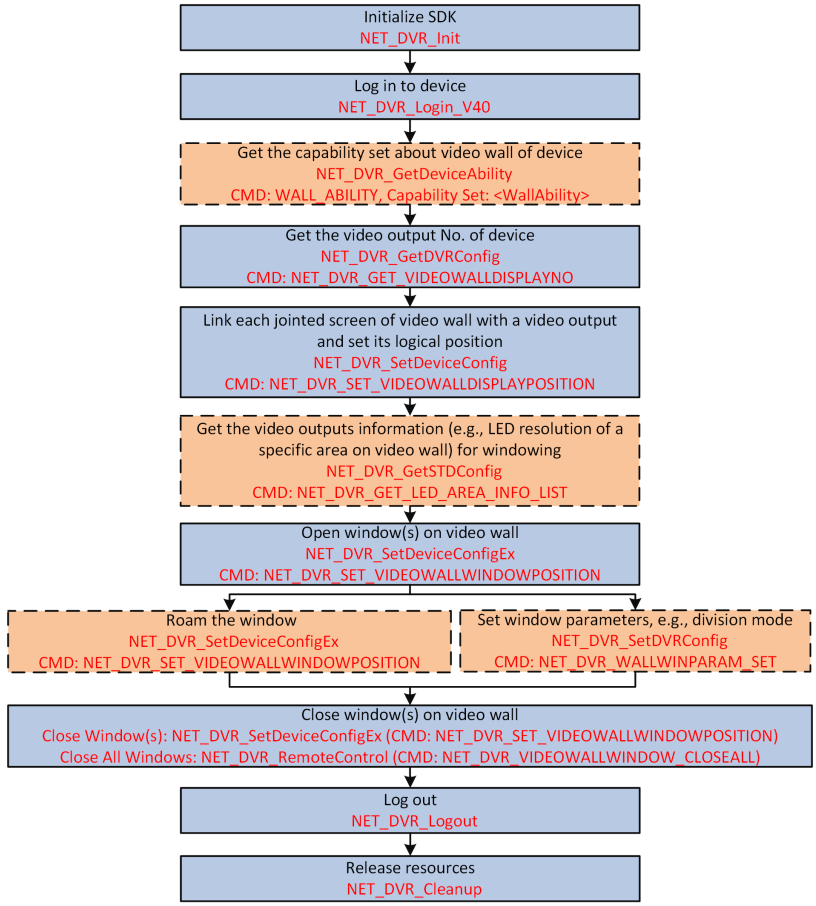

Figure 2-1 API Calling Flow of Configuring Video Wall and Opening Roaming Window

1. **Optional:** Call ***NET_DVR_GetDeviceAbility*** with "WALL_ABILITY" (command No.: 0x212) to get the capability to check whether the function is supported by device.

The video wall capability is returned in the message ***XML_WallAbility*** by **pOutBuf**.

2. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_GET_VIDEOWALLDISPLAYNO" (command No.: 1732) to get the video output No. of device for linking to jointed screens.

   The video output information is returned in the structure ***NET_DVR_DISPLAYCFG*** by **lpOutBuffer**.

3. **Optional:** Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VIDEOWALLDISPLAYPOSITION" (command No.: 1734) and set **lpInBuffer** to the video input No. for getting the default or configured relation between video input No. and jointed screen for reference.

   The relation information is returned in the structure ***NET_DVR_VIDEOWALLDISPLAYPOSITION*** by **lpOutBuffer**.

4. Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_SET_VIDEOWALLDISPLAYPOSITION" (command No.: 1733), set **lpInBuffer** to the video input No., and set to **lpInParamBuffer** to the structure ***NET_DVR_VIDEOWALLDISPLAYPOSITION*** for linking each jointed screen of video wall with a video output and setting the logical position of screens.

5. **Optional:** Call ***NET_DVR_GetSTDConfig*** with "NET_DVR_GET_LED_AREA_INFO_LIST" (command No.: 9295) and set **lpCondBuffer** to the structure ***NET_DVR_LED_AREA_COND*** for getting default or configured video output information (e.g., LED resolution of a specific area on video wall) for reference.

   The video output information is returned in the structure ***NET_DVR_LED_AREA_INFO_LIST*** by **lpOutBuffer**.

6. Call ***NET_DVR_SetDeviceConfigEx*** with "NET_DVR_SET_VIDEOWALLWINDOWPOSITION" (command No.: 1736) and set **lpInParam** to the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** for opening window(s) on video wall.

   Note

   Two windowing modes are available: 0-windowing by coordinates, 1-windowing by configured resolution.

   The window opening status will be returned by **lpStatusList** of **lpOutParam**.

7. **Optional:** Perform the following operation(s) after windowing.

   Roaming

   Call ***NET_DVR_SetDeviceConfigEx*** with "NET_DVR_SET_VIDEOWALLWINDOWPOSITION" (command No.: 1736) and set **lpInParam** to the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** for roaming the window.

   Note

   The window roaming status will be returned by **lpStatusList** of **lpOutParam**.

   Set Window Parameters

   a. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_WALLWINPARAM_GET" (command No.: 9006) to get default or configured window parameters for reference.

The window parameters are returned in the structure ***NET_DVR_WALLWINPARAM*** by **lpOutBuffer**.

b. Call ***NET_DVR_SetDVRConfig*** with "NET_DVR_WALLWINPARAM_SET" (command No.: 9005) and set **lpInBuffer** to the structure ***NET_DVR_WALLWINPARAM*** for setting the window parameters, e.g., division mode.

8. Perform one of the following operations to close the window(s) on video wall.

- Call ***NET_DVR_SetDeviceConfigEx*** with "NET_DVR_SET_VIDEOWALLWINDOWPOSITION" (command No.: 1736) and set **lpInParam** to the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** for closing one or multiple window(s).
- Call ***NET_DVR_RemoteControl*** with "NET_DVR_VIDEOWALLWINDOW_CLOSEALL" (command No,: 1737) and set **lpInBuffer** to 4-byte window No. for closing all windows.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

2.2 Display Live Video on Video Wall

For dynamic decoding, the decoder will automatically connect to the encoding devices for requesting for data stream. This decoding mode is usually applied to decode the live video from the cameras and display on video wall.

Before You Start

- Make sure you have called ***NET_DVR_Init*** and ***NET_DVR_Login_V40*** to initialize the development environment and log in to the device.
- Make sure you have added signal sources, added video outputs, and linked the video outputs to the jointed screens to configure the video wall, see details in ***Configure Video Wall and Open Roaming Window*** for details.

Steps

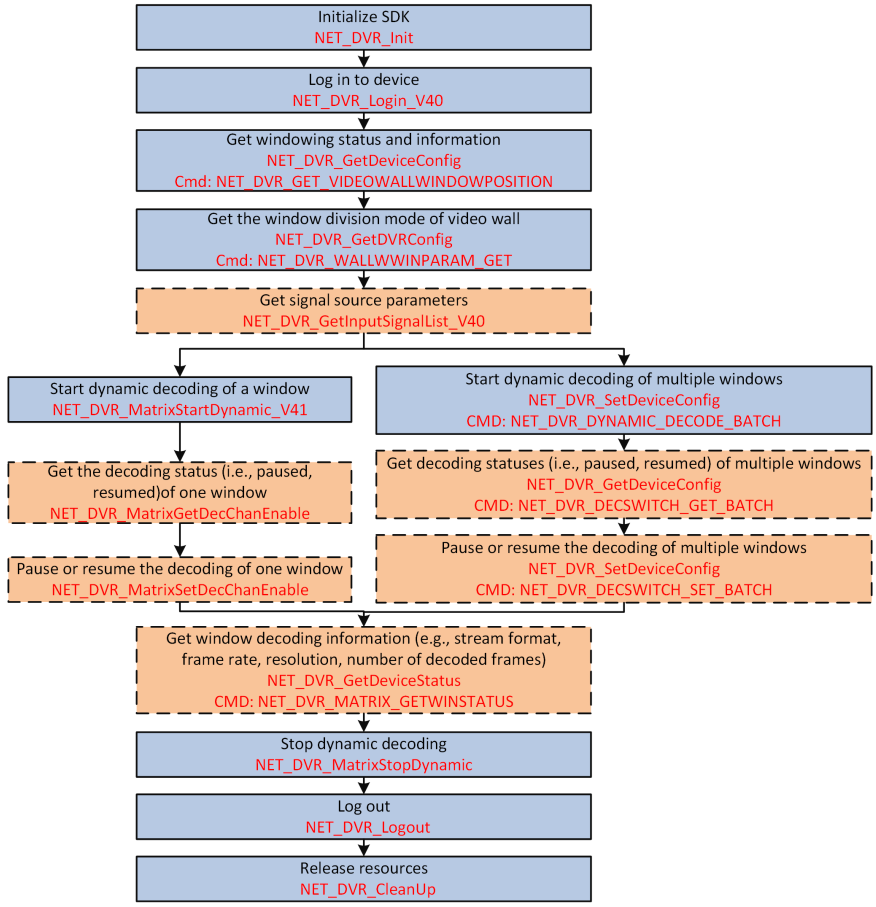

Figure 2-2 API Calling Flow of Displaying Live Video on Video Wall

1. Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VIDEOWALLWINDOWPOSITION" (command No.: 1735) and set **lpInBuffer** to a 4-byte video wall No. for getting windowing status and related information.

   The status and information are returned in the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** by **lpOutBuffer**.

2. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_WALLWINPARAM_GET" (command No.: 9005) and set **lChannel** to a window No. for getting window division mode of video wall. : structure) to get the .

   The division mode is returned in the structure ***NET_DVR_WALLWINPARAM*** by **lpOutBuffer**.

3. **Optional:** Call ***NET_DVR_GetInputSignalList_V40*** to get the local signal source parameters for starting dynamic decoding.

Note

The API in the above step is only used for getting the local signal source parameters. For starting dynamic decoding of network signal source, the following parameters are required: IP address, port No., user name, password, channel No., and stream type, which are provided by users.

4. Perform the following operation(s) to start the dynamic decoding of a window or multiple windows.

Start dynamic decoding of a window

a. Call ***NET_DVR_MatrixStartDynamic_V41*** to start one window's dynamic decoding.

b. Call ***NET_DVR_MatrixSetDecChanEnable*** to pause or resume the dynamic decoding of one window.

Note

Before setting the dynamic decoding status, you'd better call ***NET_DVR_MatrixGetDecChanEnable*** to get the current status (i.e., paused or resumed) of the window for reference.

Start dynamic decoding of multiple mindows

a. Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_DYNAMIC_DECODE_BATCH" (command No.:1769), set **lpInBuffer** to a 4-byte window No., and set **lpInParamBuffer** to the structure ***NET_DVR_PU_STREAM_CFG_V41*** for getting start dynamic decoding of multiple windows.

b. Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_DECSWITCH_SET_BATCH" (command No.: 1770), set **lpInBuffer** to a 4-byte window No., and set **lpInParamBuffer** to a 4-byte operation command (0 or 1) for pausing or resuming the decoding of multiple windows.

Note

Before setting the dynamic decoding statuses of multiple windows, you'd better call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_DECSWITCH_GET_BATCH" (command No.: 1771) and set **lpInBuffer** to a 4-byte window No. for getting the current statuses (i.e., paused or resumed) of multiple windows.

5. **Optional:** Call ***NET_DVR_GetDeviceStatus*** with "NET_DVR_MATRIX_GETWINSTATUS" (command No.: 9009) and set **lpInBuffer** to the structure ***NET_DVR_WALLWIN_INFO*** for getting the decoding information (e.g., stream format, frame rate, resolution, number of decoded frames) of the current window.

   The decoding information is returned in the structure ***NET_DVR_WALL_WIN_STATUS*** by **lpOutBuffer**.

6. Call ***NET_DVR_MatrixStopDynamic*** to stop the dynamic decoding.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

2.3 Display Playback on Video Wall

The videos stored in remote storage devices can be dynamically decoded and displayed on the video wall.

Before You Start

- Make sure you have called ***NET_DVR_Init*** and ***NET_DVR_Login_V40*** to initialize the development environment and log in to the device.
- Make sure you have added signal sources, added video outputs, and linked the video outputs to the jointed screens to configure the video wall, see details in ***Configure Video Wall and Open Roaming Window*** for details.

Steps

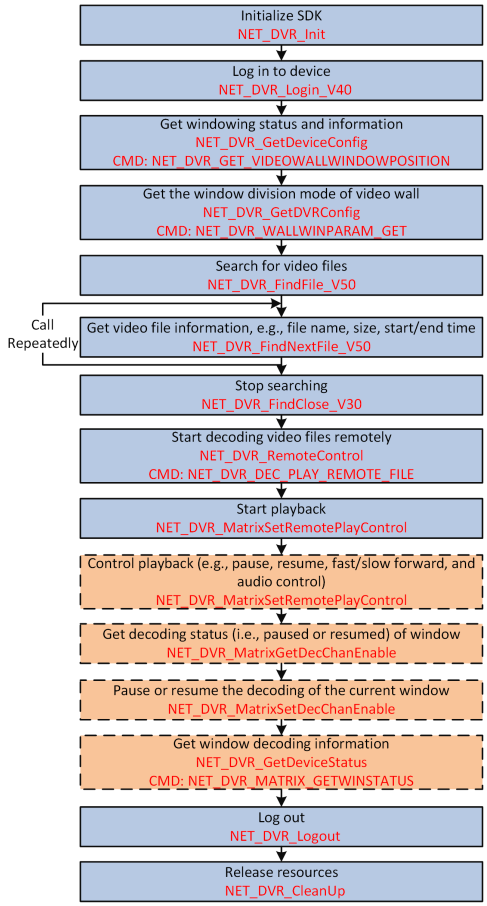

Figure 2-3 API Calling Flow of Displaying Playback on Video Wall

1. Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VIDEOWALLWINDOWPOSITION" (command No.: 1735) and set **lpInBuffer** to a 4-byte window No. for getting the windowing status and related information.

   The status and information are returned in the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** by **lpOutBuffer**.

2. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_WALLWINPARAM_GET" (command No.: 9006) and set **lChannel** to a window No. for getting the window division mode of video wall.

   The window division mode is returned in the structure ***NET_DVR_WALLWINPARAM*** by **lpOutBuffer**.
3. Call ***NET_DVR_FindFile_V50*** to remotely search for video files.
4. Call ***NET_DVR_FindNextFile_V50*** repeatedly to get the searched video information, such as file name, size, start and end time.
5. Call ***NET_DVR_FindClose_V30*** to stop searching for video files.
6. Call ***NET_DVR_RemoteControl*** with "NET_DVR_DEC_PLAY_REMOTE_FILE" (command No.: 9027) and set **lpInBuffer** to the structure ***NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX*** for starting decoding the video files remotely.
7. Call ***NET_DVR_MatrixSetRemotePlayControl*** to start playback.
8. **Optional:** Perform the following operation(s) after starting decoding videos and displaying playback on video wall.

   Control playback
   Call ***NET_DVR_MatrixSetRemotePlayControl*** to pause, resume, perform fast or slow forward, and control audio during playback.

   Get decoding status
   Call ***NET_DVR_MatrixGetDecChanEnable*** to get the dynamic decoding status (i.e., paused or resumed) of the current window.

   Set decoding status
   Call ***NET_DVR_MatrixSetDecChanEnable*** to pause or resume the dynamic decoding of the current window.

   Note
   Before setting the decoding status, you'd better call ***NET_DVR_MatrixGetDecChanEnable*** to get the current decoding status for reference.

   Get decoding information
   Call ***NET_DVR_GetDeviceStatus*** with "NET_DVR_MATRIX_GETWINSTATUS" (command No.: 9009) and set **lpInBuffer** to the structure ***NET_DVR_WALLWIN_INFO***) for getting the decoding information (e.g., stream format, frame rate, resolution, number of decoded frames) of the current window.

   The decoding information is returned in the structure ***NET_DVR_WALL_WIN_STATUS*** by **lpOutBuffer**.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

2.4 Configure Auto-Switch Decoding

For multiple signal sources, you can decode and display the videos on video wall in auto-switch mode. The auto-switch decoding mode is also a dynamic decoding mode, which automatically switches the display of signal sources' videos in a specific time interval.

Before You Start

- Make sure you have called ***NET_DVR_Init*** and ***NET_DVR_Login_V40*** to initialize the development environment and log in to the device.
- Make sure you have added signal sources, added video outputs, and linked the video outputs to the jointed screens to configure the video wall, see details in ***Configure Video Wall and Open Roaming Window*** for details.

Steps

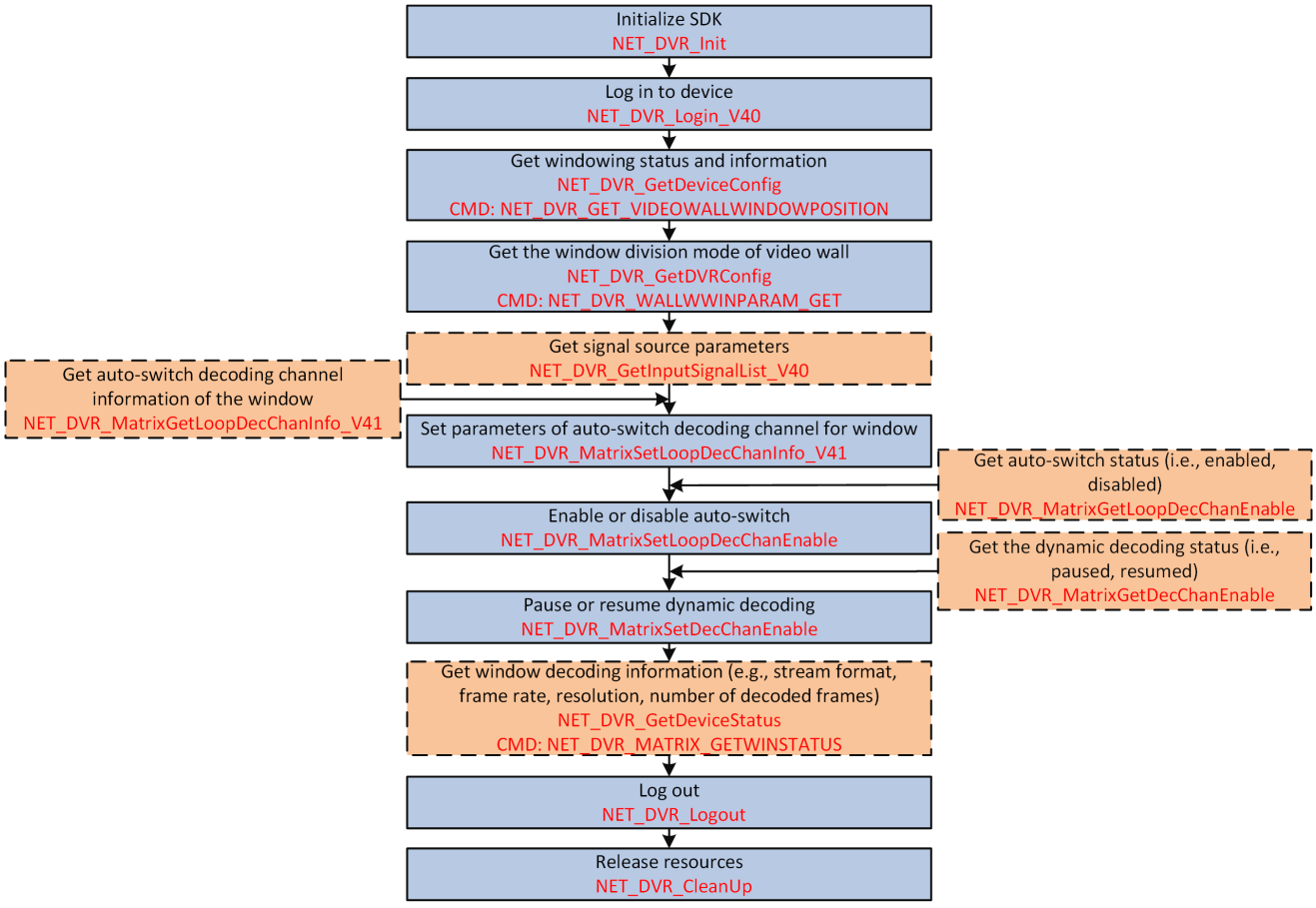

Figure 2-4 API Calling Flow of Configuring Auto-Switch Decoding

1. Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VIDEOWALLWINDOWPOSITION" (command No.: 1735) and set **lpInBuffer** to a 4-byte video wall No. for getting windowing status and related information.

The status and information are returned in the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** by **lpOutBuffer**.

2. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_WALLWINPARAM_GET" (command No.: 9005) and set **lChannel** to a window No. for getting the window division mode of video wall.

   The window division mode is returned in the structure ***NET_DVR_WALLWINPARAM*** by **lpOutBuffer**.
3. **Optional:** Call ***NET_DVR_GetInputSignalList_V40*** to get the signal source parameters.
4. Call ***NET_DVR_MatrixSetLoopDecChanInfo_V41*** to set the parameters of auto-switch decoding channel for window.

   Note

   Before setting auto-switch channel parameters, you'd better call ***NET_DVR_MatrixGetLoopDecChanInfo_V41*** to get the current configurations for reference.
5. Call ***NET_DVR_MatrixSetLoopDecChanEnable*** to enable or disable auto-switch.

   Note

   Before setting auto-switch status, you'd better call ***NET_DVR_MatrixGetLoopDecChanEnable*** to get the current status for reference.
6. Call ***NET_DVR_MatrixSetDecChanEnable*** to pause or resume the dynamic decoding.

   Note

   - Before setting dynamic decoding status, you'd better call ***NET_DVR_MatrixGetDecChanEnable*** to get the current status for reference.
   - If the auto-switch is disabled, but the dynamic decoding is resumed, the current decoding mode of this window is dynamic decoding. Only when the auto-switch and dynamic decoding are both enabled or resumed, the decoding mode turns to auto-switch decoding.
7. **Optional:** Call ***NET_DVR_GetDeviceStatus*** with "NET_DVR_MATRIX_GETWINSTATUS" (command No.: 9009) and set **lpInBuffer** to the structure ***NET_DVR_WALLWIN_INFO*** for getting the decoding information (e.g., stream format, frame rate, resolution, number of decoded frames) of the current window.

   The decoding information is returned in the structure ***NET_DVR_WALL_WIN_STATUS*** by **lpOutBuffer**.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

2.5 Configure Passive Decoding

For passive decoding, the decoder will not send decoding request, the data should be sent to the decoder for decoding, so this decoding mode is usually applied to the situation of decoding and displaying local videos on video wall.

Before You Start

- Make sure you have called ***NET_DVR_Init*** and ***NET_DVR_Login_V40*** to initialize the development environment and log in to the device.
- Make sure you have added signal sources, added video outputs, and linked the video outputs to the jointed screens to configure the video wall, see details in ***Configure Video Wall and Open Roaming Window*** for details.

Steps

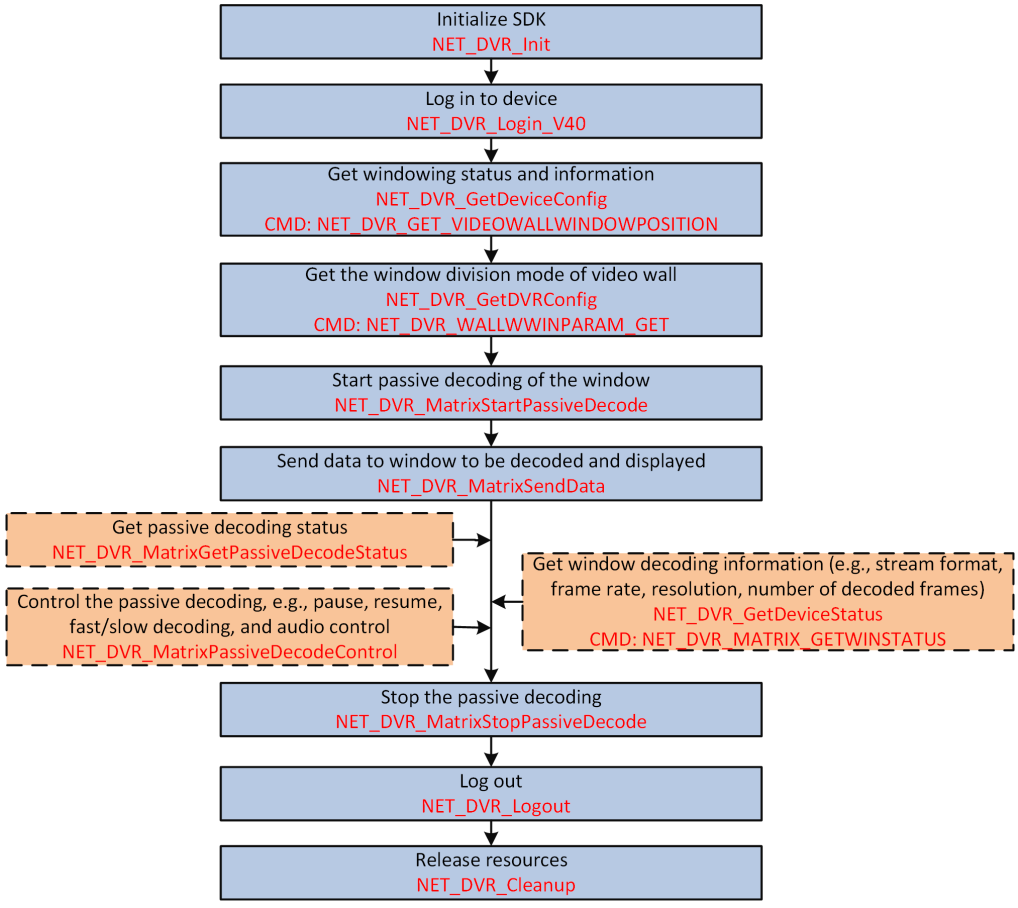

Figure 2-5 API Calling Flow of Configuring Passive Decoding

1. Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VIDEOWALLWINDOWPOSITION" (command No.: 1735) and set **lpInBuffer** to a 4-byte video wall No. for getting windowing status and related information.

The status and information are returned in the structure ***NET_DVR_VIDEOWALLWINDOWPOSITION*** by **lpOutBuffer**.

2. Call ***NET_DVR_GetDVRConfig*** with "NET_DVR_WALLWINPARAM_GET" (command No.: 9005) and set **lChannel** to a window No. for getting the window division mode of video wall.

   The window division mode is returned in the structure ***NET_DVR_WALLWINPARAM*** by **lpOutBuffer**.
3. Call ***NET_DVR_MatrixStartPassiveDecode*** to start passive decoding of the window.
4. Call ***NET_DVR_MatrixSendData*** to send data to window to be decoded and displayed.
5. **Optional:** Perform the following operation(s) after starting passive decoding and displaying.

   Get passive decoding status: Call ***NET_DVR_MatrixGetPassiveDecodeStatus*** to get the status (e.g., paused, resumed) of passive decoding.

   Control passive decoding: Call ***NET_DVR_MatrixPassiveDecodeControl*** to pause/resume passive decoding, perform fast/slow passive decoding, and control audio.

   Note

   Before controlling passive decoding, you'd better call ***NET_DVR_MatrixGetPassiveDecodeStatus*** to get the current passive decoding status for reference.

   Get window decoding information: Call ***NET_DVR_GetDeviceStatus*** with "NET_DVR_MATRIX_GETWINSTATUS" (command No.: 9009) and set **lpInBuffer** to the structure ***NET_DVR_WALLWIN_INFO*** for getting the decoding information (e.g., stream format, frame rate, resolution, number of decoded frames) of the current window.

   The decoding information is returned in the structure ***NET_DVR_WALL_WIN_STATUS*** by **lpOutBuffer**.
6. Call ***NET_DVR_MatrixStopPassiveDecode*** to stop the passive decoding.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

2.6 Switch Scene

A scene contains the configuration information of video wall, such as window size, position, window division mode, decoding sources, and so on. You can add different scenes or save the views displayed on video wall as scenes, and then, you call the configured scenes and switch scenes to easily view the required videos on the video wall.

Before You Start

- Make sure you have called ***NET_DVR_Init*** and ***NET_DVR_Login_V40*** to initialize the development environment and log in to the device.
- Make sure you have added signal sources, added video outputs, and linked the video outputs to the jointed screens to configure the video wall, see details in ***Configure Video Wall and Open Roaming Window*** for details.

Steps

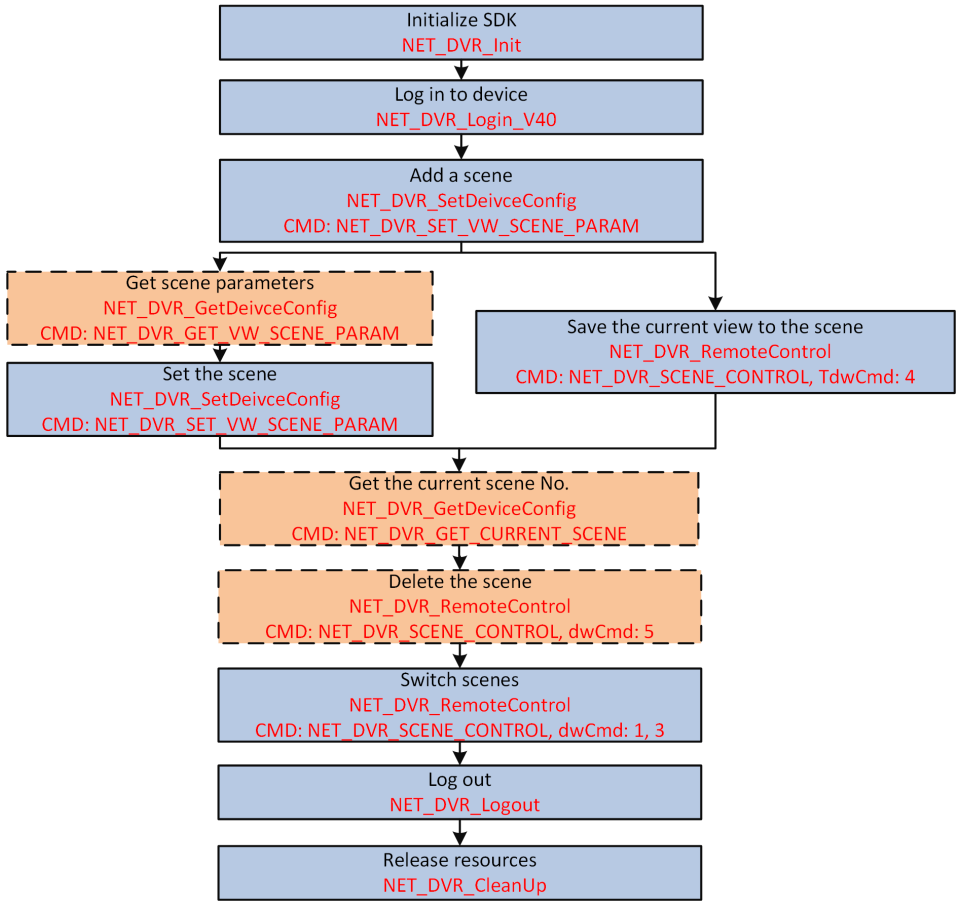

Figure 2-6 API Calling Flow of Switching Scene

1. Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_SET_VW_SCENE_PARAM" (command No.: 1747), set **lpInBuffer** to the structure ***NET_DVR_VIDEO_WALL_INFO*** , and set **lpInParamBuffer** to the structure ***NET_DVR_WALLSCENECFG*** for adding a scene.
2. Perform one of the following operations to set the scene.
   - Call ***NET_DVR_SetDeviceConfig*** with "NET_DVR_SET_VW_SCENE_PARAM" (command No.: 1747), set **lpInBuffer** to the structure ***NET_DVR_VIDEO_WALL_INFO*** , and set **lpInParamBuffer** to the structure ***NET_DVR_WALLSCENECFG*** for editing the scene parameters.

Note

Before setting the scene parameters, you'd better call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_VW_SCENE_PARAM" (command No.: 1746) and set **lpInBuffer** to the structure ***NET_DVR_VIDEO_WALL_INFO*** for getting the current scene parameters for reference.

- Call ***NET_DVR_RemoteControl*** with "NET_DVR_SCENE_CONTROL" (command No.: 1744) and set **lpInBuffer** to the structure ***NET_DVR_SCENE_CONTROL_INFO*** for saving the current view to the added scene.

  Note

  The parameter **dwCmd** in the structure ***NET_DVR_SCENE_CONTROL_INFO*** should be set to 4.

3. **Optional:** Call ***NET_DVR_GetDeviceConfig*** with "NET_DVR_GET_CURRENT_SCENE" (command No.: 1745) and set **lpInBuffer** to a 4-byte scene No. for getting the information of the current scene.

   The scene information is returned in the structure ***NET_DVR_VIDEO_WALL_INFO*** by **lpOutBuffer**.

4. **Optional:** Call ***NET_DVR_RemoteControl*** with "NET_DVR_SCENE_CONTROL" (command No.: 1744) and set **lpInBuffer** to the structure ***NET_DVR_SCENE_CONTROL_INFO*** for deleting the scene.

   Note

   The parameter **dwCmd** in the structure ***NET_DVR_SCENE_CONTROL_INFO*** should be set to 5.

5. Call ***NET_DVR_RemoteControl*** with "NET_DVR_SCENE_CONTROL" (command No.: 1744) and set **lpInBuffer** to the structure ***NET_DVR_SCENE_CONTROL_INFO*** for switching the scenes.

   Note

   The parameter **dwCmd** in the structure ***NET_DVR_SCENE_CONTROL_INFO*** should be set to 1 or 3.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

Chapter 3 API Reference

3.1 NET_DVR_Cleanup

Release the resources after the program is ended.

API Definition

```
BOOL NET_DVR_Cleanup(
);
```

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The available error codes may be returned by this API are 0 and 3. See details in ***Device Network SDK Errors*** .

Remarks

- When calling this API, you cannot call other APIs at the same time.
- ***NET_DVR_Init*** and this API should be called by pair. That is, once the NET_DVR_Init is called, you should call NET_DVR_Cleanup to release the resources when exiting the program.

3.2 NET_DVR_FindClose_V30

Stop searching for files and release resources.

API Definition

```
BOOL NET_DVR_FindClose_V30(
  LONG   lFindHandle
);
```

Parameters

lFindHandle

[IN] Handle for searching files, which is returned by ***NET_DVR_FindFile_V50*** or ***NET_DVR_FindFileByEvent_V50*** .

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API are shown as the follows: 0, 3, 12, and 17.
See error details in ***Device Network SDK Errors*** .

3.3 NET_DVR_FindFile_V50

Search for video files by file type or by time.

API Definition

```
LONG NET_DVR_FindFile_V50(
  LONG                lUserID,
  NET_DVR_FILECOND_V50    pFindCond
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

pFindCond

[IN] File information structure to be found, see details in the structure ***NET_DVR_FILECOND_V50*** .

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API: 0, 2, 3, 4, 7, 8, 9, 10, 12, 17, 19, 23, 41, 43, 44, 47, 72, and 73. For error details, refer to ***Device Network SDK Errors*** .

Remarks

This API specifies the video files to be found. After calling this API, you can call ***NET_DVR_FindNextFile_V50*** to get the file details.

See Also

NET_DVR_FindClose_V30

3.4 NET_DVR_FindFileByEvent_V50

Search for video files by event.

API Definition

```
LONG NET_DVR_FindFileByEvent_V50(
  LONG                        lUserID,
  NET_DVR_SEARCH_EVENT_PARAM_V50    lpSearchEventParam
);
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

lpSearchEventParam

[IN] Structure of video file information to search, see details in the structure ***NET_DVR_SEARCH_EVENT_PARAM_V50*** .

Return Values

Return *-1* for failure, and return other value as the handle for playback.
If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- After searching video files by event, you can call ***NET_DVR_FindNextEvent_V50*** to get the file information.
- The video files searched by event only support playback by time.

3.5 NET_DVR_FindNextEvent_V50

Get video files one by one after searching by event.

API Definition

```
LONG NET_DVR_FindNextEvent_V50(
  LONG                        lFindHandle,
  NET_DVR_SEARCH_EVENT_RET_V50    lpSearchEventRet
);
```

Parameters

lFindHandle

[IN] Handle for playback, which is returned by ***NET_DVR_FindFileByEvent_V50*** .

lpSearchEventRet

[OUT] Search results, see details in the structure ***NET_DVR_SEARCH_EVENT_RET_V50*** .

Return Values

Return *-1* for failure, and return other values as the current getting status, see details in the following table.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting file information succeeded.
NET_DVR_FILE_NOFIND	1001	No file found.

Status	Value	Description
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more file found. Search ended.
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND _TIMEOUT	1005	Search timed out.

If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

The video files searched by event only support playback by time.

3.6 NET_DVR_FindNextFile_V50

Get video files one by one after searching by time.

API Definition

```
LONG NET_DVR_FindNextFile_V50(
  LONG                lFindHandle,
  NET_DVR_FINDDATA_V50    pFindData
);
```

Parameters

lFindHandle

[IN] Handle for playback, which is returned by ***NET_DVR_FindFile_V50*** .

pFindData

[OUT] Search results, see details in the structure ***NET_DVR_FINDDATA_V50*** .

Return Values

Return *-1* for failure, and return other values as the current getting status, see details in the following table.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting file information succeeded.
NET_DVR_FILE_NOFIND	1001	No file found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more file found. Search ended.

Status	Value	Description
NET_DVR_FILE_EXCEPTION	1004	Search exception.
NET_DVR_FIND _TIMEOUT	1005	Search timed out.

If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- If you want to get all searched video files, you should repeatedly call this API.
- You can also get the linked card No. of current video file and the information of whether the file is locked via this API.
- Up to 4000 files can be searched for once.

3.7 NET_DVR_GetDeviceAbility

Get the device capabilities.

API Definition

```
BOOL NET_DVR_GetDeviceAbility(
  LONG    lUserID,
  DWORD   dwAbilityType,
  char    *pInBuf,
  DWORD   dwInLength,
  char    *pOutBuf,
  DWORD   dwOutLength
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwAbilityType

[IN] Capability types, which are different according to different devices and functions.

pInBuf

[IN] Input parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

dwInLength

[IN] Size of input buffer.

pOutBuf

[OUT] Output parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

dwOutLength

[OUT] Size of buffer for receiving data.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.8 NET_DVR_GetDeviceConfig

Get device configuration information in batch (with sending data).

API Definition

```
BOOL NET_DVR_GetDeviceConfig(
  LONG     lUserID,
  DWORD    dwCommand,
  DWORD    dwCount,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize,
  LPVOID   lpStatusList,
  LPVOID   lpOutBuffer,
  DWORD    dwOutBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device getting commands. The commands are different for different getting functions.

dwCount

[IN] Number of configurations (cameras) to get at a time. 0, 1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 64 cameras' configuration information can be obtained at a time.

lpInBuffer

[IN] Pointer of configuration condition buffer, which specifies the number (**dwCount**) of configurations to get, and relates to the getting commands.

dwInBufferSize

[IN] Size of configuration condition buffer, which saves the obtained configuration information (the number is **dwCount**).

lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras need to search, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0 or 1, it refers to getting succeeded, otherwise, this parameter value is the error code.

lpOutBuffer

[OUT] Parameters returned by device, which relates to the getting commands. And there is a one-to-one correspondence between the parameters and the cameras need to search.

If the **lpStatusList** of one camera is larger than 1, the corresponding **lpOutBuffer** is invalid.

dwOutBufferSize

[IN] Total size of returned results (the number is **dwCount**).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If returns *TRUE*, it does not mean that all configurations are obtained, you can check the value of **lpStatusList[n]** to judge which one is succeeded.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_SetDeviceConfig

3.9 NET_DVR_GetDeviceStatus

Call this API to get device status.

API Definition

```
BOOL NET_DVR_GetDeviceStatus(
  LONG     lUserID,
  DWORD    dwCommand,
  DWORD    dwCount,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize,
  LPVOID   lpStatusList,
  LPVOID   lpOutBuffer,
  DWORD    dwOutBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Commands for getting the device status.

dwCount

[IN] Number of devices' statuses to be obtained.

lpInBuffer

[IN] Condition buffer of getting status, which is different acoording to different commands (**dwCommand**).

dwInBufferSize

[IN] Condition buffer size.

lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras that need to be searched, e.g. **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0, it refers to getting succeeded; if the value is larger than 0, it indicates getting failed.

lpOutBuffer

[OUT] Status details returned by device, which veries with the command (**dwCommand**), and one-to-one corresponds to the cameras that need to be searched.

If the **lpStatusList** value of a camera is larger than 0, the corresponding **lpOutBuffer** is invalid.

dwOutBufferSize

[IN] Output buffer size

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not mean that all settings are succeeded, you can check the value of **lpStatusList[n]** to judge which one is succeeded. If returning failed, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- If you want to get all devices' statuses, you should set the **dwCount** to 0xffffffff, set **lpInBuffer** to NULL, set **dwInBufferSize** to 0, set **lpStatusList** to NULL.
- For **lpOutBuffer**, the first 4-byte is the total number of structures returned by device, and the following bytes contain the structures details. If the configured output buffer is insufficient, only a part of structures will be returned.

3.10 NET_DVR_GetDVRConfig

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetDVRConfig(
  LONG      lUserID,
  DWORD     dwCommand,
  LONG      lRuleID,
  LONG      lChannel,
  LPVOID    lpOutBuffer,
  DWORD     dwOutBufferSize,
  LPDWORD   lpBytesReturned
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device getting commands, which are different according to different getting functions.

lRuleID

[IN] Rule ID.

lChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xffffffff-invalid or all channels, 1-main NIC, 2-extended NIC.

lpOutBuffer

[OUT] Pointer of buffer to receive data. For different getting functions, the structures of this parameter are different.

dwOutBufferSize

[IN] Size of buffer to receive data (unit: byte). It cannot be 0.

lpBytesReturned

[OUT] Pointer of actually received data size. It cannot be NULL.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the ***Device Network SDK Errors*** .

See Also

NET_DVR_SetDVRConfig

3.11 NET_DVR_GetErrorMsg

Return the error information of the last operation.

API Definition

```
char *NET_DVR_GetErrorMsg(
  LONG   *pErrorNo
);
```

Parameters

pErrorNo

[OUT] Error code pointer.

Return Values

The return values are the pointers of error information, see ***Device Network SDK Errors*** for details.

Remarks

You can call ***NET_DVR_GetLastError*** to get the error codes.

3.12 NET_DVR_GetInputSignalList_V40

Get the signal source list.

API Definition

```
BOOL NET_DVR_GetInputSignalList_V40(
  LONG                  lUserID,
  DWORD                 dwDevNum,
  NET_DVR_INPUT_SIGNAL_LIST  *lpInputSignalList
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDevNum

Reserved, set to 0.

lpInputSignalList

[OUT] Signal source list, see details in the structure ***NET_DVR_INPUT_SIGNAL_LIST*** .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.13 NET_DVR_GetLastError

Return the error code of the last operation.

API Definition

```
DWORD NET_DVR_GetLastError(
);
```

Return Values

The return values are error codes, see ***Device Network SDK Errors*** for details.

Remarks

You can also call ***NET_DVR_GetErrorMsg*** to directly get the error information.

3.14 NET_DVR_GetSDKLocalCfg

Get the HCNetSDK's local configuration parameters.

API Definition

```
BOOL NET_DVR_GetSDKLocalCfg(
  NET_SDK_LOCAL_CFG_TYPE    enumType,
  void                  *lpOutBuff
);
```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different parameters, see details in ***NET_SDK_LOCAL_CFG_TYPE*** .

lpOutBuff

[OUT] Output parameters. For different configuration options, the structures of output parameters are different, see details in ***NET_SDK_LOCAL_CFG_TYPE*** .

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_SetSDKLocalCfg

3.15 NET_DVR_GetSTDConfig

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetSTDConfig(
  LONG                lUserID,
  DWORD               dwCommand,
  NET_DVR_STD_CONFIG  lpConfigParam
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **lpCondBuffer** and **lpOutBuffer** in the **lpConfigParam** are also different. See the structure ***NET_DVR_STD_CONFIG*** for details.

Note

When getting configuration parameters, the **lpInBuffer** in the **lpConfigParam** is invalid, you can set it to NULL.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_SetSTDConfig

3.16 NET_DVR_Init

Initialize the programming environment before calling other APIs.

API Definition

```
BOOL NET_DVR_Init(
);
```

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The available error codes of this API are 0, 41, and 53. See details in ***Device Network SDK Errors*** .

Remarks

Before initializing, you can call ***NET_DVR_SetSDKInitCfg*** to set the initialization parameters, such as supported capabilities, loading path of component libraries (only supported by Linux system), and so on.

See Also

NET_DVR_Cleanup

3.17 NET_DVR_Login_V40

Log in to the device (supports asynchronous login).

API Definition

```
LONG NET_DVR_Login_V40(
  NET_DVR_USER_LOGIN_INFO   pLoginInfo,
  NET_DVR_DEVICEINFO_V40    lpDeviceInfo
);
```

Parameters

pLoginInfo

[IN] Login parameters, including device address, user name, password, and so on. See details in the structure ***NET_DVR_USER_LOGIN_INFO*** .

lpDeviceInfo

[OUT] Device information. See details in the structure ***NET_DVR_DEVICEINFO_V40*** .

Return Values

- For asynchronous login, the callback function (***fLoginResultCallBack***) configured in the structure (***NET_DVR_USER_LOGIN_INFO***) returns the asynchronous login status, user ID and device information.
- For synchronous login, this API returns *-1* for logging failed, and returns other values for the returned user IDs. The user ID is unique, and it helps to realize the further device operations.
- If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- When **bUseAsynLogin** in **pLoginInfo** is 0, it indicates that login is in synchronous mode; when **bUseAsynLogin** in **pLoginInfo** is 1, it indicates that login is in asynchronous mode.
- Up to 2048 users are allowed to log in to HCNetSDK at same time, and the values of returned **UserID** are ranging from 0 to 2047.

See Also

NET_DVR_Logout

3.18 NET_DVR_LogoSwitch

Control LOGO display.

API Definition

```
BOOL NET_DVR_LogoSwitch(
  LONG   lUserID,
  DWORD  dwDecChan,
  DWORD  dwLogoSwitch
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

dwDecChan

[IN] Decoding channel No.

dwLogoSwitch

[IN] Switch on/off command. 1-NET_DVR_SHOWLOGO, display LOGO; 2-NET_DVR_HIDELOGO, hide LOGO.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.19 NET_DVR_Logout

Log out from devices.

API Definitions

```
BOOL NET_DVR_Logout(
  LONG    lUserID
);
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The available error codes may be returned by this API are 0, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in ***Device Network SDK Errors*** .

3.20 NET_DVR_MatrixGetCurrentSceneMode

Get the current scene.

API Definition

```
BOOL NET_DVR_MatrixGetCurrentSceneMode(
  LONG   lUserID,
  DWORD  *dwSceneNum
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

dwSceneNum

[OUT] Scene No.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.21 NET_DVR_MatrixGetDecChanCfg

Get configuration information of decoding channel.

API Definition

```
BOOL NET_DVR_MatrixGetDecChanCfg(
  LONG                          lUserID,
  DWORD                         dwDecChan,
  LPNET_DVR_MATRIX_DECCHAN_CONTROL   lpInter
)
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChan

[IN] Decoding channel No.

lpInter

[OUT] Parameters of decoding channel scaling control, see details in ***NET_DVR_MATRIX_DECCHAN_CONTROL*** .

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.22 NET_DVR_MatrixGetDecChanEnable

Get the decoding status of the current decoding channel or window.

API Definition

```
BOOL NET_DVR_MatrixGetDecChanEnable(
  LONG     lUserID,
  DWORD    dwDecChanNum,
  DWORD    *lpdwEnable
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or window No.

lpdwEnable

[OUT] Status: 0-Disabledd, 1-Enabled

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixSetDecChanEnable

3.23 NET_DVR_MatrixGetDeviceStatus_V41

Get status of decoding devices.

API Definition

```
BOOL NET_DVR_MatrixGetDeviceStatus_V41(
  LONG                          lUserID,
  LPNET_DVR_DECODER_WORK_STATUS_V41  lpDecoderCfg
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

lpDecoderCfg

[OUT] Device status parameters, see details in NET_DVR_DECODER_WORK_STATUS_V41.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.24 NET_DVR_MatrixGetLoopDecChanEnable

Get the auto-switch status of a decoding channel.

API Definition

```
BOOL NET_DVR_MatrixGetLoopDecChanEnable(
  LONG     lUserID,
  DWORD    dwDecChanNum,
  DWORD    *lpdwEnable
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

Decoding channel No. or window No.

lpdwEnable

[OUT] 0-Disabled, 1-Enabled.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixSetLoopDecChanEnable

3.25 NET_DVR_MatrixGetLoopDecChanInfo_V41

Get the parameters of auto-switch decoding channel.

API Definition

```
BOOL NET_DVR_MatrixGetLoopDecChanInfo_V41(
  LONG                          lUserID,
  DWORD                         dwDecChanNum,
  NET_DVR_MATRIX_LOOP_DECINFO_V41    *lpOuter
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or window No.

lpOuter

[OUT] Parameters of auto-switch decoding channel, see details in the structure ***NET_DVR_MATRIX_LOOP_DECINFO_V41***

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixSetLoopDecChanInfo_V41

3.26 NET_DVR_MatrixGetPassiveDecodeStatus

Get the passive decoding status of the decoder.

API Definition

```
LONG NET_DVR_MatrixGetPassiveDecodeStatus(
  LONG     lPassiveHandle
);
```

Parameters

lPassiveHandle

[IN] Passive decoding handle, which is returned by ***NET_DVR_MatrixStartPassiveDecode*** .

Return Values

Return *-1* for failure, and return the following values for other statuses: *1* (sent), *2* (sending paused), *3* (sending resumed), *4* (error), *5* (heartbeat information).
If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

If you immediately call this API after calling ***NET_DVR_MatrixStartPassiveDecode*** to start passive decoding, failure may occur as the implement of function costs time, and the corresponding error name is "NET_DVR_ORDER_ERROR", you can try again later.

3.27 NET_DVR_MatrixGetRemotePlayStatus

Get playback status.

API Definition

```
BOOL NET_DVR_MatrixGetRemotePlayStatus(
  LONG                              lUserID,
  DWORD                             dwDecChanNum,
  LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS  lpOuter
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel.

lpOuter

[OUT] Playback status, see details in ***NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS*** .

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- You can get playback status and control playback after connecting multi-channel decoder with the front-end device and starting playback by file name or by time. If you play back by time, you cannot control the playback progress.
- Delay exists since the playback command is executed by forwarding stream. Therefore, it's better not to call this API too frequently, and you should take network status into consideration.
- When you play back by time, you can only get the playback status of a single clip instead of the whole video of the time period. You can check whether the playback has finished by the member **dwCurDataType** in the structure ***NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS*** .

3.28 NET_DVR_MatrixGetTranInfo_V30

Get information of transparent channel.

API Definition

```
BOOL NET_DVR_MatrixGetTranInfo_V30(
  LONG                              lUserID,
  LPNET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30  lpTranInfo
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

lpTranInfo

[OUT] Transparent channel parameters, see details in ***NET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30*** .

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- This transparent channel is configured between decoder and front-end device instead of between the client and decoder.
- Multi-channel decoder does not support to build RS232 or RS485 transparent channel with the client.
- Local serial port of multi-channel decoder can only be connected as serial port control platform (by RS232) or as devices to input information such as keyboard (by RS232 or RS285).

3.29 NET_DVR_MatrixPassiveDecodeControl

Control the passive decoding.

API Definition

```
BOOL NET_DVR_MatrixPassiveDecodeControl(
  LONG                          lUserID,
  DWORD                         dwDecChanNum,
  NET_DVR_PASSIVEDECODE_CONTROL     *lpInter
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40***

dwDecChanNum

[IN] Decoding channel No.

lpInter

[IN] Parameters for controlling decoding, see details in the structure ***NET_DVR_PASSIVEDECODE_CONTROL*** .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.30 NET_DVR_MatrixSceneControl

Control scene switch.

API Definition

```
BOOL NET_DVR_MatrixSceneControl(
  LONG   lUserID,
  DWORD  dwSceneNum,
```

```
  DWORD  dwCmd,
  DWORD  dwCmdParam
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

dwSceneNum

[IN] Scene No., which can be got from the capability set.

dwCmd

[IN] Command type: 1-scene switch, 2-initializing the scene (clear the current configuration), 3-force to swich scene, 4-save the scene.

dwCmdParam

[IN] Command parameter, reserved.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.31 NET_DVR_MatrixSendData

Send data to passive decoding channel.

API Definition

```
BOOL NET_DVR_MatrixSendData(
  LONG    lPassiveHandle,
  char    *pSendBuf,
  DWORD   dwBufSize
);
```

Parameters

lPassiveHandle

[IN] Value returned by ***NET_DVR_MatrixStartPassiveDecode***

pSendBuf

[IN] Buffer for the data that needs to be sent.

lpInter

[IN] Size of buffer for the data that needs to be sent. The data size should be smaller than 512 KB, and 30 KB is suggested.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.32 NET_DVR_MatrixSetDecChanCfg

Configure decoding channel.

API Definition

```
BOOL NET_DVR_MatrixSetDecChanCfg(
  LONG                          lUserID,
  DWORD                         dwDecChan,
  LPNET_DVR_MATRIX_DECCHAN_CONTROL   lpInter
)
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChan

[IN] Decoding channel No.

lpInter

[IN] Parameters of decoding channel scaling control, see details in ***NET_DVR_MATRIX_DECCHAN_CONTROL*** .

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.33 NET_DVR_MatrixSetDecChanEnable

Set the decoding status of the current channel or window.

API Definition

```
BOOL NET_DVR_MatrixSetDecChanEnable(
  LONG    lUserID,
  DWORD   dwDecChanNum,
  DWORD   dwEnable
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or window No.

dwEnable

[IN] 0-Disable, 1-Enable

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

If the parameter **dwEnable** is set to 0, the decoding of the current window will be paused even the window is in dynamic or auto-switch decoding mode.

See Also

NET_DVR_MatrixGetDecChanEnable

3.34 NET_DVR_MatrixSetLoopDecChanEnable

Set the auto-switch status of a decoding channel.

API Definition

```
BOOL NET_DVR_MatrixSetLoopDecChanEnable(
  LONG    lUserID,
  DWORD   dwDecChanNum,
  DWORD   dwEnable
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or window No.

dwEnable

[OUT] 0-Disable, 1-Enable.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixGetLoopDecChanEnable

3.35 NET_DVR_MatrixSetLoopDecChanInfo_V41

Set the auto-switch decoding channel parameters.

API Definition

```
BOOL NET_DVR_MatrixSetLoopDecChanInfo_V41(
  LONG                          lUserID,
  DWORD                         dwDecChanNum,
  NET_DVR_MATRIX_LOOP_DECINFO_V41    *lpInter
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or window No.

lpInter

[OUT] Parameters of auto-switch decoding channel, see details in the structure ***NET_DVR_MATRIX_LOOP_DECINFO_V41***

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

Up to 64 cameras can be linked to each decoding channel of decoder for auto-switch, and the switching interval can be configured.

See Also

NET_DVR_MatrixGetLoopDecChanInfo_V41

3.36 NET_DVR_MatrixSetRemotePlay

Configure remote file playback.

API Definition

```
BOOL NET_DVR_MatrixSetRemotePlay(
  LONG                          lUserID,
  DWORD                         dwDecChanNum,
  LPNET_DVR_MATRIX_DEC_REMOTE_PLAY   lpInter
)
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No.

lpInter

[IN] Parameters of remote file playback, see details in NET_DVR_MATRIX_DEC_REMOTE_PLAY.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

After calling this API to complete configuration, you should call ***NET_DVR_MatrixSetRemotePlayControl*** （NET_DVR_PLAYSTART） to start playback.

3.37 NET_DVR_MatrixSetRemotePlayControl

Remotely control the playback.

API Definition

```
BOOL NET_DVR_MatrixSetRemotePlayControl(
  LONG    lUserID,
  DWORD   dwDecChanNum,
  DWORD   dwControlCode,
  DWORD   dwInValue,
  DWORD   *lpOutValue
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel or window No.

dwControlCode

[IN] Control commands, see details in the following table.

Command (dwControlCode)	Command No.	Description
NET_DVR_PLAYSTART	1	Start playback.
NET_DVR_PLAYSTOP	2	Stop playback.
NET_DVR_PLAYPAUSE	3	Pause playback.
NET_DVR_PLAYRESTART	4	Resume playback.
NET_DVR_PLAYFAST	5	Fast forward.
NET_DVR_PLAYSLOW	6	Slow forward.
NET_DVR_PLAYNORMAL	7	Playback in ×1 speed.
NET_DVR_PLAYSTARTAUDIO	9	Turn on audio.
NET_DVR_PLAYSTOPAUDIO	10	Turn off audio.
NET_DVR_PLAYSETPOS	12	Set the playback progress.

dwInValue

[IN] Configuration parameters

lpOutValue

[OUT] Returned value.

Return Values

Return *TRUE* for ssuccess, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

The validity of **dwInValue** and **lpOutValue** is depended on the control command, e.g., for comamnd NET_DVR_PLAYSETPOS, **dwInValue** should be assigned.

3.38 NET_DVR_MatrixSetTranInfo_V30

Set parameters of transparent channel.

API Definition

```
BOOL NET_DVR_MatrixSetTranInfo_V30(
  LONG                              lUserID,
  LPNET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30  lpTranInfo
)
```

Parameters

lUserID

[IN] User ID, which is returned by ***NET_DVR_Login_V40*** .

lpTranInfo

[IN] Transparent channel parameters, see details in ***NET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30*** .

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

So far up to 64 transparent channels are supported, including RS232 and RS485 channels. At most 1 RS232 and 1 RS485 full-duplex transparent channel can be supported and you can also choose not to set full-duplex transparent channel.

3.39 NET_DVR_MatrixStartDynamic_V41

Start dynamic decoding.

API Definition

```
BOOL NET_DVR_MatrixStartDynamic_V41(
  LONG                    lUserID,
  DWORD                   dwDecChanNum,
  NET_DVR_PU_STREAM_CFG_V41    *lpDynamicInfo
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or sub window No. (1-byte video wall No.+1-byte sub window No.+2-byte window No.)

lpDynamicInfo

[IN] Dynamic decoding parameters, see details in the structure ***NET_DVR_PU_STREAM_CFG_V41*** .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

After starting dynamic decoding by calling this API, the decoding will not stop until the NET_DVR_MatrixStopDynamic is called. If the network disconnected during decoding, the decoder will automatically reconnect to the encoding device until reconnected.

See Also

NET_DVR_MatrixStopDynamic

3.40 NET_DVR_MatrixStartPassiveDecode

Start the passive decoding.

API Definition

```
LONG NET_DVR_MatrixStartPassiveDecode(
  LONG                    lUserID,
  DWORD                   dwDecChanNum,
  NET_DVR_MATRIX_PASSIVEMODE     *lpPassiveMode
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40***

dwDecChanNum

[IN] Decoding channel No.

lpPassiveMode

[IN] Passive decoding parameters, see details in the structure ***NET_DVR_MATRIX_PASSIVEMODE*** .

Return Values

Return *-1* for failure, and return other values as the parameters of API ***NET_DVR_MatrixSendData*** . If *-1* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixStopPassiveDecode

3.41 NET_DVR_MatrixStopDynamic

Stop dynamic decoding.

API Definition

```
BOOL NET_DVR_MatrixStopDynamic(
  LONG   lUserID,
  DWORD  dwDecChanNum
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

[IN] Decoding channel No. or sub window No. (1-byte video wall No.+1-byte sub window No.+2-byte window No.)

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_MatrixStartDynamic_V41

3.42 NET_DVR_MatrixStopPassiveDecode

Stop the passive decoding.

API Definition

```
BOOL NET_DVR_MatrixStopPassiveDecode(
  LONG   lPassiveHandle
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_MatrixStartPassiveDecode*** .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.43 NET_DVR_RemoteControl

Implement remote control.

API Definition

```
BOOL NET_DVR_RemoteControl(
  LONG     lUserID,
  DWORD    dwCommand,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Control commands. To realize different functions, the commands are different.

lpInBuffer

[IN] Input parameters, which vary with different control commands.

dwInBufferSize

[IN] Size of input parameters.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

3.44 NET_DVR_SetConnectTime

Set network connection timeout and connection attempts.

API Definition

```
BOOL NET_DVR_SetConnectTime(
  DWORD   dwWaitTime,
  DWORD   dwTryTimes
);
```

Parameters

dwWaitTime

[IN] Timeout, unit: ms, value range: [300,75000]; the maximum timeout varies with different operating systems.

dwTryTimes

[IN] Connection attempts (reserved).

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- For Windows operating system, the default connection timeout is 3000 ms; for Linux operating system with version 5.2.7.2 and above, he default connection timeout is 3500 ms.
- For HCNetSDK with version 4.0 and above, when the configured timeout is larger than or smaller than the limit value, this API will not return *FALSE*, it will automatically use the timeout that is closest to the limit value as the actual timeout.

3.45 NET_DVR_SetDeviceConfig

Set device parameters in batch (sending data is supported).

API Definition

```
BOOL NET_DVR_SetDeviceConfig(
  LONG     lUserID,
  DWORD    dwCommand,
```

```
  DWORD    dwCount,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize,
  LPVOID   lpStatusList,
  LPVOID   lpInParamBuffer,
  DWORD    dwInParamBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device configuration commands, which are different according to different configurations.

dwCount

[IN] Number of cameras to be set at a time. 0,1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 256 cameras can be configured at a time.

lpInBuffer

[IN] Pointer of configuration condition buffer, e.g., stream ID, which specifies the number (**dwCount**) of cameras to set, and relates to the configuration commands.

dwInBufferSize

[IN] Size of configuration condition buffer, which saves the configured information of cameras with the number of **dwCount**.

lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras that need to be searched, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0, it refers to setting succeeded, otherwise, this parameter value is the error code.

lpInParamBuffer

[IN] Device parameters to set, which relates to the configuration commands. And there is a one-to-one correspondence between the parameters and the cameras that need to be searched.

dwInParamBufferSize

[IN] Set the size of content buffer.

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not indicate that all settings are succeeded, you can get the value of **lpStatusList[n]** to check which one is succeeded.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_GetDeviceConfig

3.46 NET_DVR_SetDeviceConfigEx

Set the device parameters in batch (sending data is supported).

API Definition

```
BOOL NET_DVR_SetDeviceConfigEx(
  LONG              lUserID,
  DWORD             dwCommand,
  DWORD             dwCount,
  NET_DVR_IN_PARAM     *lpInParam,
  NET_DVR_OUT_PARAM    *lpOutParam
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

dwCount

[IN] Number of cameras to be set at a time. 0,1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 64 cameras can be configured at a time.

lpInParam

[IN] Input parameter buffer, see details in the structure ***NET_DVR_IN_PARAM*** (the value of each member in the structure varies with the **dwCommand**).

lpOutParam

[IN] Output parameter buffer, see details in the structure ***NET_DVR_OUT_PARAM*** (the value of each member in the structure varies with the **dwCommand**).

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not mean that all settings are succeeded, you can check the value of **lpStatusList[n]** to check which one is succeeded.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

Extended From: ***NET_DVR_SetDeviceConfig***

3.47 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetDVRConfig(
  LONG     lUserID,
  DWORD    dwCommand,
  LONG     lChannel,
  LPVOID   lpInBuffer,
  DWORD    dwInBufferSize
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xFFFFFFFF-invalid, 1-main NIC, 2-extended NIC.

lpInBuffer

[IN] Pointer of input data buffer. For different configuration functions, the structures of this parameter are different.

dwInBufferSize

[IN] Size of input data buffer (unit: byte).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the ***Device Network SDK Errors*** .

See Also

NET_DVR_GetDVRConfig

3.48 NET_DVR_SetSDKInitCfg

Set initialization parameters.

API Parameters

```
BOOL NET_DVR_SetSDKInitCfg(
  NET_SDK_INIT_CFG_TYPE    enumType,
  void* const              lpInBuff
);
```

Parameters

enumType

[IN] Initialization parameter type. Different type values correspond to different parameters, see details in the table below.

Table 3-1 NET_SDK_INIT_CFG_TYPE

enumType	Value	Description	lpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	***NET_DVR_INIT_CFG_ABILITY***
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	***NET_DVR_LOCAL_SDK_PATH***
NET_SDK_INIT_CFG_LIBEAY_PATH	3	Set path (including library name) for libeay32.dll (Windows), libcrypto.so (Linux), and libcrypto.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., ***C:\\libeay32.dll***.
NET_SDK_INIT_CFG_SSLEAY_PATH	4	Set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux), libssl.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., ***C:\\ssleay32.dll***.

lpInBuff

[IN] Input parameter. Different parameter types correspond to different structures, see details in the table above.

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

This API should be called before calling ***NET_DVR_Init*** to initialize and check the dependent libraries or capabilities. This API only takes effect for POSIX. For Windows, it takes no effect but success will be returned.

3.49 NET_DVR_SetSDKLocalCfg

Set the local parameters.

API Definition

```
BOOL NET_DVR_SetSDKLocalCfg(
  NET_SDK_LOCAL_CFG_TYPE    enumType,
  void* const               lpInBuff
);
```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different SDK parameters, see details in ***NET_SDK_LOCAL_CFG_TYPE*** .

lpInBuff

[IN] Input parameters. For different configuration options, the structures of input parameters are different, see details in ***NET_SDK_LOCAL_CFG_TYPE*** .

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.
Before setting parameters for this function, make sure no device has logged in.

See Also

NET_DVR_GetSDKLocalCfg

3.50 NET_DVR_SetSTDConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetSTDConfig(
  LONG                lUserID,
  DWORD               dwCommand,
  NET_DVR_STD_CONFIG  lpConfigParam
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **lpCondBuffer** and **lpInBuffer** in the **lpConfigParam** are also different. See the structure ***NET_DVR_STD_CONFIG*** for details.

Note

When getting configuration parameters, the **lpOutBuffer** in the **lpConfigParam** is invalid, you can set it to "NULL".

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

See Also

NET_DVR_GetSTDConfig

3.51 NET_DVR_STDXMLConfig

Transmit request URL with XML or JSON format to implement some typical functions.

API Definition

```
BOOL NET_DVR_STDXMLConfig(
  LONG                              lUserID,
  const NET_DVR_XML_CONFIG_INPUT    *lpInputParam,
  NET_DVR_XML_CONFIG_OUTPUT         *lpOutputParam
);
```

Parameters

lUserID

[IN] Value returned by ***NET_DVR_Login_V40*** .

lpInputParam

[IN] Input parameters, refer to the structure ***NET_DVR_XML_CONFIG_INPUT*** for details.

lpOutputParam

[IN][OUT] Output parameters, refer to the structure ***NET_DVR_XML_CONFIG_OUTPUT*** for details.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.
If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

The input parameter **lpInputParam** and output parameter **lpOutputParam** are different when transmitting text protocol for implementing different functions, and each parameter corresponds to a component of text protocol, see the relations below:

Parameter of NET_DVR_STDXMLConfig		Component of Text Protocol
lpInputParam	**lpRequestUrl** (see in structure ***NET_DVR_XML_CONFIG_INPUT***)	Method+URL E.g., GET /ISAPI/System/capabilities
	lpInBuffer (see in structure ***NET_DVR_XML_CONFIG_INPUT***)	Request Message
lpOutputParam	**lpOutBuffer** (see in structure ***NET_DVR_XML_CONFIG_OUTPUT***)	Response Message
	lpStatusBuffer (see in structure ***NET_DVR_XML_CONFIG_OUTPUT***)	Response Message

3.52 NET_DVR_UploadLogo

Upload LOGO.

API Definition

```
BOOL NET_DVR_UploadLogo(
  LONG                 lUserID,
```

```
 DWORD                dwDecChanNum,
 LPNET_DVR_DISP_LOGOCFG  lpDispLogoCfg,
 char                 *sLogoBuffer
)
```

Parameters

lUserID

User ID, which is returned by ***NET_DVR_Login_V40*** .

dwDecChanNum

Decoding channel No.

lpDispLogoCfg

Parameters of LOGO, see details in structure ***NET_DVR_DISP_LOGOCFG*** .

sLogoBuffer

LOGO data buffer, the maximum size is 100k, and the width and the height should be multiples of 32.

Return Value

Return *TRUE* for success and *FALSE* for failure. If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Appendix A. Appendixes

A.1 Callback Function

A.1.1 CHAR_ENCODE_CONVERT

Encoding type conversion callback function.

Callback Function Definition

```
typedef int(CALLBACK *CHAR_ENCODE_CONVERT)(
  char      *pInput,
  DWORD     dwInputLen,
  DWORD     dwInEncodeType,
  char      *pOutput,
  DWORD     dwOutputLen,
  DWORD     dwOutEncodeType
);
```

Parameters

pInput

[IN] Input string, whose memory and size is applied and provided by the third-party platform

dwInputLen

[IN] Input buffer size.

dwInEncodeType

[IN] Encoding types of input string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

pOutput

[OUT] Output string, whose memory is applied by the third-party platform.

dwOutputLen

[OUT] Output buffer size.

dwOutEncodeType

[OUT] Encoding tupes of output string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

Return Values

Return *-1* for failure, and return *0* for success.

A.1.2 fLoginResultCallBack

Login Status Callback Function

Member	Data Type	Description
lUserID	LONG	User ID, which is returned by ***NET_DVR_Login_V40*** .
dwResult	DWORD	Login status: 0-asynchronously logging in failed, 1-asynchronously logged in.
lpDeviceInfo	***NET_DVR_DEVICEINFO_V40***	Device information, such as serial No., channel, capability, and so on.
pUser	void*	User data.

A.2 Data Structure

A.2.1 NET_DVR_ADDRESS

IP address and port parameter structure.

```
struct{
  NET_DVR_IPADDR    struIP;
  WORD              wPort;
  BYTE              byRes[2];
}NET_DVR_ADDRESS,*LPNET_DVR_ADDRESS;
```

Members

struIP

IP address, see details in the structure .

wPort

Port number.

byRes

Reserved.

A.2.2 NET_DVR_ATMFINDINFO

Condition structure for searching files with ATM information.

Structure Definition

```
struct{
  BYTE    byTransactionType;
  BYTE    byRes[3];
  DWORD   dwTransationAmount;
}NET_DVR_ATMFINDINFO, *LPNET_DVR_ATMFINDINFO;
```

Members

byTransactionType

Transaction type: 0-all, 1-search, 2-withdraw, 3-deposit, 4-change password, 5-transfer, 6-search without card, 7-deposit without card, 8-retract cash, 9-retract card, 10-custom.

byRes

Reserved, set to 0.

dwTransationAmount

Transaction amount, -1: search without displaying amount.

See Also

NET_DVR_SPECIAL_FINDINFO_UNION

A.2.3 NET_DVR_BUF_INFO

Structure about buffer information.

Structure Definition

```
struct{
  void*   pBuf;
  DWORD   nLen;
}NET_DVR_BUF_INFO,*LPNET_DVR_BUF_INFO;
```

Members

pBuf

Buffer pointer

nLen

Buffer size

See Also

NET_DVR_IN_PARAM
NET_DVR_OUT_PARAM

A.2.4 NET_DVR_CAM_MODE

Structure about the enumeration of signal source types.

Structure Definition

```
enum _NET_DVR_CAM_MODE_{
  NET_DVR_UNKNOW      = 0,
  NET_DVR_CAM_BNC     = 1,
  NET_DVR_CAM_VGA     = 2,
  NET_DVR_CAM_DVI     = 3,
  NET_DVR_CAM_HDMI    = 4,
  NET_DVR_CAM_IP      = 5,
  NET_DVR_CAM_RGB     = 6,
  NET_DVR_CAM_DECODER  = 7,
  NET_DVR_CAM_MATRIX   = 8,
  NET_DVR_CAM_YPBPR    = 9,
  NET_DVR_CAM_USB     = 10,
  NET_DVR_CAM_SDI     = 11,
  NET_DVR_CAM_HDI     = 12,
  NET_DVR_CAM_DP      = 13,
  NET_DVR_CAM_HDTVI    = 14,
  NET_DVR_CAM_JOINT    = 15,
  NET_DVR_CAM_HDBASET  = 16
}NET_DVR_CAM_MODE
```

Members

NET_DVR_UNKNOW

Invalid

NET_DVR_CAM_BNC

BNC input

NET_DVR_CAM_VGA

VGA input

NET_DVR_CAM_DVI

DVI input

NET_DVR_CAM_HDMI

HDMI input

NET_DVR_CAM_IP

IP input

NET_DVR_CAM_RGB

RGB input

NET_DVR_CAM_DECODER

Decoding board input

NET_DVR_CAM_MATRIX

Matrix signal source

NET_DVR_CAM_YPBPR

YPbPr input

NET_DVR_CAM_USB

USB input

NET_DVR_CAM_SDI

SDI input

NET_DVR_CAM_HDI

HDI input

NET_DVR_CAM_DP

DP input

NET_DVR_CAM_HDTVI

HDTVI input

NET_DVR_CAM_JOINT

Jointed signal source

NET_DVR_CAM_HDBASET

HDBASET input

A.2.5 NET_DVR_CETTIFICATE_INFO

Certificate information structure

Structure Definition

```
struct{
  DWORD        dwSize;
  char         szIssuer[MAX_CERTIFICATE_ISSUER_LEN/*64*/];
  char         szSubject[MAX_CERTIFICATE_SUBJECT_LEN/*64*/];
  NET_DVR_TIME   struStartTime;
  NET_DVR_TIME   struEndTime;
  BYTE         byRes1[1024];
}NET_DVR_CETTIFICATE_INFO, *LPNET_DVR_CETTIFICATE_INFO;
```

Members

dwSize

Structure size.

szIssuer

Certificate issuer.

szSubject

Certificate holder.

struStartTime

Start time of expiry date, refer to the structure ***NET_DVR_TIME*** for details.

struEndTime

End time of expiry date, refer to the structure ***NET_DVR_TIME*** for details.

byRes1

Reserved.

A.2.6 NET_DVR_DDNS_ADDRESS

Structure about DDNS information.

Structure Definition

```
struct{
  BYTE    byDevAddress[MAX_DOMAIN_NAME/*64*/];
  BYTE    byDevDdns[MAX_DOMAIN_NAME/*64*/];
  BYTE    byDdnsType;
  BYTE    byRes1[3];
  WORD    wDevPort;
  WORD    wDdnsPort;
  BYTE    byres[64];
 }NET_DVR_DDNS_ADDRESS, *LPNET_DVR_DDNS_ADDRESS;
```

Members

byDevAddress

Device domain name

byDevDdns

DDNS IP address

byDdnsType

DDNS type: 0-IPServer, 1-Dyndns, 2- PeanutHull, 3- NO-IP, 4- hiDDNS

byRes1

Reserved

wDevPort

Device port number

wDdnsPort

DDNS port number

byRes

Reserved

See Also

NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX

A.2.7 NET_DVR_DEC_DDNS_DEV

Configuration parameter structure about getting stream by DDNS.

Structure Definition

```
struct{
  NET_DVR_DEV_DDNS_INFO        struDdnsInfo;
  NET_DVR_STREAM_MEDIA_SERVER   struMediaServer;
}NET_DVR_DEC_DDNS_DEV,*LPNET_DVR_DEC_DDNS_DEV;
```

Members

struDdnsInfo

DDNS configuration parameters, see the structure ***NET_DVR_DEV_DDNS_INFO*** for details.

struMediaServer

Stream media server configuration parameters, see the structure ***NET_DVR_STREAM_MEDIA_SERVER*** for details.

See Also

NET_DVR_DEC_STREAM_MODE

A.2.8 NET_DVR_DEC_STREAM_DEV_EX

Configuration parameter structure about getting stream from device.

Structure Definition

```
struct{
  NET_DVR_STREAM_MEDIA_SERVER   struStreamMediaSvrCfg;
  NET_DVR_DEV_CHAN_INFO_EX      struDevChanInfo;
}NET_DVR_DEC_STREAM_DEV_EX,*LPNET_DVR_DEC_STREAM_DEV_EX;
```

Members

struStreamMediaSvrCfg

Stream media server configuration parameters, see the structure ***NET_DVR_STREAM_MEDIA_SERVER*** for details.

struDevChanInfo

Device channel configuration parameters, see the structure ***NET_DVR_DEV_CHAN_INFO_EX*** for details.

See Also

NET_DVR_DEC_STREAM_MODE

A.2.9 NET_DVR_DEC_STREAM_MODE

Configuration parameter union about streaming mode.

Structure Definition

```
union{
  NET_DVR_DEC_STREAM_DEV_EX    struDecStreamDev;
  NET_DVR_PU_STREAM_URL        struUrlInfo;
  NET_DVR_DEC_DDNS_DEV         struDdnsDecInfo;
  BYTE                         byRes[300];
}NET_DVR_DEC_STREAM_MODE,*LPNET_DVR_DEC_STREAM_MODE;
```

Members

struDecStreamDev

Get stream from device or stream media server by IP address or domain name, see the structure ***NET_DVR_DEC_STREAM_DEV_EX*** for the configuration details.

struUrlInfo

Get stream from device or stream media server by URL, see the structure ***NET_DVR_PU_STREAM_URL*** for the configuration details.

struDdnsDecInfo

Get stream from device by DDNS, see the structure ***NET_DVR_DEC_DDNS_DEV*** for the configuration details.

byRes

Reserved, set to 0.

A.2.10 NET_DVR_DEV_CHAN_INFO_EX

Structure about front-end device information.

Structure Definition

```
struct{
  BYTE    byChanType;
  BYTE    byStreamId[STREAM_ID_LEN/*32*/];
```

```
  BYTE    byRes1[3];
  DWORD   dwChannel;
  BYTE    byRes2[24];
  BYTE    byAddress[MAX_DOMAIN_NAME/*64*/];
  WORD    wDVRPort;
  BYTE    byChannel;
  BYTE    byTransProtocol;
  BYTE    byTransMode;
  BYTE    byFactoryType;
  BYTE    byDeviceType;
  BYTE    byDispChan;
  BYTE    bySubDispChan;
  BYTE    byResolution;
  BYTE    byRes[2];
  BYTE    sUserName[NAME_LEN/*32*/];
  BYTE    sPassword[PASSWD_LEN/*16*/];
}NET_DVR_DEV_CHAN_INFO_EX,*LPNET_DVR_DEV_CHAN_INFO_EX;
```

Members

byChanType

Channel type: 0-normal channel, 1-channel-zero, 2-stream ID, 3-local input source

byStreamId

Stream ID, it is valid only when the **byChanType** is 2.

byRes1

Reserved

dwChannel

Channel No., it is valid only when the byChanType is 0, 1, 3. If the channel type is local input source, this member is the local source index No.

byRes2

Reserved

byAddress

Device IP address or domain name

wDVRPort

Device port number

byChannel

Invalid

byTransProtocol

Transfer protocol type: 0-TCP, 1-UDP

byTransMode

Stream mode: 0-main stream, 1-sub-stream

byFactoryType

Front-end device manufacturer

byDeviceType

Device type (for MVC only): 1-network camera, 2-encoding device

byDispChan

Display channel No.

bySubDispChan

Sub display channel No.

byResolution

Resolution: 1- CIF, 2- 4CIF, 3- 720P, 4- 1080P, 5- 500W

byRes

Reserved, set to 0.

sUserName

User name for log in to device.

sPassword

Device password.

See Also

NET_DVR_DEC_STREAM_DEV_EX

A.2.11 NET_DVR_DEV_DDNS_INFO

Configuration parameter structure about DDNS.

Structure Definition

```
struct{
  BYTE    byDevAddress[MAX_DOMAIN_NAME/*64*/];
  BYTE    byTransProtocol;
  BYTE    byTransMode;
  BYTE    byDdnsType;
  BYTE    byRes1;
  BYTE    byDdnsAddress[MAX_DOMAIN_NAME/*64*/];
  WORD    wDdnsPort;
  BYTE    byChanType;
  BYTE    byFactoryType;
  DWORD   dwChannel;
  BYTE    byStreamId[STREAM_ID_LEN/*32*/];
  BYTE    sUserName[NAME_LEN/*32*/];
  BYTE    sPassword[PASSWD_LEN/*16*/];
  WORD    wDevPort;
  BYTE    byRes2[2];
}NET_DVR_DEV_DDNS_INFO,*LPNET_DVR_DEV_DDNS_INFO;
```

Members

byDevAddress

Device domain name

byTransProtocol

Transfer protocol type: 0-TCP, 1-UDP, 2-multicast

byTransMode

Stream mode: 0-main stream, 1-sub-stream

byDdnsType

DDNS type: 0-IPServer, 1-Dyndns, 2- PeanutHull, 3- NO-IP, 4-hiDDNS

byRes1

Reserved

byDdnsAddress

DDNS address

wDdnsPort

DDNS port number

byChanType

Channel type: 0-normal channel, 1-channel-zero, 2-stream ID

byFactoryType

Front-end device manufacturer

dwChannel

Device channel No.

byStreamId

Stream ID, it is valid only when the **byChanType** is 2.

sUserName

User name for log in to device

sPassword

Device password.

wDevPort

Device port number

byRes2

Reserved

See Also

NET_DVR_DEC_DDNS_DEV

A.2.12 NET_DVR_DEVICEINFO_V30

Device parameter structure (V30).

Device Parameter Structure (V30)

Member	Data Type	Description
sSerialNumber	BYTE	Device serial No.
byAlarmInPortNum	BYTE	Number of analog alarm inputs
byAlarmOutPortNum	BYTE	Number of analog alarm outputs
byDiskNum	BYTE	Number of HDDs
byDVRType	BYTE	Device type
byChanNum	BYTE	Number of analog channels
byStartChan	BYTE	Start No. of analog channel, which starts from 1.
byAudioChanNum	BYTE	Number of two-way audio channels
byIPChanNum	BYTE	Number of digital channels, low 8-bit.
byZeroChanNum	BYTE	Number of channel-zero
byMainProto	BYTE	Transmission protocol type of main stream: 0-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision Private Protocol+RTSP
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision Private Protocol+RTSP
bySupport	BYTE	Capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported. • bySupport&0x1: whether supports VCA search. • bySupport&0x2: whether supports backup. • bySupport&0x4: whether supports getting encoding parameters. • bySupport&0x8: whether supports dual-NIC. • bySupport&0x10: whether supports remote SADP.

Member	Data Type	Description
		• bySupport&0x20: whether supports RAID card. • bySupport&0x40: whether supports searching in IPSAN directory. • bySupport&0x80: whether supports RTP over RTSP.
bySupport1	BYTE	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported. • bySupport1&0x1: whether supports SNMP with version 30. • bySupport1&0x2: whether supports playback and downloading video files. • bySupport1&0x4: whether supports setting the arming priority. • bySupport1&0x8: whether supports extending the arming time period. • bySupport1&0x10: whether supports multiple HDDs (more than 33). • bySupport1&0x20: whether supports RTP over RTSP. • bySupport1&0x80: whether supports license plate recognition alarm.
bySupport2	BYTE	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported. • bySupport2&0x1: whether supports getting stream via URL. • bySupport2&0x2: whether supports FTP with version 40. • bySupport2&0x4: whether supports ANR. • bySupport2&0x20: whether supports getting device status. • bySupport2&0x40: whether supports encrypting stream.
wDevType	WORD	Device model

Member	Data Type	Description
bySupport3	BYTE	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported. • bySupport3&0x1: whether supports multi-stream. • bySupport3&0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on). • bySupport3&0x20: whether supports getting stream via DDNS.
byMultiStreamProto	BYTE	Whether supports multi-stream, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support. • byMultiStreamProto&0x1: whether supports third-stream. • byMultiStreamProto&0x2: whether supports fourth-stream. • byMultiStreamProto&0x40: whether supports main stream. • byMultiStreamProto&0x80: whether supports sub-stream.
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).
byStartDTalkChan	BYTE	Start No. of two-way audio channel, 0-no two-way audio channel.
byHighDChanNum	BYTE	Number of digital channels, high 8-bit.
bySupport4	BYTE	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.

Member	Data Type	Description
		• bySupport4&0x01: whether all stream types support RTSP and Hikvision Private Protocol. • bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). • bySupport4&0x10: whether supports loading network disk by domain name.
byLanguageType	BYTE	Supported language types, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support. • byLanguageType ==0: this field is not supported by device. • byLanguageType&0x1: whether supports Chinese. • byLanguageType&0x2: whether supports English.
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- The maximum number of digital channels equal to byIPChanNum+byHighDChanNum*256.
- For login via text protocol, the following parameters are not supported: **byMainProto**, **bySubProto**, **bySupport**, **bySupport1**, **bySupport2**, **bySupport3**, **bySupport4**, **bySupport5**, **bySupport6**, **bySupport7**, **byMultiStreamProto**, **byStartDTalkChan**, **byVoiceInChanNum**, **byStartVoiceInChanNo**, **byMirrorChanNum**, and **wStartMirrorChanNo**.

See Also

NET_DVR_DEVICEINFO_V40

A.2.13 NET_DVR_DEVICEINFO_V40

Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	***NET_DVR_DEVICEINFO_V30***	Device parameters
bySupportLock	BYTE	Whether supports locking function: 1-support.
byRetryLoginTime	BYTE	Remaining login attempts, it is valid when the user name or password is incorrect and the **bySupportLock** is 1.
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when **bySupportLock** is 1. During the locing time, if the user try to log in to again, the remaining locking time will resume to 30 minutes.
byCharEncodeType	BYTE	Character encodings. 0-no decoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, …, 21-ISO8859-15 (Western European)
bySupportDev5	BYTE	Whether to support getting the parameters of devices that support HCNetSDK version 5.0 or above, the size of device name and type name are extended to 64 bytes.
bySupport	BYTE	Whether it supports uploading changes, it depends on the result of bitwise AND (&) operation: 0-not support, 1-support. The result of **bySupport**&0x1 indicates that this member is reserved; the result of **bySupport**&0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.

Member	Data Type	Description
byLoginMode	BYTE	Login mode: 0-login via private protocol, 1-login via text protocol. For private protocol, the default login port number is 8000, and for text protocol, the default login port number is 80 or 443.
dwOEMCode	DWORD	OEM code.
iResidualValidity	int	Remaining valid days of the user's password, unit: day. If the negative number is returned, it indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member **iResidualValidity** is valid: 0-invalid, 1-valid.
bySingleStartDTalkChan	BYTE	Start channel No. for connecting independent audio tracks to the device. The value 0 is reserved and invalid. The channel No. of audio tracks cannot start from 0.
bySingleDTalkChanNums	BYTE	Total number of channels of the device connected with independent tracks, 0-not support.
byPassWordResetLevel	BYTE	Whether to prompt the non-admin user to change the password: 0 (invalid), 1 (If the administrator creates a non-admin user account with an initial password, the non-admin user will be prompted "Please change the initial password" each time he/she logs in to the device until he/she changes the initial password), 2(If the non-admin user's password has been changed by the administrator, the non-admin user will be prompted "Please set a new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of **bySupportStreamEncrypt**&0x1 indicates whether to support RTP/TLS streaming, the

Member	Data Type	Description
		result of **bySupportStreamEncrypt**&0x2 indicates whether to support SRTP/UDP streaming, and the result of **bySupportStreamEncrypt**&0x4 indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:
  - Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
  - Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).
  - Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
  - Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

A.2.14 NET_DVR_DISPLAYCFG

Structure about video output configuration parameters.

Structure Definition

```
struct{
  DWORD               dwSize;

                          NET_DVR_DISPLAYPARAM
                           struDisplayParam[MAX_DISPLAY_NUM/*512*/];
  BYTE              byRes[128];
}NET_DVR_DISPLAYCFG,*LPNET_DVR_DISPLAYCFG;
```

Members

dwSize

Structure size

struDisplayParam

Video output parameters, see details in the structure ***NET_DVR_DISPLAYPARAM*** .

byRes

Reserved, set to 0.

A.2.15 NET_DVR_DISPLAYPARAM

Structure about video output parameters.

Structure Definition

```
struct{
  DWORD    dwDisplayNo;
  BYTE     byDispChanType;
  BYTE     byRes[11];
}NET_DVR_DISPLAYPARAM,*LPNET_DVR_DISPLAYPARAM;
```

Members

dwDisplayNo

Video output No.

byDispChanType

Connection mode: 1-BNC, 2-VGA, 3-HDMI®, 4-DVI, 5-SDI, 6-FIBER, 7-RGB, 8-YPrPb, 9-VGA/HDMI®/DVI, 10-3GSDI, 11-VGA/DVI, 0xff-invalid.

byRes

Reserved, set to 0.

See Also

NET_DVR_DISPLAYCFG

A.2.16 NET_DVR_DISP_LOGOCFG

Structure of uploaded LOGO.

Structure Definition

```
struct{
  DWORD   dwCorordinateX;
  DWORD   dwCorordinateY;
```

```
  BYTE    byRes1[8];
  BYTE    byFlash;
  BYTE    byTranslucent;
  BYTE    byRes2[6];
  DWORD   dwLogoSize;
}NET_DVR_DISP_LOGOCFG,*LPNET_DVR_DISP_LOGOCFG;
```

Members

dwCorordinateX

X-coordinate for LOGO display

dwCorordinateY

Y-coordinate for LOGO display

byRes1

Reserved, please set it to 0

byFlash

Whether it is flashing: 0-no, 1-yes

byTranslucent

Whether it is translucent:0- no, 1- yes

byRes2

Reserved, please set it to 0

dwLogoSize

Size of LOGO, including header of BMP

A.2.17 NET_DVR_FILECOND_V50

File search condition structure.

Structure Definition

```
struct{
  NET_DVR_STREAM_INFO          struStreamID;
  NET_DVR_TIME_SEARCH_COND     struStartTime;
  NET_DVR_TIME_SEARCH_COND     struStopTime;
  BYTE                  byFindType;
  BYTE                  byDrawFrame;
  BYTE                  byQuickSearch;
  BYTE                  byStreamType;
  DWORD                    dwFileType;
  DWORD                    dwVolumeNum;
  BYTE                  byIsLocked;
  BYTE                  byNeedCard;
  BYTE                  byOnlyAudioFile;
  BYTE                  bySpecialFindInfoType;
```

```
  char                    szCardNum[32];
  char                    szWorkingDeviceGUID[16];
  NET_DVR_SPECIAL_FINDINFO_UNION    uSpecialFindInfo;
  DWORD                   dwTimeout;
  BYTE                    byRes[252];
}NET_DVR_FILECOND_V50, *LPNET_DVR_FILECOND_V50;
```

Memebers

struStreamID

Stream ID or channel No., see details in the structure of ***NET_DVR_STREAM_INFO*** .

struStartTime

Start time, see details in the structure of ***NET_DVR_TIME_SEARCH_COND*** .

struStopTime

End time, see details in the structure of ***NET_DVR_TIME_SEARCH_COND*** .

byFindType

File storage type for search: 0-search in normal volume, 1-search in storage volume, 2-search in N+1 hot spare.

byDrawFrame

Whether to extract the frame: 0-no, 1-yes.

byQuickSearch

Whether to enable searching by calendar: 0-no, 1-yes.

byStreamType

Stream types: 0-main stream, 1-sub-stream, 2-third stream, 0xff-all.

dwFileType

File types.

dwVolumeNum

Storage volume No., it is valid only when **byFindType** is 1.

byIsLocked

Whether to lock the file: 0-no, 1-yes, 0xff-lock or unlock all files.

byNeedCard

Whether the card search is required: 0-no, 1-yes.

byOnlyAudioFile

Audio or video file: 0-video file, 1-audio file.

bySpecialFindInfoType

Search condition type: 0-invalid, 1-search files with ATM information

szCardNum

Card No.,it is valid only when **byNeedCard** is 1.

szWorkingDeviceGUID

Working station GUID, which is obtained from N+1 hot spare, it is valid only when **byFindType** is 2.

uSpecialFindInfo

Specific search condition union, see details in ***NET_DVR_SPECIAL_FINDINFO_UNION*** .

dwTimeout

Timeout time of searching for files, value range: [5000,15000], 0-no changes of timeout time. Unit: millisecond.

byRes

Reserved.

Remarks

For login based on ISAPI protocol, only the continuously recorded video can be searched.

Related API

NET_DVR_FindFile_V50

A.2.18 NET_DVR_FINDDATA_V50

Structure about file search information.

Structure Definition

```
struct{
  char              sFileName[100];
  NET_DVR_TIME_SEARCH    struStartTime;
  NET_DVR_TIME_SEARCH    struStopTime;
  NET_DVR_ADDRESS       struAddr;
  DWORD               dwFileSize;
  BYTE              byLocked;
  BYTE              byFileType;
  BYTE              byQuickSearch;
  BYTE              byStreamType;
  DWORD               dwFileIndex;
  char              sCardNum[32];
  BYTE              byRes[256];
}NET_DVR_FINDDATA_V50,*LPNET_DVR_FINDDATA_V50;
```

Members

sFileName

File name, it is invalid when searching by time.

struStartTime

File start time, see details in the structure .

struStopTime

File stop time, see details in the structure .

struAddr

Video segment address, see details in the structure , which is used for cluster playback.

dwFileSize

File size.

byLocked

Whether the file is locked: 0-no, 1-yes.

byFileType

Video file type, it is valid when searching by time. 0-continuous recording, 1- motion detection, 2- alarm triggered, 3- motion detection | alarm, 4-motion detection & alarm, 5-command triggered, 6- manual recording, 7-VCA recording, 10-PIR alarm, 11-wireless alarm, 12-panic alarm, 13-all, 14-intelligent traffic events, 15-line crossing, 16-intrusion, 17-sound exception, 18-scene change, 19-line crossing|intrusion|face detection|sound exception|scene change, 20-face detection, 21-sensor, 22-callback, 23-copy back recording, 24-video tampering, 25-POS recording, 26-region entrance, 27-region exiting, 28-loitering detection, 29-people gathering, 30-fast moving, 31-parking detection, 32-unattended baggage, 33-object removal, 34-fire source detection, 35-tampering detection, 36-ship detection, 37-temperature pre-alarm, 38-temperature alarm, 39-fight detection, 40-getting up detection, 41-sleepy detection, 42-temperature difference alarm, 43-offline temperature measurement alarm, 44-zone alarm, 45-panic alarm, 46-inquiry service, 47-getting up detection, 48-climbing detection, 49-in-toilet overtime, 50-running detection, 51-playground overstay detection, 75-dredger detection alarm. It is valid when searching by time.

Note

When logging in to device by ISAPI method, only continuous recording is supported.

byQuickSearch

Search result type, 0-result of normal search, 2-result of searching by time.

byStreamType

Stream type: 0-main stream, 1-sub-stream, 2-third stream.

dwFileIndex

File No.

sCardNum

Card No.

byRes

Reserved.

Remarks

When logging in to device by ISAPI method, the following parameters **byLocked**, **byQuickSearch**, **byStreamType**, **dwFileIndex**, and **sCard** are not supported.

Related API

NET_DVR_FindNextFile_V50

A.2.19 NET_DVR_IN_PARAM

Structure about input parameters.

Structure Definition

```
struct{
                              NET_DVR_BUF_INFO
                               struCondBuf;

                              NET_DVR_BUF_INFO
                               struInParamBuf;
  DWORD            dwRecvTimeout;
  BYTE             byRes[32];
}NET_DVR_IN_PARAM,*LPNET_DVR_IN_PARAM;
```

Members

struCondBuf

Buffer for storing conditions

struInParamBuf

Buffer for storing input parameters

dwRecvTimeout

Data receiving timeout, unit: ms

byRes

Reserved, set to 0

A.2.20 NET_DVR_INIT_CFG_ABILITY

Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	Maximum number of users can log in, see details below:

Member	Data Type	Description
		enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048, INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }INIT_CFG_MAX_NUM
enumMaxAlarmNum	INIT_CFG_MAX_NUM	Maximum number of alarm channels, see details below: enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048, INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }INIT_CFG_MAX_NUM
byRes	Array of BYTE	Reserved, set to 0.

Remarks

By default, up to 2048 channels are supported. More channels require higher computer performance and network bandwidth.

See Also

NET_DVR_SetSDKInitCfg

A.2.21 NET_DVR_INPUT_SIGNAL_LIST

Structure about signal source list.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwInputSignalNums;
  BYTE     *pBuffer;
  BYTE     byRes1[3];
  DWORD    dwBufLen;
  BYTE     byRes2[64];
}NET_DVR_INPUT_SIGNAL_LIST,*LPNET_DVR_INPUT_SIGNAL_LIST;
```

Members

dwSize

Structure size

dwInputSignalNums

Number of signal sources

pBuffer

Signal source information buffer, it is a pointer pointing to the buffer that stores the structure ***NET_DVR_INPUTSTREAMCFG_V40*** (whose number (**dwInputSignalNums**) determines the buffer size).

byRes1

Reserved, set to 0.

dwBufLen

Allocated buffer size

byRes

Reserved, set to 0.

Remarks

- If the member **pBuffer** is set to "NULL", and **dwBufLen** is set to 0, when you call the API ***NET_DVR_GetInputSignalList_V40*** , you can get the number of signal sources (**dwInputSignalNums**).
- The size of signal source information buffer (**pBuffer**) is allocated acoording to the returned number of signal sources (**dwInputSignalNums**), so when you call the API ***NET_DVR_GetInputSignalList_V40*** again, you can get the list of signal source information (**pBuffer**).

A.2.22 NET_DVR_INPUTSTREAMCFG_V40

Structure about input stream parameters.

Structure Definition

```
struct{
  DWORD                 dwSize;
  BYTE                  byValid;
  BYTE                  byCamMode;
  WORD                  wInputNo;
  BYTE                  sCamName[NAME_LEN/*32*/];
  NET_DVR_VIDEOEFFECT       struVideoEffect;
  NET_DVR_PU_STREAM_CFG_V41   struPuStream;
  WORD                  wBoardNum;
  WORD                  wInputIdxOnBoard;
  WORD                  dwResolution;
  BYTE                  byVideoFormat;
  BYTE                  byStatus;
  BYTE                  sGroupName[NAME_LEN];
```

```
  BYTE              byJointMatrix;
  BYTE              byJointNo;
  BYTE              byColorMode;
  BYTE              byScreenServer;
  BYTE              byRes1[2];
  DWORD               dwInputSignalNo;
  BYTE              byRes[120];
}NET_DVR_INPUTSTREAMCFG,*LPNET_DVR_INPUTSTREAMCFG;
```

Members

dwSize

Structure size

byValid

Valid: 0-no, 1-yes

byCamMode

Signal source types, see the structure ***NET_DVR_CAM_MODE*** for details.

wInputNo

Signal source No., this field is extended, so **dwInputSignalNo** is suggested.

sCamName

Signal source name

struVideoEffect

Video parameters, it is valid only when **byColorMode** is 0, see details in the structure ***NET_DVR_VIDEOEFFECT*** .

struPuStream

Input stream parameters, it is valid when entering IP address, see the structure ***NET_DVR_PU_STREAM_CFG_V41*** for details.

wBoardNum

Decoding board No., which with signal source, it can only be obtained.

wInputIdxOnBoard

Signal source position on the decoding board, it can only be obtained.

dwResolution

Resolution

byVideoFormat

Video standards, see the structure ***VIDEO_STANDARD*** for details.

byStatus

Signal source status: 0-invalid, 1-with signal, 2-no signal, 3-exception

sGroupName

Network signal source group name

byJointMatrix

Link to matrix: 0-not link, 1-link. It is invalid when the **byCamMode** is NET_DVR_CAM_BNC, NET_DVR_CAM_VGA, NET_DVR_CAM_DVI, or NET_DVR_CAM_HDMI.

byJointNo

Jointing No. of jointed signal source: 0-normal signal source, value larger than 0-jointed signal source.

byColorMode

Color mode: 0-custom, 1-sharp, 2-normal, 3-soft. If it is set to 0, you should set the color mode by the structure ***NET_DVR_VIDEOEFFECT*** .

byScreenServer

Link to screen server: 0-not link, 1-link

byRes1

Reserved, set to 0

dwInputSignalNo

Signal source No., it is compatible with **wInputNo**.

byRes

Reserved, set to 0

A.2.23 NET_DVR_IP_ADDRESS

Structure about device IP address information.

Structure Definition

```
struct{
  BYTE    byDevAddress[MAX_DOMAIN_NAME/*64*/];
  WORD    wDevPort;
  BYTE    byres[134];
}NET_DVR_IP_ADDRESS, *LPNET_DVR_IP_ADDRESS;
```

Members

byDevAddress

Device IP address

wDevPort

Device port number

byRes

Reserved

See Also

NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX

A.2.24 NET_DVR_IPADDR_UNION

IP Address Union

Member	Data Type	Description
szIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
szIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

A.2.25 NET_DVR_LED_AREA_COND

Configuration condition structure about LED area.

Structure Definition

```
struct{
  DWORD dwSize;
  DWORD dwVideoWallNo;
  DWORD dwLEDAreaNo;
  BYTE  byRes[32];
}NET_DVR_LED_AREA_COND, *LPNET_DVR_LED_AREA_COND;
```

Members

dwSize

Structure size.

dwVideoWallNo

Video wall No.

dwLEDAreaNo

LED area No.

byRes

Reserved, set to 0.

A.2.26 NET_DVR_LED_AREA_INFO

Structure about LED area information.

Structure Definition

```
struct{
  DWORD           dwSize;
  DWORD           dwLEDAreaNo;

                        NET_DVR_RECTCFG_EX
                        struRect;
  DWORD           dwaOutputNo[MAX_NUM_OUTPUT_CHANNEL/*512*/];
  BYTE          byRes[32];
}NET_DVR_LED_AREA_INFO, *LPNET_DVR_LED_AREA_INFO;
```

Members

dwSize

Structure size.

dwLEDAreaNo

LED area No.

struRect

Rectangle area information, see details in the structure ***NET_DVR_RECTCFG_EX*** .

dwaOutputNo

Outputs list

byRes

Reserved, set to 0.

A.2.27 NET_DVR_LED_AREA_INFO_LIST

Information structure about LED area list.

Structure Definition

```
struct{
  DWORD               dwSize;
  DWORD               dwLEDAreaNum;
  LPNET_DVR_LED_AREA_INFO    lpstruBuffer;
  DWORD               dwBufferSize;
  BYTE                byRes[32];
}NET_DVR_LED_AREA_INFO_LIST, *LPNET_DVR_LED_AREA_INFO_LIST;
```

Members

dwSize

Structure size.

dwLEDAreaNum

Number of LED areas

lpstruBuffer

A pointer that points to the buffer for storing LED area information (see details in the structure ***NET_DVR_LED_AREA_INFO***).

dwBufferSize

Buffer size, which depends on the number (**dwLEDAreaNum**) of structure ***NET_DVR_LED_AREA_INFO*** .

byRes

Reserved, set to 0.

A.2.28 NET_DVR_LOCAL_ABILITY_PARSE_CFG

Structure about capability of analysis library configuration.

Structure Definition

```
struct{
  BYTE      byEnableAbilityParse;
  BYTE      byRes[127];
}NET_DVR_LOCAL_ABILITY_PARSE_CFG, *LPNET_DVR_LOCAL_ABILITY_PARSE_CFG;
```

Members

byEnableAbilityParse

Whether to enable capability analysis library: 0-disable, 1-enable (default).

byRes

Reserved, set to 0.

Remarks

By default, the analog capability is disabled, you can enable the analog capability via this structure, and then call ***NET_DVR_GetDeviceAbility*** and load the "LocalXml.zip" to the directory of HCNetSDK to get the capabilities of devices.

A.2.29 NET_DVR_LOCAL_ASYNC_CFG

Structure about Asynchronous Configuration Parameter

Member	Data Type	Description
bEnable	BOOL	Whether to enable asynchronous configuration: "TRUE"-yes, "FALSE"-no (default).
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 60 bytes.

Remarks

- After enabling asynchronous configuration, the notifications about disconnection and reconnection of devices will be received in asynchronous mode. This function can be adopted when you need to manage tens of thousands of devices. By default, this function is disabled.
- After enabling asynchronous configuration, the interval configuration of heartbeat interaction turns invalid (related command: "NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV").
- After enabling asynchronous configuration, the API ***NET_DVR_SetConnectTime*** for setting network connection timeout turns invalid.

A.2.30 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT

Structure about encoding format conversion configuration.

Structure Definition

```
struct{
  CHAR_ENCODE_CONVERT    fnCharConvertCallBack
  BYTE                   byRes[256];
}NET_DVR_LOCAL_BYTE_ENCODE_CONVERT, *LPNET_DVR_LOCAL_BYTE_ENCODE_CONVERT;
```

Members

fnCharConvertCallBack

Callback function of encoding type conversion, see details in ***CHAR_ENCODE_CONVERT*** .

byRes

Reserved, set to 0.

Remarks

- The device character encoding type is returned by the login API.
- By default, the encoding type conversion is realized by the "libiconv.dll" of HCNetSDK, but the users can set the encoding type conversion callback via this structure and call their own encoding API to convert the encoding type.

A.2.31 NET_DVR_LOCAL_CERTIFICATION

Cettificate configuration parameter structure

Structure Definition

```
struct{
  char                    szLoadPath[MAX_FILE_PATH_LEN/*256*/];
  fnCertVerifyResultCallBack    fnCB;
  void                    *pUserData;
  BYTE                    byRes[64];
}NET_DVR_LOCAL_CERTIFICATION, *LPNET_DVR_LOCAL_CERTIFICATION;
```

Members

szLoadPath

Certificate saving path.

fnCB

Certificate verification callback function, see details below.

```
typedef BOOL(CALLBACK *fnCertVerifyResultCallBack)(
  DWORD                 uiResult,
  NET_DVR_CETTIFICATE_INFO   lpCertificateInfo,
  char                  *pUserData
);
```

uiResult

Certificate verification results: 0-verification failed, other values-verified.

lpCertificateInfo

Certificate information, see details in ***NET_DVR_CETTIFICATE_INFO*** .

pUserData

User data pointer.

pUserData

User data.

byRes

Reserved, set to 0.

See Also

NET_SDK_LOCAL_CFG_TYPE

A.2.32 NET_DVR_LOCAL_CFG_TYPE_PTZ

PTZ interaction configuration structure.

Structure Definition

```
struct{
  BYTE    byWithoutRecv;
  BYTE    byRes[63];
}NET_DVR_LOCAL_PTZ_CFG, *LPNET_DVR_LOCAL_PTZ_CFG;
```

Members

byWithoutRecv

Whether to receive the response from device: 0-yes, 1-no.

byRes

Reserved, set to 0

Remarks

This configuration is applicable to 3G network.

A.2.33 NET_DVR_LOCAL_CHECK_DEV

Heartbeat time interval configuration structure.

Structure Definition

```
struct{
  DWORD      dwCheckOnlineTimeout;
  DWORD      dwCheckOnlineNetFailMax;
  BYTE       byRes[256];
}NET_DVR_LOCAL_CHECK_DEV, *LPNET_DVR_LOCAL_CHECK_DEV;
```

Members

dwCheckOnlineTimeout

Online health monitoring time interval, unit: ms, range: 30-120 (s), 0-120s (default), the recommended value is 30s.

dwCheckOnlineNetFailMax

The maximum number of network failure attempts, if the failure attempts are larger than this threshold, exception message will the called back. 0-1 (default), the recommended value is 3.

byRes

Reserved, set to 0.

A.2.34 NET_DVR_LOCAL_GENERAL_CFG

General configurations structure.

Structure Definition

```
struct{
  BYTE      byExceptionCbDirectly;
  BYTE      byNotSplitRecordFile;
  BYTE      byResumeUpgradeEnable;
  BYTE      byAlarmJsonPictureSeparate;
  BYTE      byRes[4];
  UINT64    i64FileSize;
  DWORD     dwResumeUpgradeTimeout;
  BYTE      byAlarmReconnectMode;
  BYTE      byStdXmlBufferSize;
  BYTE      byMultiplexing;
  BYTE      byFastUpgrade;
  BYTE      byRes[232];
}NET_DVR_LOCAL_GENERAL_CFG, *LPNET_DVR_LOCAL_GENERAL_CFG;
```

Members

byExceptionCbDirectly

Exception callback type: 0-callback via thread pool, 1-callback via upper-layer.

byNotSplitRecordFile

Whether to subpackage the local video files: 0-yes (default), 1-no.

byResumeUpgradeEnable

Whether to enable upgrading ANR (Automatic Network Replenishment): 0-disable (default), 1-enable.

byAlarmJsonPictureSeparate

Whether to separate the alarm data and the alarm picture which will be transmitted in JSON format: 0-not separate, 1-separate (the **lCommand** in the callback function will be "COMM_ISAPI_ALARM").

byRes

Reserved.

i64FileSize

Maximum file size, unit: byte. When subpackaging is enabled, if the saved video file size is larger than the value of this parameter, the file will be subpackaged to multiple file segments for storage.

dwResumeUpgradeTimeout

ANR reconnection timeout, unit: millisecond.

byAlarmReconnectMode

Reconnection mode: 0-dependent thread reconnection (default), 1-thread pool reconnection.

byStdXmlBufferSize

Buffer size for receiving data transmitted by ISAPI: 1-1 MB, other values-default.

byMultiplexing

Whether to enable multiplexing of normal link (non-TLS link): 0-disable, 1-enable.

byFastUpgrade

Upgrading mode: 1-normal upgrading, 2-fast upgrading.

byRes1

Reserved.

A.2.35 NET_DVR_LOCAL_LOG_CFG

Log configuration structure.

Structure Definition

```
struct{
  WORD    wSDKLogNum;
  BYTE    byRes[254];
}NET_DVR_LOCAL_LOG_CFG, *LPNET_DVR_LOCAL_LOG_CFG;
```

Members

wSDKLogNum

Number of log files in overwritten mode, "0"-10 log files (default).

byRes

Reserved, set to 0.

A.2.36 NET_DVR_LOCAL_MEM_POOL_CFG

Local configuration structure of storage pool.

Structure Definition

```
struct{
  DWORD    dwAlarmMaxBlockNum;
  DWORD    dwAlarmReleaseInterval;
  BYTE    byRes[60];
}NET_DVR_LOCAL_MEM_POOL_CFG, *LPNET_DVR_LOCAL_MEM_POOL_CFG;
```

Members

dwAlarmMaxBlockNum

The maximum number of memory blocks can be applied, the maximum size of each applied block is 64MB, if the required memory block size is larger than the threshold, do not apply for it

from the system. If the value of this parameter is set to 0, it refers that the number of memory block can be applied is not limited.

dwAlarmReleaseInterval

The time interval between each free memory blocks to be released, unit: s, 0-not release the free memory.

byRes

Reserved, set to 0.

A.2.37 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG

Structure about timeout configuration by module.

Structure Definition

```
struct{
  DWORD   dwPreviewTime;
  DWORD   dwAlarmTime;
  DWORD   dwVodTime;
  DWORD   dwElse;
  BYTE    byRes[512];
}NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG, *LPNET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG;
```

Members

dwPreviewTime

Live view module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwAlarmTime

Alarm module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwVodTime

Playback module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwElse

Other modules' receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

byRes

Reserved, set to 0.

A.2.38 NET_DVR_LOCAL_PORT_MULTI_CFG

Configuration parameter structure of port multiplier.

Structure Definition

```
struct{
  BOOL    bEnable;
  BYTE    byRes[60];
}NET_DVR_LOCAL_PORT_MULTI_CFG, *LPNET_DVR_LOCAL_PORT_MULTI_CFG;
```

Members

bEnable

Whether to enable port multiplier: true-yes.

byRes

Reserved, set to 0.

See Also

NET_SDK_LOCAL_CFG_TYPE

A.2.39 NET_DVR_LOCAL_PROTECT_KEY_CFG

Key Parameter Structure

Member	Data Type	Description
byProtectKey	Array of BYTE	Key, the default value is 0. The maximum size is 128 bytes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 128 bytes.

A.2.40 NET_DVR_LOCAL_SDK_PATH

Path Information Structure for Loading Component Libraries

Member	Data Type	Description
sPath	Array of char	Component libraries' addresses
byRes	Array of BYTE	Reserved.

Remarks

If the path of HCNetSDKCom folder and HCNetSDK libraries are same, but the path of executable programs are different, you can call ***NET_DVR_SetSDKInitCfg*** to specify the path of HCNetSDKCom folder to make sure the component libraries are loaded normally.

A.2.41 NET_DVR_LOCAL_STREAM_CALLBACK_CFG

Key Parameter Structure

Member	Data Type	Description
byPlayBackEndFlag	BYTE	Whether to call back playback end flag:0-No, 1-Yes
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 255 bytes.

A.2.42 NET_DVR_LOCAL_TALK_MODE_CFG

Two-way audio configuration structure.

Structure Definition

```
struct{
  BYTE   byTalkMode;
  BYTE   byRes[127];
}NET_DVR_LOCAL_TALK_MODE_CFG, *LPNET_DVR_LOCAL_TALK_MODE_CFG;
```

Members

byTalkMode

Two-way audio mode: 0-enable two-way audio library (default), 1-enable Windows API mode.

byRes

Reserved, set to 0.

Remarks

If the two-way audio library is enabled, you must load the "AudioIntercom.dll" and "OpenAL32.dll".

A.2.43 NET_DVR_LOCAL_TCP_PORT_BIND_CFG

Local binding configuration structure of TCP port.

Structure Definition

```
struct{
  WORD     wLocalBindTcpMinPort;
  WORD     wLocalBindTcpMaxPort;
  BYTE     byRes[60];
}NET_DVR_LOCAL_TCP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_TCP_PORT_BIND_CFG;
```

Members

wLocalBindTcpMinPort

The minimum TCP port number to be bound locally.

wLocalBindTcpMaxPort

The maximum TCP port number to be bound locally.

byRes

Reserved, set to 0.

Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.
- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

A.2.44 NET_DVR_LOCAL_UDP_PORT_BIND_CFG

Local binding configuration structure of UDP port.

Structure Definition

```
struct{
  WORD    wLocalBindUdpMinPort;
  WORD    wLocalBindUdpMaxPort;
  BYTE    byRes[60];
}NET_DVR_LOCAL_UDP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_UDP_PORT_BIND_CFG;
```

Members

wLocalBindUdpMinPort

The minimum UDP port number to be bound locally.

wLocalBindUdpMaxPort

The maximum UDP port number to be bound locally.

byRes

Reserved, set to 0.

Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.
- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

A.2.45 NET_DVR_MATRIX_CHAN_INFO_V41

Structure about auto-switch decoding channel information.

Structure Definition

```
struct{
  BYTE                byEnable;
  BYTE                byStreamMode;
  BYTE                byRes[2];

                          NET_DVR_DEC_STREAM_MODE
                           uDecStreamMode;
}NET_DVR_MATRIX_CHAN_INFO_V41,*LPNET_DVR_MATRIX_CHAN_INFO_V41;
```

Members

byEnable

Enable/disable: 0-disable, 1-enable

byStreamMode

Streaming mode: 0-invalid, 1-by IP address or domain name, 2-by URL, 3-by DDNS

byRes

Reserved, set to 0.

uDecStreamMode

Streaming configuration information, see details in the union ***NET_DVR_DEC_STREAM_MODE*** .

A.2.46 NET_DVR_MATRIX_DECCHAN_CONTROL

Structure about zoom control parameters of decoding.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byDecChanScaleStatus;
  BYTE     byDecodeDelay;
  BYTE     byEnableSpartan;
  BYTE     byLowLight;
  BYTE     byNoiseReduction;
  BYTE     byDefog;
  BYTE     byEnableVcaDec;
  BYTE     byRes1;
  DWORD    dwAllCtrlType;
  BYTE     byRes[56];
}NET_DVR_MATRIX_DECCHAN_CONTROL,*LPNET_DVR_MATRIX_DECCHAN_CONTROL;
```

Members

dwSize

Structure size

byDecChanScaleStatus

Display of zoom control of decoding channel: 1-zoom display, 0-real display

byDecodeDelay

Decoding delay: 0- default, 1- most real-time, 2- more real-time, 3- real-time and fluent, 4- more fluent, 5- most fluent, 0xff- automatically adjust

byEnableSpartan

Whether to enable fluent display: 0-no, 1-yes

byLowLight

Low light: 0- disable, 1 to 8 indicates the low light level; the higher the level is, the higher the strength of the low light will be.

byNoiseReduction

Whether to enable 3D noise reduction: 0-disable, 1- enable, 2- automatically adjust

byDefog

Whether to enable defog: 0- disable, 1 to 7 indicates the defog level, the higher the level is, the higher the strength of the defog will be.

byEnableVcaDec

Whether to enable smart decoding: 0-no, Others-yes

byRes1

Reserved, set as 0.

dwAllCtrlType

Whether to configure the type of operating all windows at a same time 0- not configure, 1- configure; this member is valid for configuring parameters and is represented by byte.

byRes

Reserved, set as 0.

Remarks

Check the device capability (whether to supports smart decoding) via the video wall capability(WallAbility); API: NET_DVR_GetDeviceAbility, capability type:WALL_ABILITY, node:<**VcaDecode**>.

A.2.47 NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX

Parameter structure about remote decoding and playback on video wall.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwDecChannel;
  BYTE     byAddressType;
  BYTE     byChannelType;
  BYTE     byres[2];
  BYTE     sUserName[NAME_LEN/*32*/];
  BYTE     sPassword[PASSWD_LEN/*16*/];
  DWORD    dwChannel;
  BYTE     byStreamId[STREAM_ID_LEN/*32*/];
  DWORD    dwPlayMode;
  union
  {
    BYTE              byRes[200];
    NET_DVR_IP_ADDRESS    struIpAddr;
    NET_DVR_DDNS_ADDRESS  struDdnsAddr;
  }unionAddr
  union
  {
    BYTE                  byRes[128];
    NET_DVR_PLAY_BACK_BY_TIME   struPlayBackByTime;
    char                  sFileName[128];
  }unionPlayBackInfo;
}NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX, *LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_EX;
```

Members

dwSize

Structure size.

dwDecChannel

Decoding channel No. or window No.

byAddressType

Device address type: 0-IP address, 1-DDNS domain.

byChannelType

Channel type: 0-Normal channel, 1-Channel-Zero, 2-Stream ID

byres

Reserved

sUserName

User name

sPassword

Password

dwChannel

Devoice channel No.

byStreamId

Stream ID, this parameter is valid only when the channel type (**byChannelType**) is Stream ID.

dwPlayMode

Playback mode, 0-by file, 1-by time

unionAddr

Device address union, see its members below:

byRes

Union size (200 bytes)

struIpAddr

Device IP address, see details in the structure ***NET_DVR_IP_ADDRESS*** .

struDdnsAddr

Device DDNS domain name, see details in the structure ***NET_DVR_DDNS_ADDRESS*** .

unionPlayBackInfo

Playback parameter union, see its members below:

byRes

Union size (200 bytes)

struPlayBackByTime

File information (when playback by time, see details in the structure ***NET_DVR_PLAY_BACK_BY_TIME***)

sFileName

File information (when playback by file name)

A.2.48 NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS

Structure of playback status.

Structure Definition

```
struct{
  DWORD   dwSize;
  DWORD   dwCurMediaFileLen;
  DWORD   dwCurMediaFilePosition;
  DWORD   dwCurMediaFileDuration;
  DWORD   dwCurPlayTime;
  DWORD   dwCurMediaFIleFrames;
  DWORD   dwCurDataType;
  BYTE    res[72];
}NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS,*LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS;
```

Members

dwSize

Structure size

dwCurMediaFileLen

Length of currently playing file

dwCurMediaFilePosition

Playing position of currently playing file

dwCurMediaFileDuration

Duration of currently playing file

dwCurPlayTime

The time of currently playing

dwCurMediaFIleFrames

Total frames of currently playing file

dwCurDataType

Data type of current tansmission: 19- head file, 20- stream data, 21- flag of playing end

res

Reserved, please set it to 0

A.2.49 NET_DVR_MATRIX_DEC_REMOTE_PLAY_V50

Table A-1 Structure about Remote Decoding and Playback Parameters on Video Wall

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwDecChannel	DWORD	Decoding channel No.
byAddressType	BYTE	Device address type: 0-IP address, 1-domain name, 2-URL.
byChannelType	BYTE	Channel type: 0-normal channel, 1-channel-zero, 2-stream ID.
byStreamEncrypt	BYTE	Whether to encrypt the stream: 0-no, 1-yes.
byRes1	BYTE	Reserved. The maximum length is one byte.
sUserName	BYTE	User name. The maximum length is "NAME_LEN" (32 bytes).
sPassword	BYTE	Password. The maximum length is "PASSWD_LEN" (16 bytes).
dwChannel	DWORD	Device channel No.
byStreamId	BYTE	Stream ID. This member is valid when the value of byChannelType is 2. The maximum length is "STREAM_ID_LEN" (32 bytes).
dwPlayMode	DWORD	Decoding and playing mode: 0-by file, 1-by time.
unionAddr	union	Address parameter union. See details in the table below.
unionPlayBackInfo	union	Playback information union. See details in the table below.

Member	Data Type	Description
struURL	***NET_DVR_PU_STREAM_URL***	Streaming URL. This member is valid when the value of byAddressType is 2.
sStreamPassword	BYTE	Stream encryption password. The maximum length is "STREAM_PASSWD_LEN" (12 bytes).
byRes2	BYTE	Reserved. The maximum length is 116 bytes.

Table A-2 Address Parameter Union (unionAddr)

Member	Data Type	Description
byRes	BYTE	Reserved. The maximum length is 200 bytes.
struIpAddr	***NET_DVR_IP_ADDRESS***	IP address. This member is valid when the value of byAddressType is 0.
struDdnsAddr	***NET_DVR_DDNS_ADDRESS***	Domain name. This member is valid when the value of byAddressType is 1.

Table A-3 Playback Information Union (unionPlayBackInfo)

Member	Data Type	Description
byRes	BYTE	Reserved. The maximum length is 128 bytes.
struPlayBackByTime	***NET_DVR_PLAY_BACK_BY_TIME***	Information when playback by time.
sFileName	char	Information when playback by file. The maximum length is 128 bytes.

A.2.50 NET_DVR_MATRIX_LOOP_DECINFO_V41

Structure about auto-switch decoding parameters.

Structure Definition

```
struct{
  DWORD                  dwSize;
  DWORD                  dwPoolTime;

                         NET_DVR_MATRIX_CHAN_INFO_V41
                         struchanConInfo[MAX_CYCLE_CHAN_V30/*64*/];
  BYTE                 byRes[16];
}NET_DVR_MATRIX_LOOP_DECINFO_V41,*LPNET_DVR_MATRIX_LOOP_DECINFO_V41;
```

Members

dwSize

Structure size

dwPoolTime

Auto-switch interval, unit: second

struchanConInfo

Auto-switch channel information, see details in the structure ***NET_DVR_MATRIX_CHAN_INFO_V41*** .

byRes

Reserved, set to 0

A.2.51 NET_DVR_MATRIX_PASSIVEMODE

Structure about passive decoding parameters.

Structure Definition

```
struct{
  WORD         wTransProtol;
  WORD         wPassivePort;


                         struMcastIP;
  BYTE         byStreamType;
  BYTE         byRes[7];
}NET_DVR_MATRIX_PASSIVEMODE,*LPNET_DVR_MATRIX_PASSIVEMODE;
```

Members

wTransProtol

Transmission protocol type: 0-TCP, 1-UDP, 2-MCAST

wPassivePort

TCP or UDP port. For TCP, the default port number is 8000; for UDP, the port number should be set according to different decoding channel.

struMcastIP

This parameter is valid only when the transmission protocol is MCAST, and it is the multicast address (currently reserved), see details in the structure .

byStreamType

Stream type: 1-real-time stream, 2-file stream.

byRes

Reserved, set to 0

A.2.52 NET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30

Structure of transparent channel configuration.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     by232IsDualChan;
  BYTE     by485IsDualChan;
  DWORD    byRes[2];
  NET_DVR_MATRIX_TRAN_CHAN_INFO_V30   struTranInfo[MAX_SERIAL_NUM/*64*/];
}NET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30,*LPNET_DVR_MATRIX_TRAN_CHAN_CONFIG_V30;
```

Members

dwSize

Structure size

by232IsDualChan

Set 232 some transparent channels to full-duplex, range: [1,MAX_SERIAL_NUM].

by485IsDualChan

Set 485 some transparent channels to full-duplex, range: [1,MAX_SERIAL_NUM].

byRes

Reserved, please set it to 0.

struTranInfo

Structure of transparent channel parameters. The number of transparent channels supported at the same time is MAX_SERIAL_NUM.

A.2.53 NET_DVR_MESSAGE_CALLBACK_PARAM_V51

Alarm Callback Configuration Parameters

Key Parameter Structure

Member	Data Type	Description
byVcaAlarmJsonType	BYTE	JSON format for alarm transmission (COMM_VCA_ALARM): 0-new JSON format, 1-old JSON format.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 63 bytes.

A.2.54 NET_DVR_MIME_UNIT

Input Content Details Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
szContentType	Array of char	Content type (corresponds to **Content-Type** field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
szName	Array of char	Content name (corresponds to **name** field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to **filename** field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by **szFilename**.
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

See Also

NET_DVR_XML_CONFIG_INPUT

A.2.55 NET_DVR_OUT_PARAM

Structure about output parameters.

Structure Definition

```
struct{
                              NET_DVR_BUF_INFO
                               struOutBuf;
  void*            lpStatusList;
  BYTE             byRes[32];
}NET_DVR_OUT_PARAM,*LPNET_DVR_OUT_PARAM;
```

Members

struOutBuf

Buffer for storing output parameters

pStatusList

Buffer for storing statuses

byRes

Reserved, set to 0

A.2.56 NET_DVR_PASSIVEDECODE_CONTROL

Structure about passive decoding control parameters.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwPlayCmd;
  DWORD    dwCmdParam;
  BYTE     byRes[16];
}NET_DVR_PASSIVEDECODE_CONTROL,*LPNET_DVR_PASSIVEDECODE_CONTROL;
```

Members

dwSize

Structure size

dwPlayCmd

Passive decoding control commands, see details in the following table:

Command (dwPlayCmd)	Command No.	Description
PASSIVE_DEC_PAUSE	1	Pause passive decoding (valid for file stream only).
PASSIVE_DEC_RESUME	2	Resume passive decoding (valid for file stream only).
PASSIVE_DEC_FAST	3	Fast forward (valid for file stream only).
PASSIVE_DEC_SLOW	4	Slow forward (valid for file stream only).
PASSIVE_DEC_NORMAL	5	Decode in ×1 speed (valid for file stream only).
PASSIVE_DEC_ONEBYONE	6	Decode in single frame (reserved)
PASSIVE_DEC_AUDIO_ON	7	Turn on audio.
PASSIVE_DEC_AUDIO_OFF	8	Turn off audio.
PASSIVE_DEC_RESETBUFFER	9	Clear buffer.

dwCmdParam

Control parameter. The requirement of this parameter depends on the control command.

A.2.57 NET_DVR_PLAY_BACK_BY_TIME

Structure about playback by time.

Structure Definition

```
struct{
                        NET_DVR_TIME
                         StartTime;

                        NET_DVR_TIME
                         StopTime;
}NET_DVR_PLAY_BACK_BY_TIME, *LPNET_DVR_PLAY_BACK_BY_TIME;
```

Members

StartTime

Playback start time.

StopTime

Playback end time

See Also

NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX

A.2.58 NET_DVR_PU_STREAM_CFG_V41

Dynamic decoding parameter structure.

Structure Definition

```
struct{
  DWORD                dwSize;
  BYTE                 byStreamMode;
  BYTE                 byStreamEncrypt;
  BYTE                 byRes1[2];
  NET_DVR_DEC_STREAM_MODE    uDecStreamMode;
  DWORD                dwDecDelayTime;
  BYTE                 sStreamPassword[STREAM_PASSWD_LEN/*12*/];
  BYTE                 byRes2[48];
}NET_DVR_PU_STREAM_CFG_V41,*LPNET_DVR_PU_STREAM_CFG_V41;
```

Members

dwSize

Structure size

byStreamMode

Streaming mode: 0-invalid, 1-get stream by IP address or domain name, 2-get stream by URL, 3-get stream from device by DDNS.

byStreamEncrypt

Whether to encrypt the stream: 0-no, 1-yes.

byRes1

Reserved, set to 0

uDecStreamMode

Streaming configuration parameters, see the structure ***NET_DVR_DEC_STREAM_MODE*** for details.

dwDecDelayTime

Decoding delay time duration, unit: millisecond.

sStreamPassword

Stream encryption password. The sensitive information should be encrypted.

byRes2

Reserved, set to 0.

A.2.59 NET_DVR_PU_STREAM_URL

Configuration parameter structure about getting stream by URL.

Structure Definition

```
struct{
  BYTE    byEnable;
  BYTE    strURL[240];
  BYTE    byTransPortocol;
  WORD    wIPID;
  BYTE    byChannel;
  BYTE    byRes[7];
}NET_DVR_PU_STREAM_URL,*LPNET_DVR_PU_STREAM_URL;
```

Members

byEnable

Enable/disable getting stream by URL: 0-disable, 1-enable.

strURL

Stream URL

byTransPortocol

Transfer protocol type: 0-TCP, 1-UDP

wIPID

Device ID= iDevInfoIndex + iGroupNO*64 +1

byChannel

Device channel No.

byRes

Reserved, set to 0.

Remarks

The stream URL format is {rtsp://ip[:port]/urlExtension}[?username=username][?password=password][?linkmode=linkmode]. You can also customize the URL format if the network camera supports custom URL.

A.2.60 NET_DVR_RECTCFG_EX

Configuration parameter structure about rectangle region position.

Structure Definition

```
struct{
  DWORD   dwXCoordinate;
  DWORD   dwYCoordinate;
  DWORD   dwWidth;
  DWORD   dwHeight;
  BYTE    byRes[4];
}NET_DVR_RECTCFG_EX, *LPNET_DVR_RECTCFG_EX;
```

Members

dwXCoordinate

X-coordinate of the point in the upper-left corner of the rectangle region.

dwYCoordinate

X-coordinate of the point in the upper-left corner of the rectangle region.

dwWidth

Rectangle region width

dwHeight

Rectangle region height

byRes

Reserved, set to 0

A.2.61 NET_DVR_RGB_COLOR

Structure about color parameters.

Structure Definition

```
struct{
  BYTE    byRed;
  BYTE    byGreen;
  BYTE    byBlue;
  BYTE    byRes;
}NET_DVR_RGB_COLOR, *LPNET_DVR_RGB_COLOR;
```

Members

byRed

Red component in RGB color model.

byGreen

Green component in RGB color model.

byBlue

Blue component in RGB color model.

byRes

Reserved

Remarks

The color component value is based on RGB888 standard.

A.2.62 NET_DVR_RTSP_PARAMS_CFG

RTSP Parameter Structure

Member	Data Type	Description
dwMaxBuffRoomNum	DWORD	Maximum number of buffers for RTP over UDP sorting, the default value is 20. If the value is 0, it indicates that the member is invalid. One buffer size is about 10 KB, more number of buffers indicates higher sorting ability, more fluent, and longer delay.
byUseSort	BYTE	Whether to enable RTP over UDP sorting: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 123 bytes.

A.2.63 NET_DVR_SCENE_CONTROL_INFO

Structure about scene control parameters.

Structure Definition

```
struct{
  DWORD                 dwSize;

                          NET_DVR_VIDEO_WALL_INFO
                          struVideoWallInfo;
  DWORD                 dwCmd;
  BYTE                byRes[4];
}NET_DVR_SCENE_CONTROL_INFO,*LPNET_DVR_SCENE_CONTROL_INFO;
```

Members

dwSize

Structure size

struVideoWallInfo

Video wall information, see the structure ***NET_DVR_VIDEO_WALL_INFO*** for details.

dwCmd

Scene control command: 1-switch scene (it is invalid if switching to the current scene), 2-initialize scene (clear configured scene parameters, if the scene is displayed, the display image will be cleared), 3-force switching scene, 4-save current view as a scene

byRes

Reserved, set to 0

A.2.64 NET_DVR_SEARCH_EVENT_PARAM_V50

Condition structure about searching videos by event.

Structure Definition

```
struct{
  WORD                    wMajorType;
  WORD                    wMinorType;
  NET_DVR_TIME_SEARCH_COND    struStartTime;
  NET_DVR_TIME_SEARCH_COND    struEndTime;
  BYTE                  byLockType;
  BYTE                  byRes[255];
  union{
    BYTE                byLen[SEARCH_EVENT_INFO_LEN_V40/*800*/];
    struct{
      WORD                wAlarmInNo[128];
      BYTE                byRes[544];
    }struAlarmParam;
    struct{
      WORD                wMotDetChanNo[MAX_CHANNUM_V30/*64*/];
      BYTE                byRes[672];
    }struMotionParam;
    struct{
      WORD                wChanNo[MAX_CHANNUM_V30/*64*/];
      BYTE                byRuleID;
      BYTE                byRes[671];
    }struVcaParam;
    struct{
      BYTE                byRoomIndex;
      BYTE                byRes[799];
    }struInquestParam;
    struct{
      BYTE                byAll;
      BYTE                byRes1[3];
      WORD                wChanNo[MAX_CHANNUM_V30/*64*/];
      BYTE                byRes1[668];
    }struVCADetect;
```

```
  struct{
   NET_DVR_STREAM_INFO   struIDInfo;
   DWORD              dwCmdType;
   BYTE               byBackupVolumeNum;
   BYTE               byRes[723];
  }struStreamIDParam;
  struct{
   WORD               wChannel[MAX_CHANNUM_V30/*64*/];
   BYTE               byAllChan;
   BYTE               byCaseSensitive;
   BYTE               byCombinateMode;
   BYTE               byRes1;
   char               sKeyWord[MAX_POS_KEYWORDS_NUM/*3*/][MAX_POS_KEYWORD_LEN/*128*/];
   BYTE               byRes[284];
  }struPosAlarm;
  struct{
   BYTE               byCaseNo[SEARCH_CASE_NO_LEN/*56*/];
   BYTE               byCaseName[SEARCH_CASE_NAME_LEN/*100*/];
   BYTE               byLitigant1[SEARCH_LITIGANT_LEN/*32*/];
   BYTE               byLitigant2[SEARCH_LITIGANT_LEN/*32*/];
   BYTE               byChiefJudge[SEARCH_CHIEF_JUDGE_LEN/*32*/];
   BYTE               byCaseType;
   BYTE               byRes[547];
  }struTrialParam;
  struct{
   DWORD              dwMajor;
   DWORD              dwMinor;
   BYTE               byCardNo[ACS_CARD_NO_LEN/*32*/];
   BYTE               byName[NAME_LEN/*32*/];
   BYTE               byMACAddr[MACADDR_LEN/*6*/];
   BYTE               byRes[722];
  }struACSAlarm;
  struct{
   WORD               wDeviceType;
   WORD               wEventType;
   WORD               wChannel[MAX_CHANNUM_V30/*64*/];
   BYTE               byAllChan;
   BYTE               byCaseSensitive;
   BYTE               byCombinateMode;
   BYTE               bySearchType;
   char               sKeyWord[MAX_POS_KEYWORDS_NUM/*3*/][MAX_POS_KEYWORD_LEN/*128*/];
   WORD               wZoneNo;
   BYTE               byRes[278];
  }struIOTAlarm;
 }uSeniorParam;
}NET_DVR_SEARCH_EVENT_PARAM_V50,*LPNET_DVR_SEARCH_EVENT_PARAM_V50;
```

Members

wMajorType

Major alarm/event types, see details below:

```
enum _MAIN_EVENT_TYPE_{
  EVENT_MOT_DET       = 0,
  EVENT_ALARM_IN      = 1,
  EVENT_VCA_BEHAVIOR   = 2,
  EVENT_INQUEST       = 3,
  EVENT_VCA_DETECTION  = 4,
  EVENT_POS           = 5,
  EVENT_TRIAL_CASE     = 6,
  EVENT_ACS_CASE      = 7,
  EVENT_IOT_CASE      = 8,
  EVENT_STREAM_INFO    = 100,
}MAIN_EVENT_TYPE
```

EVENT_MOT_DET

Motion detection

EVENT_ALARM_IN

Alarm input

EVENT_VCA_BEHAVIOR

Behavior analysis

EVENT_INQUEST

Inquest event (not support)

EVENT_VCA_DETECTION

VCA detection

EVENT_POS

POS information

EVENT_TRIAL_CASE

Trial case information (not support)

EVENT_ACS_CASE

Access control event

EVENT_IOT_CASE

IoT device event

EVENT_STREAM_INFO

Stream ID information

wMinorType

Minor alarm/event types, which vary with the major types. 0xffff-all types. For motion detection, alarm input, and POS recording, there is no minor type, so this parameter is set to "oxffff"; for access control event, refer to the integration manual of access control applications; for the major types of behavior analysis, VCA detection, and stream ID information, the corresponding minor types are shown below:

Table A-4 EVENT_VCA_BEHAVIOR

Minor Type	Value	Description
EVENT_TRAVERSE_PLANE	0	Line crossing
EVENT_ENTER_AREA	1	Entering the area,support regional rule
EVENT_EXIT_AREA	2	Leaving the area, support regional rule
EVENT_INTRUSION	3	Perimeter intrusion, support regional rule
EVENT_LOITER	4	Loitering, support regional rule
EVENT_LEFT_TAKE	5	Droping/picking up, support regional rule
EVENT_PARKING	6	Parking, support regional rule
EVENT_RUN	7	Running, support regional rule
EVENT_HIGH_DENSITY	8	People density in the area, support regional rule
EVENT_STICK_UP	9	Sticking a note, support regional rule
EVENT_INSTALL_SCANNER	10	Installing card reader, support regional rule
EVENT_OPERATE_OVER_TIME	11	Operation timeout
EVENT_FACE_DETECT	12	Abnormal face detection
EVENT_LEFT	13	Unattended baggage
EVENT_TAKE	14	Object removal
EVENT_LEAVE_POSITION	15	Absence event
EVENT_TRAIL_INFO	16	Tailing
EVENT_HUMAN_ENTER	18	Human entrance
EVENT_FALL_DOWN_INFO	19	Falling down
EVENT_OBJECT_PASTE	20	Sticking script area
EVENT_FACE_CAPTURE_INFO	21	Normal face
EVENT_MULTI_FACES_INFO	22	Multiple faces

Minor Type	Value	Description
EVENT_AUDIO_ABNORMAL_INFO	23	Sudden change of sound intensity
EVENT_DETECT	24	VCA detection
EVENT_SUNGLASSES_FACE_INFO	25	Face with sunglasses
EVENT_CALLING_FACE_INFO	26	Person is making call
EVENT_VIOLENT_MOTION	27	Violent motion
EVENT_SAFETY_HELMET	28	Hart hat detection

Table A-5 EVENT_VCA_DETECTION

Minor Type	Value	Description
EVENT_VCA_TRAVERSE_PLANE	1	Line crossing
EVENT_FIELD_DETECTION	2	Intrusion detection
EVENT_AUDIO_INPUT_ALARM	3	Audio loss detection
EVENT_SOUND_INTENSITY_ALARM	4	Sudden increase of sound increase/decrease detection
EVENT_FACE_DETECTION	5	Face detection
EVENT_VIRTUAL_FOCUS_ALARM	6	Defocus detection
EVENT_SCENE_CHANGE_ALARM	7	Scene change detection
EVENT_PIR_ALARM	8	PIR alarm
EVENT_ENTER_REGION	9	Region entrance
EVENT_EXIT_REGION	10	Region exiting
EVENT_LOITERING	11	Loitering
EVENT_GROUPDETECTION	12	People gathering in the area, support regional rule
EVENT_RAPIDMOVE	13	Fast moving
EVENT_PARK	14	Parking
EVENT_UNATTENDED_BAGGAGE	15	Unattended baggage

Minor Type	Value	Description
EVENT_ATTENDEDBAGGAGE	16	Object removal
EVENT_VEHICLE_DETECTION	17	Vehicle detection

Table A-6 EVENT_STREAM_INFO

Minor Type	Value	Description
EVENT_STREAM_ID	0	Stream ID
EVENT_TIMING	1	Timing record
EVENT_MOTION_DETECT	2	Motion detection
EVENT_ALARM	3	Alarm record
EVENT_ALARM_OR_MOTION_DETECT	4	Alarm or motion detection
EVENT_ALARM_AND_MOTION_DETECT	5	Alarm and motion detection
EVENT_COMMAND_TRIGGER	6	Command triggering
EVENT_MANNUAL	7	Manual record
EVENT_BACKUP_VOLUME	8	Storage volume record
STREAM_EVENT_SEMAPHORE	9	Sensor alarm
STREAM_EVENT_HIDE	10	Video tempering
STREAM_EVENT_INVERSE	11	Driving in opposite directio
STREAM_EVENT_VIDEO_LOST	. 12	Video loss
STREAM_EVENT_WIRELESS_ALARM	13	Wirless alarm
STREAM_EVENT_PIR_ALARM	14	PIR alarm
STREAM_EVENT_CALLHELP_ALARM	15	Call for help alarm
STREAM_EVENT_FACESNAP	16	Face capture
STREAM_EVENT_FACE_DETECTION	17	Face detection
STREAM_EVENT_ITS_PLATE	18	Vehicle detection
STREAM_EVENT_PDC	19	People counting

Minor Type	Value	Description
STREAM_EVENT_ SCENECHANGE_DETECTION	20	Scene change
STREAM_EVENT_DEFOCUS_ DETECTION	21	Defocus detection
STREAM_EVENT_ AUDIOEXCEPTION	22	Audio exception
VCA_EVENT_TRAVERSE_ PLANE	23	Line crossing
VCA_EVENT_INTRUSION	24	Intrusion
VCA_EVENT_ENTER_AREA	25	Region entrance
VCA_EVENT_EXIT_AREA	26	Region exiting
VCA_EVENT_LOITER	27	Loitering detection
VCA_EVENT_HIGH_DENSITY	28	People gathering
VCA_EVENT_RUN	29	Fast moving
VCA_EVENT_PARKING	30	Illegal parking
VCA_EVENT_LEFT	31	Unattended baggage
VCA_EVENT_TAKE	32	Object removal

struStartTime

Start time of search, refer to the structure for details.

struEndTime

Stop time of search, refer to the structure for details.

byLockType

Whether to lock: 0xff-all, 0-no, 1-yes.

byRes

Reserved, set to 0.

uSeniorParam

Condition structure of searching for video files based on event/alarm, see details below:

byLen

Union size, which is 800 bytes.

struAlarmParam

Condition structure of searching for video files based on alarm input alarm, see details below:

wAlarmInNo

Alarm input No., "0xffff"-the following are invalid. For example, wAlarmInNo[0]==1&&wAlarmInNo[1]==2 indicates the alarm input event of alarm input No.1 and alarm input No.2.

byRes

Reserved, set to 0.

struMotionParam

Condition structure of searching for video files based on motion detection, see details below:

wMotDetChanNo

The channel No., "0xffff"-the following are invalid. For example,wMotDetChanNo[0]==1&&wMotDetChanNo[1]==2 indicates the motion detection event of channel No.1 and channel No.2.

byRes

Reserved, set to 0.

struVcaParam

Condition structure of searching for video files based on behavior analysis, see details below:

wChanNo

The event triggered channel No., "0xffff"-the following are invalid. For example, wChanNo[0]==1&&wChanNo[1]==2 indicates the behavior analysis event of channel No.1 and channel No.2.

byRuleID

Rule ID, 0xff-all

byRes

Reserved, set to 0.

struInquestParam

Condition structure of searching for video files based on inquest event, see details below:

byRoomIndex

The inquest room index No., starts from 1.

byRes

Reserved, set to 0.

struVCADetect VCA detection

Condition structure of searching for video files based on VCA detection, see details below:

byAll

0-Search by channel No., 1-Search in all channels.

byRes1

Reserved, set to 0.

wChanNo

VCA detection channel No., "0xffff"-the following are invalid. For example, wChanNo[0]==1&&wChanNo[1]==2 indicates the VCA detection of channel No.1 and channel No.2.

byRes

Reserved, set to 0.

struStreamIDParam

Condition structure of searching for video files with stream ID information, see details below:

struIDInfo

Stream ID information, and it is 72 bytes, refer to for details.

dwCmdType

The external triggering type, which is used for NVR accessing to the cloud storage.

byBackupVolumeNum

The storage volume No., which is available for CVR.

byRes

Reserved, set to 0.

struTrialParam

Condition structure of searching for video files based on trial information, see details below:

byCaseNo

Case No.

byCaseName

Case name.

byLitigant1

Litigant No.1.

byLitigant2

Litigant No.2

byChiefJudge

Chief judge

byCaseType

Case type: 0-all, 1criminal case, 2-civil case.

byRes

Reserved, set to 0

struPosAlarm

Condition structure of searching for video files with POS information, see details below:

wChannel

Channel No., "0xffff"-the following are invalid. For example, wChannel[0]==1&&wChannel[1]==2 indicates the video with POS information of channel No.1 and channel No.2.

byAllChan

Whether to search in all channels: 0-no (**wChannel** is valid), 1-all channels (**wChannel** is invalid).

byCaseSensitive

Whether to enable case sensitive: 0-no, 1-yes.

byCombinateMode

The keyword combination mode: 0-or, 1-and.

byRes1

Reserved, set to 0.

sKeyWord

Search by keyword.

byRes

Reserved, set to 0.

struACSAlarm

Condition structure of searching for video files based on access control event, see details below:

dwMajor

Alarm/event major type, 0-all.

dwMinor

Alarm/event minor type, 0-all.

byCardNo

Card No.

byName

Name

byMACAddr

Physical MAC address.

byRes

Reserved, set to 0.

struIOTAlarm

Condition structure of searching for video files based on IoT device event, see details below:

wDeviceType

Device type: 0-Hikvision access controller, 1-Hikvision video intercom, 2-Hikvision security control panel, 3-GJD security control panel, 4-Luminite security control panel, 5-OPTEX security control panel, 6-Detector

wEventType

Event searching sub type, it varies according to main type, 0xffff-all

wChannel

Channel No.

byAllChan

Search all channel: 0-no, the **wChannel** is valid, 1-all channel, the **wChannel** is invalid

byCaseSensitive

Case sensitivity or not: 1-no, 1-yes

byCombinateMode

Key word combination mode: 0-or, 1-and

bySearchType

Search method: 1-according to video source, now the channel is video channel.

sKeyWord

Key word

wZoneNo

Zone No., it is valid only when the **wDeviceType** values "2-Hikvision security control panel" and the sub type **wEventType** values "1"

byRes

Reserved, set to 0.

Related API

NET_DVR_FindFileByEvent_V50

A.2.65 NET_DVR_SEARCH_EVENT_RET_V50

Information structure of video files searched by event.

Structure Definition

```
struct{
  WORD              wMajorType;
  WORD              wMinorType;
  NET_DVR_TIME_SEARCH    struStartTime;
```

```
NET_DVR_TIME_SEARCH    struEndTime;
NET_DVR_ADDRESS        struAddr;
WORD                   wChan[MAX_CHANNUM_V40/*512*/];
BYTE                   byRes[256];
union{
 BYTE                  byLen[800];
 struct{
  DWORD                dwAlarmInNo;
  BYTE                 byRes[796];
 }struAlarmRet;
 struct{
  DWORD                dwMotDetNo;
  BYTE                 byRes[796];
 }struMotionRet;
 struct{
  DWORD                dwChanNo;
  BYTE                 byRuleID;
  BYTE                 byRes1[3];
  BYTE                 byRuleName[NAME_LEN/*32*/];
  NET_VCA_EVENT_UNION  uEvent;
  BYTE                 byRes[668];
 }struVcaRet;
 struct{
  BYTE                 byRoomIndex;
  BYTE                 byDriveIndex;
  BYTE                 byRes1[6];
  DWORD                dwSegmentNo;
  WORD                 wSegmetSize;
  WORD                 wSegmentState;
  BYTE                 byRes2[784];
 }struInquestRet;
 struct{
  DWORD                dwRecordType;
  DWORD                dwRecordLength;
  BYTE                 byLockFlag;
  BYTE                 byDrawFrameType;
  BYTE                 byRes1[2];
  BYTE                 byFileName[NAME_LEN/*32*/];
  DWORD                dwFileIndex;
  BYTE                 byRes[752];
 }struStreamIDRet;
 struct{
  DWORD                dwChanNo;
  BYTE                 byRes[796];
 }struPosRet;
 struct{
  BYTE                 byRoomIndex;
  BYTE                 byDriveIndex;
  WORD                 wSegmetSize;
  DWORD                dwSegmentNo;
  BYTE                 bySegmentState;
  BYTE                 byCaseType;
```

```
    BYTE            byRes[2];
    BYTE            byCaseNo[CASE_NO_RET_LEN/*52*/];
    BYTE            byCaseName[CASE_NAME_RET_LEN/*64*/];
    BYTE            byLitigant1[LITIGANT_RET_LEN/*24*/];
    BYTE            byLitigant2[LITIGANT_RET_LEN/*24*/];
    BYTE            byChiefJudge[CHIEF_JUDGE_RET_LEN/*24*/];
    BYTE            byRes1[600];
   }struTrialRet;
  }uSeniorRet;
}NET_DVR_SEARCH_EVENT_RET_V50,*LPNET_DVR_SEARCH_EVENT_RET_V50;
```

Members

wMajorType

Major types, see details below:

```
enum _MAIN_EVENT_TYPE_{
 EVENT_MOT_DET      = 0,
 EVENT_ALARM_IN     = 1,
 EVENT_VCA_BEHAVIOR  = 2,
 EVENT_INQUEST      = 3,
 EVENT_VCA_DETECTION  = 4,
 EVENT_TRIAL_CASK    = 6,
 EVENT_STREAM_INFO   = 100,
}MAIN_EVENT_TYPE
```

EVENT_MOT_DET

Motion detection

EVENT_ALARM_IN

Alarm input

EVENT_VCA_BEHAVIOR

Behavior analysis

EVENT_INQUEST

Inquest event (not support)

EVENT_VCA_DETECTION

VCA detection

EVENT_POS

POS information

EVENT_TRIAL_CASK

Trial case information (not support)

EVENT_ACS_CASK

Access control event

EVENT_STREAM_INFO

Stream ID information

wMinorType

Minor alarm/event types, which vary with the major types. 0xffff-all types. For motion detection, alarm input, and POS recording, there is no minor type, so this parameter is set to "oxffff"; for access control event, refer to the integration manual of access control applications; for the major types of behavior analysis, VCA detection, and stream ID information, the corresponding minor types are shown below:

Table A-7 EVENT_VCA_BEHAVIOR

Minor Type	Value	Description
EVENT_TRAVERSE_PLANE	0	Line crossing
EVENT_ENTER_AREA	1	Entering the area,support regional rule
EVENT_EXIT_AREA	2	Leaving the area, support regional rule
EVENT_INTRUSION	3	Perimeter intrusion, support regional rule
EVENT_LOITER	4	Loitering, support regional rule
EVENT_LEFT_TAKE	5	Droping/picking up, support regional rule
EVENT_PARKING	6	Parking, support regional rule
EVENT_RUN	7	Running, support regional rule
EVENT_HIGH_DENSITY	8	People density in the area, support regional rule
EVENT_STICK_UP	9	Sticking a note, support regional rule
EVENT_INSTALL_SCANNER	10	Installing card reader, support regional rule
EVENT_OPERATE_OVER_TIME	11	Operation timeout
EVENT_FACE_DETECT	12	Abnormal face detection
EVENT_LEFT	13	Unattended baggage
EVENT_TAKE	14	Object removal
EVENT_LEAVE_POSITION	15	Absence event

Minor Type	Value	Description
EVENT_TRAIL_INFO	16	Tailing
EVENT_FALL_DOWN_INFO	19	Falling down
EVENT_OBJECT_PASTE	20	Sticking script area
EVENT_FACE_CAPTURE_INFO	21	Normal face
EVENT_MULTI_FACES_INFO	22	Multiple faces
EVENT_AUDIO_ABNORMAL_INFO	23	Sudden change of sound intensity
EVENT_SUNGLASSES_FACE_INFO	25	Face with sunglasses
EVENT_CALLING_FACE_INFO	26	Person is making call

Table A-8 EVENT_VCA_DETECTION

Minor Type	Value	Description
EVENT_VCA_TRAVERSE_PLANE	1	Line crossing detection
EVENT_FIELD_DETECTION	2	Intrusion detection
EVENT_AUDIO_INPUT_ALARM	3	Audio loss detection
EVENT_SOUND_INTENSITY_ALARM	4	Sudden increase of sound increase/decrease detection
EVENT_FACE_DETECTION	5	Face detection
EVENT_VIRTUAL_FOCUS_ALARM	6	Defocus detection
EVENT_SCENE_CHANGE_ALARM	7	Scene change detection
EVENT_PIR_ALARM	8	PIR alarm
EVENT_ENTER_REGION	9	Region entrance
EVENT_EXIT_REGION	10	Region exiting
EVENT_LOITERING	11	Loitering
EVENT_GROUPDETECTION	12	People gathering in the area, support regional rule
EVENT_RAPIDMOVE	13	Fast moving

Minor Type	Value	Description
EVENT_PARK	14	Parking
EVENT_UNATTENDED_BAGGAGE	15	Unattended baggage
EVENT_ATTENDEDBAGGAGE	16	Object removal
EVENT_VEHICLE_DETECTION	17	Vehicle detection

Table A-9 EVENT_STREAM_INFO

Minor Type	Value	Description
EVENT_STREAM_ID	0	Stream ID
EVENT_TIMING	1	Timing record
EVENT_MOTION_DETECT	2	Motion detection
EVENT_ALARM	3	Alarm record
EVENT_ALARM_OR_MOTION_DETECT	4	Alarm or motion detection
EVENT_ALARM_AND_MOTION_DETECT	5	Alarm and motion detection
EVENT_COMMAND_TRIGGER	6	Command triggering
EVENT_MANNUAL	7	Manual record
EVENT_BACKUP_VOLUME	8	Storage volume record
STREAM_EVENT_SEMAPHORE	9	Sensor alarm
STREAM_EVENT_HIDE	10	Video tempering
STREAM_EVENT_INVERSE	11	Driving in opposite direction
STREAM_EVENT_VIDEO_LOST	12	Video loss
STREAM_EVENT_WIRELESS_ALARM	13	Wirless alarm
STREAM_EVENT_PIR_ALARM	14	PIR alarm
STREAM_EVENT_CALLHELP_ALARM	15	Call for help alarm
STREAM_EVENT_FACESNAP	16	Face capture
STREAM_EVENT_FACE_DETECTION	17	Face detection

Minor Type	Value	Description
STREAM_EVENT_ITS_PLATE	18	Vehicle detection
STREAM_EVENT_PDC	19	People counting
STREAM_EVENT_ SCENECHANGE_DETECTION	20	Scene change
STREAM_EVENT_DEFOCUS_ DETECTION	21	Defocus detection
STREAM_EVENT_ AUDIOEXCEPTION	22	Audio exception
VCA_EVENT_TRAVERSE_ PLANE	23	Line crossing
VCA_EVENT_INTRUSION	24	Intrusion
VCA_EVENT_ENTER_AREA	25	Region entrance
VCA_EVENT_EXIT_AREA	26	Region exiting
VCA_EVENT_LOITER	27	Loitering detection
VCA_EVENT_HIGH_DENSITY	28	People gathering
VCA_EVENT_RUN	29	Fast moving
VCA_EVENT_PARKING	30	Illegal parking
VCA_EVENT_LEFT	31	Unattended baggage
VCA_EVENT_TAKE	32	Objet removal

struStartTime

Start time of search, refer to the structure for details.

struEndTime

Stop time of search, refer to the structure for details.

struAddr

Address information of video segment, which for cluster playback, refer to for details.

wChan

Alarm triggered or event occurred channel No. 0xffff-the followings are invalid.

byRes

Reserved, set to 0.

uSeniorRet

Result union of searching for video files based on event/alarm, see details below:

byLen

Union size, it is 800 bytes.

struAlarmRet

Result structure of searching for video files based on alarm input alarm, see details below:

dwAlarmInNo

Alarm input No..

byRes

Reserved.

struMotionRet

Result structure of searching for video files based on motion detection, see details below:

dwMotDetNo

Motion detection channel No.

byRes

Reserved, set to 0.

struVcaRet

Result structure of searching for video files based on behavior analysis alarm, see details below:

dwChanNo

Behavior analysis channel No.

byRuleID

Rule ID, 0xff-all rules

byRes1

Reserved, set to 0.

byRuleName

Rule name.

uEvent

Behavior analysis parameters, which depends on the parameter **wMinorType**, refer to the integration manual of behavior analysis applications for details.

byRes

Reserved, set to 0.

struInquestRet

Result structure of searching for video files based on inquest event, see details below:

byRoomIndex

Inquest room No., starts from 1.

byDriveIndex

Recorder No., starts from 1.

byRes1

Reserved, set to 0.

dwSegmentNo

Video segment No. of this inquest, starts from 1.

wSegmetSize

Video segment size, unit: MB

wSegmentState

Recording status: 0-normal, 1-exception, 2-unrecorded

byRes2

Reserved, set to 0.

struStreamIDRet

Result structure of searching for video files with stream ID information, see details below:

dwRecordType

Recording types: 0-scheduled recording, 1-based on motion detection, 2-based on alarm input alarm, 3-based on alarm input alarm or motion detection, 4-based on alarm input alarm and motion detection, 5-based on command, 6-manual recording, 7-based on vibration alarm, 8-based on environment alarm, 9-based on VCA alarm (including driving in the opposite direction, line crossing, unattended baggage, object removal and so on), 10-based on video tampering alarm, 13-based on event (motion detection, PIR, wireless panic alarm, and so on), 24-by video montage.

dwRecordLength

Video file size.

byLockFlag

Whether to lock: 0-no, 1-yes.

byDrawFrameType

Whether to extract the frame when recording: 0-no, 1-yes.

byRes1

Reserved, set to 0.

byFileName

File name.

dwFileIndex

File index No. in storage volume.

byRes

Reserved, set to 0.

struPosRet

Result structure of searching for video files with POS information, see details below:

dwChanNo

POS event channel No.

byRes

Reserved, set to 0.

struTrialRet

Result structure of searching for video files with trial information, see details below:

byRoomIndex

Inquest room No., starts from 1 .

byDriveIndex

Recorder No., starts from 1.

wSegmetSize

Video segment size, unit: MB.

dwSegmentNo

Video segment No. in this inquest, starts from 1.

bySegmentState

Recording status: 0-normal, 1-exception, 2-unrecorded.

byCaseType

Case type: 0-all, 1-criminal case, 2-civil case

byRes

Reserved, set to 0.

byCaseNo

Case No.

byCaseName

Case name.

byLitigant1

Litigant No.1.

byLitigant2

Litigant No.2.

byChiefJudge

Chief judge

byRes1

Reserved, set to 0.

Related API

NET_DVR_FindNextEvent_V50

A.2.66 NET_DVR_SIMXML_LOGIN

Structure about Complement Fields by Stimulation Capability

Member	Data Type	Description
byLoginWithSimXml	BYTE	Whether to complement fields by stimulation capability: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 127 bytes.

A.2.67 NET_DVR_SPECIAL_FINDINFO_UNION

Specific search condition union.

Union Definition

```
union{
  BYTE              byLenth[8];
  NET_DVR_ATMFINDINFO    struATMFindInfo;
}NET_DVR_SPECIAL_FINDINFO_UNION,*LPNET_DVR_SPECIAL_FINDINFO_UNION;
```

Members

byLenth

Union size, it is 8 bytes.

struATMFindInfo

Condition for searching file with ATM information, see details in the structure of ***NET_DVR_ATMFINDINFO*** .

A.2.68 NET_DVR_STD_CONFIG

Structure About Configuring Input and Output Parameters

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters, e.g., channel No., it can be set to "NULL".
dwCondSize	DWORD	Size of buffer for storing condition parameters
lpInBuffer	LPVOID	Input parameters (a structure)
dwInSize	DWORD	Size of buffer for storing input parameters
lpOutBuffer	LPVOID	Output parameters (a structure)
dwOutSize	DWORD	Size of buffer for storing output parameters
lpStatusBuffer	LPVOID	Returned status parameters in XML format, it can be set to NULL.
dwStatusSize	DWORD	Size of buffer for storing status parameters
lpXmlBuffer	LPVOID	Request or response message in XML format, it is valid when **byDataType** is 1.
dwXmlSize	DWORD	Size of memory pointed by **lpXmlBuffer**.
byDataType	BYTE	Input or output parameter type: 0-valid when the input or output parameters is a structure; 1-valid when the input or output parameters is a XML message.
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 32 bytes.

A.2.69 NET_DVR_STREAM_INFO

Stream information structure.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byID[STREAM_ID_LEN/*32*/];
  DWORD    dwChannel;
  BYTE     byRes[32];
}NET_DVR_STREAM_INFO,*LPNET_DVR_STREAM_INFO;
```

Members

dwSize

Structure size.

byID

Stream ID, which consists of letters, digits, and dashes, 0-invalid.

dwChannel

Linked device channel. When it is 0xffffffff, if setting the stream source, this parameter indicates that no device channel is linked; if setting configuration condition, this parameter is invalid.

byRes

Reserved, set to 0.

Remarks

- If the device does not support marking stream ID, e.g., DVR, the parameter **byID** should be set to 0.
- For transcoder, when setting the stream source, only one of **byID** and **dwChannel** can be valid; when transcoding, both the **byID** and **dwChannel** can be invalid, the transcoding channel or stream ID is automatically allocated by device.
- For other devices (e.g., CVR), when this structure is inputted as configuration condition, if both the **byID** and **dwChannel** are invalid, error code (17) will be returned, if they are valid, but mismatched, error may also be returned, so only setting one of these two parameters is suggested.

A.2.70 NET_DVR_STREAM_MEDIA_SERVER

Structure about stream media server parameters.

Structure Definition

```
struct{
  BYTE    byValid;
  BYTE    byRes1[3];
  BYTE    byAddress[MAX_DOMAIN_NAME/*64*/];
  WORD    wDevPort;
  BYTE    byTransmitType;
  BYTE    byRes2[5];
}NET_DVR_STREAM_MEDIA_SERVER,*LPNET_DVR_STREAM_MEDIA_SERVER;
```

Members

byValid

Enable/disable stream media server to get stream: 0-disable, 1-enable.

byRes1

Reserved, set to 0.

byAddress

IP address or domain name of stream media server

wDevPort

Port number of stream media server

byTransmitType

Transfer protocol type: 0-TCP, 1-UDP

byRes2

Reserved, set to 0.

See Also

NET_DVR_DEC_STREAM_DEV_EX
NET_DVR_DEC_DDNS_DEV

A.2.71 NET_DVR_TIME

Time Parameter Structure

Member	Data Type	Description
dwYear	DWORD	Year
dwMonth	DWORD	Month
dwDay	DWORD	Day
dwHour	DWORD	Hour
dwMinute	DWORD	Minute
dwSecond	DWORD	Second

A.2.72 NET_DVR_TIME_SEARCH

OSD time information structure.

Structure Definition

```
struct{
  WORD    wYear;
  BYTE    byMonth;
  BYTE    byDay;
  BYTE    byHour;
  BYTE    byMinute;
  BYTE    bySecond;
  char    cTimeDifferenceH;
  char    cTimeDifferenceM;
```

```
  BYTE    byRes[3];
}NET_DVR_TIME_SEARCH, *LPNET_DVR_TIME_SEARCH;
```

Members

wYear

Year

byMonth

Month

byDay

Day

byHour

Hour

byMinute

Minute

bySecond

Second

cTimeDifferenceH

Time difference (hour) between OSD time and UTC, which ranges from -12 to +14.

cTimeDifferenceM

Time difference (minute) between OSD time and UTC, the values can be -30, 0, 30, and 45.

byRes

Reserved.

A.2.73 NET_DVR_TIME_SEARCH_COND

Time data structure of search condition.

Structure Definition

```
struct{
  WORD    wYear;
  BYTE    byMonth;
  BYTE    byDay;
  BYTE    byHour;
  BYTE    byMinute;
  BYTE    bySecond;
  BYTE    byLocalOrUTC;
  BYTE    byRes[4];
}NET_DVR_TIME_SEARCH_COND, *LPNET_DVR_TIME_SEARCH_COND;
```

Members

wYear

Year, device OSD time.

byMonth

Month, device OSD time.

byDay

Day, device OSD time.

byHour

Hour, device OSD time.

byMinute

Minute, device OSD time.

bySecond

Second, device OSD time.

byLocalOrUTC

Time type: 0-device local time (device's OSD time), 1-UTC time.

byRes

Reserved, set to 0.

See Also

NET_DVR_FILECOND_V50

A.2.74 NET_DVR_USER_LOGIN_INFO

Structure About Login Parameters

Member	Data Type	Description
sDeviceAddress	char	Device IP address, or domain name.
byUseTransport	BYTE	Enable capability transmission or not: 0-no (default), 1-yes.
wPort	WORD	Device port number, e.g., 8000 (when login by private protocol), 80 (when login by text protocol).
sUserName	char	User name for logging in to device.
sPassword	char	Login password.

Member	Data Type	Description
cbLoginResult	***fLoginResultCallBack***	Callback function used to return login status, it is valid only when **bUseAsynLogin** is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is unknown, and this mode does not support asynchronous login).
byHttps	BYTE	Whether to enable TLS for login (by private protocol or by text protocol): 0-no, 1-yes, 2-self-adaptive (which is usually used when the protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not available), 2-unidirectional verification (it is valid when **byLoginMode** is 0 and **byHttps** is 1); when **byVerifyMode** is 0, CA certificate is not required, when **byVerifyMode** is 2, you should call NET_DVR_SetSDKLocalCfg to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

A.2.75 NET_DVR_VIDEO_WALL_INFO

Video wall information structure.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwWindowNo;
  DWORD    dwSceneNo;
  BYTE     byRes[20];
}NET_DVR_VIDEO_WALL_INFO,*LPNET_DVR_VIDEO_WALL_INFO;
```

Members

dwSize

Structure size

dwWindowNo

Window No.=1-byte video wall No.+reserved 1-byte+2-byte window No., e.g., 1<<24

dwSceneNo

Scene No.

byRes

Reserved, set to 0

See Also

NET_DVR_SCENE_CONTROL_INFO

A.2.76 NET_DVR_VIDEOEFFECT

Video parameter structure

Structure Definition

```
struct{
  BYTE    byBrightnessLevel;
  BYTE    byContrastLevel;
  BYTE    bySharpnessLevel;
  BYTE    bySaturationLevel;
  BYTE    byHueLevel;
  BYTE    byEnableFunc;
  BYTE    byLightInhibitLevel;
  BYTE    byGrayLevel;
}NET_DVR_VIDEOEFFECT, *LPNET_DVR_VIDEOEFFECT;
```

Members

byBrightnessLevel

Brightness, range: [0,100]

byContrastLevel

Contrast, range: [0,100]

bySharpnessLevel

Sharpness, range: [0,100]

bySaturationLevel

Saturation, range: [0,100]

byHueLevel

Hue, range: [0,100], reserved.

byEnableFunc

Bit: bit0-enable/disable Smart IR, bit1-enable/disable low illumination, bit2-enable/disable HLC, bit3-sharpness adjusting mode (manual or auto). Value: 0-disable/auto, 1-enable/manual. E.g., byEnableFunc&0x2==1, it indicates that the low illumination function is enabled.

byLightInhibitLevel

HLC level, range: [1,3]

byGrayLevel

Gray scale, 0-[0,255], 1-[16,235]

A.2.77 NET_DVR_VIDEOWALLDISPLAYPOSITION

Configuration parameter structure of jointed screen position on video wall.

Structure Definition

```
struct{
  DWORD            dwSize;
  BYTE             byEnbale;
  BYTE             byRes1[3];
  DWORD            dwVideoWallNo;
  DWORD            dwDisplayNo;

                        NET_DVR_RECTCFG_EX
                         struScreenBaseInfo;
  BYTE             byRes2[64];
}NET_DVR_VIDEOWALLDISPLAYPOSITION, *LPNET_DVR_VIDEOWALLDISPLAYPOSITION;
```

Members

dwSize

Structure size

byEnbale

Status: 0-disabled, 1-enabled

byRes1

Reserved, set to 0

dwVideoWallNo

Video wall No.=1-byte video wall No.+reserved 3-byte, e.g., 1<<24 | 0

dwDisplayNo

Jointed screeen No.=1-byte device No.+reserved 1-byte+2-byte jointed screen No. (starts from 1), e.g., 1<<24 | 0<<16 | 1. It is valid when getting all in batch.

struScreenBaseInfo

Jointed screen position coordinates, it should be the integral multiples of reference coordinates (which is obtained via video wall capability set). By default, the width and height are same as the reference values without configured. See the structure ***NET_DVR_RECTCFG_EX*** for details.

byRes2

Reserved, set to 0

A.2.78 NET_DVR_VIDEOWALLWINDOWPOSITION

Information structure of window position on video wall.

Structure Definition

```
struct{
  DWORD          dwSize;
  BYTE           byEnable;
  BYTE           byRes1[7];
  DWORD          dwWindowNo;
  DWORD          dwLayerIndex;

                 NET_DVR_RECTCFG_EX
                  struRect;

                 NET_DVR_RECTCFG_EX
                  struResolution;
  DWORD          dwXCoordinate;
  DWORD          dwYCoordinate;
  BYTE           byRes2[36];
}NET_DVR_VIDEOWALLWINDOWPOSITION,*LPNET_DVR_VIDEOWALLWINDOWPOSITION;
```

Members

dwSize

Structure size

byEnbale

Status: 0-disabled, 1-enabled

byRes1

Reserved, set to 0

dwWindowNo

Window No., it is valid when getting all in batch.

dwLayerIndex

Layer No., it can be obtained only, and it is linked to the device.

struRect

Target window coordinates (relative to the video wall), see the structure ***NET_DVR_RECTCFG_EX*** for details.

struResolution

Target window resolution, it is valid when getting or setting by resolution.

dwXCoordinate

X-coordinate of top left corner of LED area, it is valid when getting or setting by resolution.

dwYCoordinate

Y-coordinate of top left corner of LED area, it is valid when getting or setting by resolution.

byRes2

Reserved, set to 0

A.2.79 NET_DVR_WALL_WIN_STATUS

Window status information structure of video wall.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     byDecodeStatus;
  BYTE     byStreamType;
  BYTE     byPacketType;
  BYTE     byFpsDecV;
  BYTE     byFpsDecA;
  BYTE     byRes1[7];
  DWORD    dwDecodedV;
  DWORD    dwDecodedA;
  WORD     wImgW;
  WORD     wImgH;
  BYTE     byRes2[32];
}NET_DVR_WALL_WIN_STATUS, *LPNET_DVR_WALL_WIN_STATUS;
```

Members

dwSize

Structure size

byDecodeStatus

Current decoding status: 0-disabled, 1-enabled

byStreamType

Encoding format: 0-NET_DVR_ENCODER_UNKOWN (unknown); 1-NET_DVR_ENCODER_H264 (private 264); 2-NET_DVR_ENCODER_S264 (standard H264); 3-NET_DVR_ENCODER_MPEG4 (MPEG4); NET_DVR_ORIGINALSTREAM (raw stream); 5-NET_DVR_PICTURE (VCA alarm picture); 6-NET_DVR_ENCODER_MJPEG (MJPEG); 7-NET_DVR_ECONDER_MPEG2 (MPEG2)

byPacketType

Packet format: 0-NET_DVR_STREAM_TYPE_UNKOWN (unknown); 1-NET_DVR_STREAM_TYPE_PRIVT (custom); 7-NET_DVR_STREAM_TYPE_TS (TS); 8-NET_DVR_STREAM_TYPE_PS (PS); 9-NET_DVR_STREAM_TYPE_RTP (RTP); 10-NET_DVR_STREAM_TYPE_ORIGIN (unpackaged)

byFpsDecV

Video decoding frame rate

byFpsDecA

Audio decoding frame rate

byRes1

Reserved, set to 0

dwDecodedV

Decoded video frames

dwDecodedA

Decoded audio frames

wImgW

Current image width

wImgH

Current image height

byRes

Reserved, set to 0

A.2.80 NET_DVR_WALLOUTPUTPARAM

Video output parameter structure of video wall

Structure Definition

```
struct{
  DWORD             dwSize;
```

```
  DWORD              dwResolution;
  NET_DVR_VIDEOEFFECT    struRes;
  BYTE               byVideoFormat;
  BYTE               byDisplayMode;
  BYTE               byBackgroundColor;
  BYTE               byUseEDIDResolution;
  WORD               wLEDWidth;
  WORD               wLEDHeight;
  NET_DVR_RGB_COLOR      struBackColor;
  BYTE               byRes2[52];
}NET_DVR_WALLOUTPUTPARAM, *LPNET_DVR_WALLOUTPUTPARAM;
```

Members

dwSize

Structure size

dwResolution

Resolution

struRes

Video parameters, see the structure ***NET_DVR_VIDEOEFFECT*** for details.

byVideoFormat

Video standard, see the structure ***VIDEO_STANDARD*** for details.

byDisplayMode

Output connection mode: 1-BNC, 2-VGA, 3-HDMI®, 4-DVI, 5-SDI, 6-FIBER, 7-RGB, 8-YprPb, 9-VGA/HDMI®/DVI, 10-3GSDI, 11-VGA/DVI, 12-HDBaseT, 13-HDTV, 14-TVI, 0xff-invalid

byBackgroundColor

Background color: 0-invalid, not support; 1-read; 2-green; 3-blue; 4-yellow; 5-purple; 6-cyan; 7-black; 8-white, 0xff-custom.

byUseEDIDResolution

EDID resolution: 0-diable, 1-enable

wLEDWidth

Width of LED output resolution

wLEDHeight

Height of LED output resolution

struBackColor

Background color, see details in the structure ***NET_DVR_RGB_COLOR*** ; this member is valid when the value of byBackgroundColor is 0xff.

byRes2

Reserved, set to 0

A.2.81 NET_DVR_WALLSCENECFG

Scene configuration parameter structure of video wall.

Structure Definition

```
struct{
  DWORD    dwSize;
  BYTE     sSceneName[NAME_LEN/*32*/];
  BYTE     byRes[80];
}NET_DVR_WALLSCENECFG, *LPNET_DVR_WALLSCENECFG;
```

Members

dwSize

Structure size

sSceneName

Scene name

byRes

Reserved, set to 0.

A.2.82 NET_DVR_WALLWIN_INFO

Window information structure of video wall.

Structure Definition

```
struct{
  DWORD    dwSize;
  DWORD    dwWinNum;
  DWORD    dwSubWinNum;
  BYTE     byRes[16];
}NET_DVR_WALLWIN_INFO, *LPNET_DVR_WALLWIN_INFO;
```

Members

dwSize

Structure size

dwWinNum

Window No.

dwSubWinNum

Sub window No.

byRes

Reserved, set to 0.

A.2.83 NET_DVR_WALLWINPARAM

Window parameter structure of video wall.

Structure Definition

```
struct{
  DWORD   dwSize;
  BYTE    byTransparency;
  BYTE    byWinMode;
  BYTE    byEnableSpartan;
  BYTE    byDecResource;
  BYTE    byWndShowMode;
  BYTE    byEnabledFeature;
  BYTE    byFeatureMode;
  BYTE    byRes1;
  DWORD   dwAmplifyingSubWndNo;
  BYTE    byWndTopKeep;
  BYTE    byWndOpenKeep;
  BYTE    byRes[22];
}NET_DVR _WALLWINPARAM, *LPNET_DVR _WALLWINPARAM;
```

Members

dwSize

Structure size

byTransparency

Transparency: 0-disable, 1-enable

byWinMode

Window division mode, which can be obtained via the video wall capability set.

byEnableSpartan

Fluent video: 0-disable, 1-enable

byDecResource

Allocate decoding resources: 1-D1, 2-720P, 3-1080P, 4-300W, 5-500W

byWndShowMode

Window display mode: 0-reserved, 1-sub window mode, 2-sub window in full screen mode

byEnabledFeature

Scene close-up: 0-disable, 1-enable

byFeatureMode

Close-up mode, it is valid only when the scene close-up is enabled: 0-invalid, 1-"1+5" mode

byRes1

Reserved.

dwAmplifyingSubWndNo

Sub window No.=1-byte video wall No.+1-byte sub window No.+2-byte window No., it is valid only when **byWndShowMode** is 2.

byWndTopKeep

Window stays on top: 0-not support, 1-support

byWndOpenKeep

Window remains open: 0-not support, 1-support

byRes

Reserved

Remarks

- The scene close-up can be enabled only when the **byWinMode** is in 1×1 window division mode.
- The window with scene close-up enabled is called close-up window, and the larger sub window is called main sub window (the No. is 1), the other sub windows (the No. starts from 2) are called close-up sub window.
- After enabling scene close-up, the image of the original window will be displayed in the main sub window, and the close-up images will be displayed in the close-up sub window.

A.2.84 NET_DVR_XML_CONFIG_INPUT

Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
lpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in ***NET_DVR_MIME_UNIT*** .
dwInBufferSize	DWORD	Input buffer size.
dwRecvTimeOut	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
byForceEncrpt	BYTE	Whether to enable force encryption (the messages will be encrypted by AES algorithm for transmission): 0-no, 1-yes.

Member	Data Type	Description
byNumOfMultiPart	BYTE	Number of message segments: 0-invalid; other values-number of message segments, which is transmitted by the parameter **lpInBuffer** in the structure ***NET_DVR_MIME_UNIT*** .
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.2.85 NET_DVR_XML_CONFIG_OUTPUT

Output Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.
lpStatusBuffer	void*	Response status (ResponseStatus message). This parameter will not be assigned if performing GET operation succeeded, and you can also set it to "NULL" if not required.
dwStatusSize	DWORD	Size of response status buffer.
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.3 Enumeration

A.3.1 NET_SDK_LOCAL_CFG_TYPE

Enumerate the local configuration types of device network SDK.

Enumeration Definition

```
enum{
  NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND      =0,
  NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND      =1,
  NET_SDK_LOCAL_CFG_TYPE_MEM_POOL           =2,
  NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT =3,
  NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE      =4,
  NET_SDK_LOCAL_CFG_TYPE_TALK_MODE          =5,
  NET_SDK_LOCAL_CFG_TYPE_PROTECT_KEY        =6
  NET_SDK_LOCAL_CFG_TYPE_CFG_VERSION        =7
  NET_SDK_LOCAL_CFG_TYPE_RTSP_PARAMS        =8
  NET_SDK_LOCAL_CFG_TYPE_SIMXML_LOGIN       =9
  NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV          =10,
  NET_SDK_LOCAL_CFG_TYPE_SECURITY           =11
  NET_SDK_LOCAL_CFG_TYPE_EZVIZLIB_PATH      =12
  NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE        =13,
  NET_SDK_LOCAL_CFG_TYPE_PROXYS             =14
  NET_DVR_LOCAL_CFG_TYPE_LOG                =15,
  NET_DVR_LOCAL_CFG_TYPE_STREAM_CALLBACK    =16
  NET_DVR_LOCAL_CFG_TYPE_GENERAL            =17,
  NET_DVR_LOCAL_CFG_TYPE_PTZ                =18,
  NET_DVR_LOCAL_CFG_MESSAGE_CALLBACK_V51    =19
  NET_SDK_LOCAL_CFG_CERTIFICATION           =20,
  NET_SDK_LOCAL_CFG_PORT_MULTIPLEX          =21,
  NET_SDK_LOCAL_CFG_ASYNC                   =22
}NET_SDK_LOCAL_CFG_TYPE
```

Members

NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND

Local binding configuration of TCP port, see details in ***NET_DVR_LOCAL_TCP_PORT_BIND_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND

Binding configuration of local UDP port, see details in ***NET_DVR_LOCAL_UDP_PORT_BIND_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_MEM_POOL

Local configuration of storage pool, see details in ***NET_DVR_LOCAL_MEM_POOL_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT

Timeout configuration by module, see details in ***NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE

Capability analysis library configuration, see details in ***NET_DVR_LOCAL_ABILITY_PARSE_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_TALK_MODE

Two-way audio configuration, see details in ***NET_DVR_LOCAL_TALK_MODE_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_PROTECT_KEY

Key configuration, see details in ***NET_DVR_LOCAL_PROTECT_KEY_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_CFG_VERSION

Check the device compatibility when setting parameters.

NET_SDK_LOCAL_CFG_TYPE_RTSP_PARAMS

RTSP parameters, see details in ***NET_DVR_RTSP_PARAMS_CFG*** .

NET_SDK_LOCAL_CFG_TYPE_SIMXML_LOGIN

Parameters of using stimulation capability to complement fields, see details in ***NET_DVR_SIMXML_LOGIN*** .

NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV

Heartbeat time interval, see details in ***NET_DVR_LOCAL_CHECK_DEV*** .

NET_SDK_LOCAL_CFG_TYPE_SECURITY

SDK security parameters.

NET_SDK_LOCAL_CFG_TYPE_EZVIZLIB_PATH

Communication library address of EZVIZ cloud.

NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE

Encoding format conversion configuration, see details in ***NET_DVR_LOCAL_BYTE_ENCODE_CONVERT*** .

NET_SDK_LOCAL_CFG_TYPE_PROXYS

Proxy types.

NET_DVR_LOCAL_CFG_TYPE_LOG

Log parameters, see details in ***NET_DVR_LOCAL_LOG_CFG*** .

NET_DVR_LOCAL_CFG_TYPE_STREAM_CALLBACK

Stream callback parameters, see details in ***NET_DVR_LOCAL_STREAM_CALLBACK_CFG*** .

NET_DVR_LOCAL_CFG_TYPE_GENERAL

General parameters, see details in ***NET_DVR_LOCAL_GENERAL_CFG*** .

NET_DVR_LOCAL_CFG_TYPE_PTZ

PTZ interaction parameters, see details in ***NET_DVR_LOCAL_CFG_TYPE_PTZ*** .

NET_DVR_LOCAL_CFG_MESSAGE_CALLBACK_V51

Local parameters of alarm callback, see details in ***NET_DVR_MESSAGE_CALLBACK_PARAM_V51*** .

NET_SDK_LOCAL_CFG_CERTIFICATION

Certificate parameters, see details in ***NET_DVR_LOCAL_CERTIFICATION*** .

NET_SDK_LOCAL_CFG_PORT_MULTIPLEX

Port multiplier parameters, see details in ***NET_DVR_LOCAL_PORT_MULTI_CFG*** .

NET_SDK_LOCAL_CFG_ASYNC

Asynchronous mode parameters, see details in ***NET_DVR_LOCAL_ASYNC_CFG*** .

A.3.2 VIDEO_STANDARD

Enumerate the video standards.

Enumeration Definition

```
enum{
  VS_NON  = 0,
  VS_NTSC = 1,
  VS_PAL  = 2
}VIDEO_STANDARD
```

Members

VS_NON

None

VS_NTSC

NTSC standard

VS_PAL

PAL standard

A.4 Request URIs

A.4.1 /ISAPI/DisplayDev/VideoWall/<ID>/windows/capabilities

Get video wall window capabilities.

Request URI Definition

Table A-10 GET /ISAPI/DisplayDev/VideoWall/<ID>/windows/capabilities

Method	GET
Description	Get video wall window capabilities.
Query	None.
Request	None.
Response	Succeeded: ***XML_WallWindowCap*** Failed: ***XML_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to the video wall ID.

A.4.2 /ISAPI/DisplayDev/VideoWall/<ID>/windows/subSteam/capabilities?format=json

Get the configuration capability of the stream type of the windows.

Request URI Definition

Table A-11 GET /ISAPI/DisplayDev/VideoWall/<ID>/windows/subSteam/capabilities?format=json

Method	GET
Description	Get the configuration capability of the stream type of the windows.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_MutiScreenSubStreamCap*** Failed: ***XML_ResponseStatus***

Remarks

The <**ID**> in the request URI refers to video wall ID.

A.4.3 /ISAPI/DisplayDev/VideoWall/<ID>/windows/subSteam?format=json

Operations about the configuration of the stream type for streaming when the number of windows exceeds the limit.

Request URI Definition

Table A-12 GET /ISAPI/DisplayDev/VideoWall/<ID>/windows/subSteam?format=json

Method	GET
Description	Get the configuration parameters of the stream type for streaming when the number of windows exceeds the limit.
Query	**format**: determine the format of request or response message.
Request	None.
Response	Succeeded: ***JSON_MutiScreenSubStream*** Failed: ***JSON_ResponseStatus***

A.5 Request and Response Messages

A.5.1 JSON_MutiScreenSubStream

MutiScreenSubStream message in JSON format

```
{
  "MutiScreenSubStream":{
    "enabled": ,
/*optional, boolean type, whether to enable the function*/
    "winConutLimit":"",
/*required, string type, threshold number of divided windows. When the number of divided windows exceeds the threshold, streams of low performance will be used*/
  }
}
```

A.5.2 JSON_MutiScreenSubStreamCap

JSON message about configuration capability of sub stream of the windows

```
{
  "MutiScreenSubStreamCap":{
    "enabled":{
/*optional, boolean*/
      "@opt":[true,false]
    },
    "winConutLimit":{
/*required, string,  threshold number of divided windows. When the number of divided windows exceeds the threshold, streams of low performance will be used*/
      "@opt":["1","4","6","8","9","16","32","64"]
    }
  }
}
```

A.5.3 JSON_ResponseStatus

JSON message about response status

```
{
  "requestURL":"",
/*optional, string, request URL*/
  "statusCode": ,
/*required, int, status code*/
  "statusString":"",
/*required, string, status description*/
  "subStatusCode":"",
```

```
/*required, string, sub status code*/
  "errorCode": ,
/*optional, int, error code, which corresponds to subStatusCode, this field is required when statusCode is not 1. The returned value is the transformed decimal number*/
  "errorMsg":"",
/*optional, string, error details, this field is required when statusCode is not 1*/
}
```

Note

See ***Response Codes of Text Protocol*** for details about the status codes, sub status codes, error codes, and error descriptions.

A.5.4 XML_ResponseStatus

XML message about response status

```
<ResponseStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <requestURL>
    <!--required, read-only, xs:string, request URL-->
  </requestURL>
  <statusCode>
    <!--required, read-only, xs:integer, status code: 0,1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content, 7-Reboot Required, 9-Additional Error-->
  </statusCode>
  <statusString>
    <!--required, read-only, xs:string, status description: OK, Device Busy, Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content, Reboot, Additional Error-->
  </statusString>
  <subStatusCode>
    <!--required, read-only, xs:string, describe the error reason in detail-->
  </subStatusCode>
</ResponseStatus>
```

Note

See ***Response Codes of Text Protocol*** for details about sub status codes and corresponding error codes.

A.5.5 XML_WallAbility

XML message about video wall capability

```
<?xml version="1.0" encoding="UTF-8"?>
<WallAbility><!--required, the sequence number of window -->
  <winNum min=""  max="" /><!--required, the sequence number of scene-->
  <senceNum min=""  max="" /><!--required, the count of supported screens-->
  <maxScreenNum><!--required, the sequence number of opt cascade integrated platform--></maxScreenNum>
  <platformNum min="" max="" /><!--required, the sequence number of opt basemap-->
```

```
 <basePicNum min=""  max="" /><!--required, the count of supported windows-->
 <windowMode opt="1,4,9,16,25,36"><!--required, the reference value of each screen's size--> </windowMode>
 <WindowBase>
  <windowBaseX>128</windowBaseX>
  <windowBaseY>128</windowBaseY>
 </WindowBase>
 <Transparency><!--required, transparency, not displayed if not supported-->
  <layerNum><!--required, supported layers of transparency--></layerNum>
  <range min="" max=""/>
  <!--required, transparency range-->
 </Transparency>
 <VoutResource>
  <videoOutNum><!--required, xs: inter, number of video outputs--></videoOutNum>
  <audioOutNum><!--required, xs: inter, number of audio outputs--></audioOutNum>
  <AllOutputConfigOnly><!--optional, whether all output ports should be configured together-->
   <enabled><!--required, xs:boolean,"true"--></enabled>
  </AllOutputConfigOnly>
  <SupportLEDOutputConfigAll><!--optional, whether LED output port parameters support whole wall configuration --
>
   <enabled>
    <!--required, xs:boolean,"true"-->
   </enabled>
  </SupportLEDOutputConfigAll>
  <supportLEDResolutionVoutType opt="hdmi,dvi,sdi"/><!--optional, type of output port supporting LED resolution,
supported by default if the node does not exist, xs:string-->
  <VoutResourceEntry><!--required, multiple same-level nodes are allowed-->
   <id></id>
   <voutType><!--required, xs:string,bnc,vga,hdmi,dvi,sdi,3gsdi,vga/dvi--></voutType>
   <voutNo min=""  max="" /><!--required, xs:inter, output No. of this output type-->
   <videoFormat opt = "NULL,NTSC,PAL"/><!--required, supported video standard-->
   <outputWindowMode opt="1,4,9,16,25,36"/><!--required, window resolution supported by display output
interface-->
   <backgroundColor opt="red,green,blue,yellow,purple,cyan,black,white"/><!--required, supported background
color-->
   <VideoEffect><!--required, display output effect parameters-->
    <brightnessLevel min="" max=""/><!--required, brightness level-->
    <contrastLevel min="" max=""/><!--required, contrast level-->
    <sharpnessLevel min="" max=""/><!--required, sharpness level-->
    <saturationLevel min="" max=""/><!--required, saturation level-->
    <hueLevel min="" max=""/><!--required, hue level-->
   </VideoEffect>
   <Audio>
    <enabled><!--required, whether to support independent audio--></enabled>
   </Audio>
   <VoutMotionFluency>
    <enabled><!--required, whether smooth video is supported by output port--></enabled>
   </VoutMotionFluency>
   <VoutResolutionEntry><!--required, multiple same-level nodes are allowed-->
    <resolutionName>
     <!--required, xs:string,  "1080P_60HZ(1920*1080)"..., display output resolution name-->
    </resolutionName>
    <index><!--required, xs:inter, the index corresponds to the resolution--></index>
```

```
    </VoutResolutionEntry>
  </VoutResourceEntry>
</VoutResource>
<layOutNo min="" max=""/>
<!--required, layout No.-->
<baseMapSpan></baseMapSpan>
<!--required, background picture span-->
<overlayStringNum>8</overlayStringNum>
<!--required, number of supported string overlay-->
<inputStreamNo min="" max=""/>
<!--required, input No.-->
<outputChanNo min="" max=""/>
<!--required, output channel No.-->
<planNo min="" max=""/>
<!--required, plan No.-->
<maxPictureViewNum>4</maxPictureViewNum>
<!--required, max. display channels, not displayed if not supported-->
<maxCamGroupNum></maxCamGroupNum>
<!--required, max. groups, not displayed if not supported-->
<maxNetSignalNum></maxNetSignalNum>
<!--required, max. number of network sources, not displayed if not supported-->

<ScreenLinkConfig>
  <!--required, the node is not required if screen connection adjustment is not supported-->
  <enabled>true</enabled>
  <!--required, screen connection adjustment-->
</ScreenLinkConfig>

<StreamMedia>
  <!--required, the node is not required if stream media is not supported-->
  <enabled>true</enabled>
  <!--required, stream media,-->
</StreamMedia>
<ScreenControl>
  <!--required, the node is not required if screen control is not supported-->
  <enabled>true</enabled>
  <!--required, screen control,-->
</ScreenControl>
  <!--required, the node is not required if screen control is not supported-->
  <enabled>true</enabled>
  <!--required, screen control,-->
</ScreenControl>
<maxLocalInputDeviceNum>
  <!--required, max. number of local encoding input devices, not returned if not supported-->
</maxLocalInputDeviceNum>

<maxLocalOutputDeviceNum>
  <!--required, number of local display output devices, not returned if not supported-->
</maxLocalOutputDeviceNum>

<wallNo min="" max=""/>
<!--, video wall No.-->
```

```
 <VirtualScreen>
  <!--required, ultra-HD input sub system capability (virtual screen capability), not returned if not supported-->
  <InputChanInitCap>
   <!--optional, virtual screen input channel initialization capability-->
   <inputChanNums>
    <!--optional, max. number of input channels, xs:integer-->
   </inputChanNums>
   <maxJoinResolution>
    <!--optional, max. multi-screen resolution (number of 1080P), xs:integer-->
   </maxJoinResolution>
   <selfdefinResNums>
    <!--optional, max. number of custom resolution supported by single channel, xs:integer-->
   </selfdefinResNums>
  </InputChanInitCap>
<Resolution>
    <!--required, display resolution supported by virtual screen-->
    <ResolutionEntry>
     <!--required, multiple same-level nodes are allowed-->
     <resolutionName>
<!--required, xs:string, "1080P_60HZ(1920*1080)"...,display output resolution name-->
</resolutionName>
      <index>
      <!--required, xs:inter,the index corresponding to the resolution-->
</index>
    </ResolutionEntry>
  </Resolution>
 </VirtualScreen>

 <baseMapWinNo min="" max=""/>
 <!--required, supported background picture window No. range-->
 <BaseMapPicSize>
  <!--required, size of supported background picture, if the node is not returned, it will be processed as max
resolution of 5120*2880, max file size of 6 MB-->
  <width min="" max=""/>
  <!--required, picture width-->
  <height min="" max=""/>
  <!--required, picture height-->
  <maxPicFileSize>10</maxPicFileSize>
  <!--required, max size of background picture, unit: MB-->
 </BaseMapPicSize>
 <BaseMapWinMove>
  <!--required, whether to support background picture window moving-->
  <enabled>true</enabled>
 </BaseMapWinMove>
 <PlatformPassiveDecode>
  <!--required, whether passive decoding is supported by platform-->
  <enabled>true</enabled>
 </PlatformPassiveDecode>

 <WallLogo>
  <!--required, video wall logo, the node is returned by device that supports logo No.-->
```

```
  <logoNo min="" max=""/>
  <!--required, range of supported logo No., not displayed if not supported-->
  <logoSize min="" max=""/>
  <!--required, supported logo size, in byte-->
  <logoNameLen min="" max=""/>
  <!--required, logo name length size-->
  <logoStatus opt = "show, hide"/>
  <!--required, logo display status-->
  <logoCorordinateX min="" max=""/>
  <!--required, logo display area x coordinate range-->
  <logoCorordinateY min="" max=""/>
  <!--required, logo display area y coordinate range-->
  <LogoFlash>
   <!--required, logo flash-->
   <enabled>true</enabled>
  </LogoFlash>
  <LogoTranslucent>
   <!--required, logo translucent-->
   <enabled>true</enabled>
  </LogoTranslucent>
  <LogoDelete>
   <!--required, logo delete-->
   <enabled>true</enabled>
  </LogoDelete>
</WallLogo>

<WindowLoop>
  <!--required, window auto-switch decoding-->
  <maxLoopNum>12</maxLoopNum>
  <!--required, supported number of window auto-switch decoding channels-->
  <maxMonitorNum>64</maxMonitorNum>
  <!--required, max number of cameras of each auto-switch channel-->
</WindowLoop>
<WindowDecResourceAlloc>
  <!--required, window decoding resource allocation, not displayed if not supported-->
  <WindowDecResourceAllocEntry>
   <!--required, window decoding resource allocation, multiple same-level nodes are allowed-->
   <name>D1</name>
   <!--required, allocation resource name-->
   <index>1</index>
   <!--required, index-->
  </WindowDecResourceAllocEntry>
</WindowDecResourceAlloc>
<MotionFluency>
  <!--required, whether to support window fluent video-->
  <enabled>true</enabled>
</MotionFluency>

<WinZoomStatus>
  <!--required, get window digital zoome status, not displayed if not supported-->
  <enabled>true</enabled>
</WinZoomStatus>
```

```
<WinTopAndBottom>
 <!--required, window set top/bottom-->
 <enabled>true</enabled>
</WinTopAndBottom>

<CloseAllWin>
 <!--required, close all windows-->
 <enabled>true</enabled>
</CloseAllWin>

<windowStaticMode opt="black,lastframe"/>
<!--required, stop decoding window display mode configuration-->
<LowLight opt="off,level1,level2, level3, level4, level5, level6, level7, level8"/>
<!--required, low illumination-->
<NosieReduction>
 <!--required, whether to support 3D noise reduction-->
 <enabled>true</enabled>
 <autoNosieReduction>enable</autoNosieReduction>
</NosieReduction>

<Defog opt="off,level1,level2, level3, level4, level5, level6, level7"/>
<!--required, defog level-->

<VcaDecode>
 <!--required, whether to support smart decoding-->
 <enabled>true</enabled>
 <allWinCtrl>true</allWinCtrl>
 <!--optional, operation in all windows-->
</VcaDecode>

<AllWinDecSwitch>
 <!--required, start/stop decoding in all windows-->
 <enabled>true</enabled>
</AllWinDecSwitch>

<VirtualLed>
 <!--required, virtual led-->
 <dispMode opt="transparent, translucent,cover"/>
 <!--required, display mode-->
 <FontColor>
  <!--required, font color-->
  <colorY min="" max=""/>
  <colorU min="" max=""/>
  <colorV min="" max=""/>
 </FontColor>
 <BackgroundColor>
  <!--required, background color-->
  <colorY min="" max=""/>
  <colorU min="" max=""/>
  <colorV min="" max=""/>
 </BackgroundColor>
```

```
  <characterNum min="" max=""/>
  <!--required, number of characters-->
  <moveMode opt="normal, smooth,static"/>
  <!--required, moving mode-->
  <moveSpeed opt="speed1,speed2"/>
  <!--required, character moving speed-->
  <fontSize opt="1times,2times,4times"/>
  <!--required, font size-->
  <moveDirection opt="lefttoright,righttoleft,toptobottom,bottomtotop"/>
  <!--required, character moving direction-->
</VirtualLed>
<ExternalMatrix>
  <!--required, external matrix capability-->
  <maxMatrixNum >4</maxMatrixNum>
  <!--required, max number of external matrixes-->
  <matrixNameLen min="" max=""/>
  <!--required, name length of external matrix-->
  <maxMatrixInputNum>256</maxMatrixInputNum>
  <!--required, max number of supported external matrix input-->
  <maxMatrixOutputNum>256</maxMatrixOutputNum>
  <!--required, max number of supported external matrix output-->
  <matrixChanType opt="BNC,VGA,RGB,DVI"/>
  <!--required, supported matrix output channel type-->
  <matrixProtocol opt="ZT1.0,ZT1.0,Extron,Creator"/>
  <!--required, supported matrix protocol-->
  <matrixType opt="analogmatrix,digitalmatrix"/>
  <!--required, supported matrix type-->
  <DigitalMatrix>
    <!--required, digital matrix capability-->
    <nicNum min="" max=""/>
    <!--required, supported network port No.-->
  </DigitalMatrix>
  <AnalogMatrix>
    <serPortNum min="" max=""/>
    <!--required, range of controller serial port No., serial port range based on serial port capability set-->
  </AnalogMatrix>
</ExternalMatrix>
<!--required, supported picture display parameters-->
<PicViewParam>
  <picResolution opt="QCIF,CIF,D1" />
  <!--required, picture resolution supported by display-->
  <picFrameRate min="" max=""/>
  <!--required, frame rate supported by display-->
</PicViewParam>

<!--required, display channel No. on video wall-->
<ShowDispChanNo>
  <channelType opt="DisplayChanNo,ScreenNo"/>
  <!--required, supported channel No. type for display-->
  <SupportDispByWallNo>
    <!--required, display by wall No.-->
    <enabled>true</enabled>
```

```
  </SupportDispByWallNo>
</ShowDispChanNo>

<AudioMatrix>
  <!--required, audio matrix switch-->
  <audioChanNameLen min="" max=""/>
  <!--required, audio channel name length-->
  <audioSwitchType opt="switchbyip,switchbywin" />
  <!--required, supported audio switch mode-->
</AudioMatrix>

<baseMapWinNo min="" max="" />
<!--required, range of supported background picture window No.-->

<DownloadLogo>
  <!--required, LOGO download-->
  <enabled>true</enabled>
</DownloadLogo>

<JointSignal>
  <!--required, joint signal source-->
  <enabled>true</enabled>
  <JointNo min="" max=""/><!--required, joint No.-->?
  <JointItem>
    <!--required, type of signal source that supports jointing, multiple same-level nodes are allowed-->
    <id>
      <!--required, signal source ID-->
    </id>
    <SignalType>bnc</SignalType>
    <!--required, xs:string,bnc,vga,hdmi,dvi,ip,rgb,matrix, yprpb,usb,sdi,hdi,dp,hdtvi -->
  </JointItem>
  <JointScale>
    <!--optional, jointing scale, return if supported by MVC-->
    <X86JointScale>
      <!--optional, X86 jointing board capability-->
      <maxInputNum></maxInputNum>
      <!--optional, max. number of inputs of single jointing-->
      <maxHeight></maxHeight>
      <!--optional, max. jointing height-->
      <maxWidth></maxWidth>
      <!--optional, max. jointing width-->
    </X86JointScale>
    <NormalJointScale>
      <!--optional, normal jointing board capability-->
      <maxInputNum></maxInputNum>
      <!--optional, max. number of inputs of single jointing-->
      <maxHeight></maxHeight>
      <!--optional, max. jointing height-->
      <maxWidth></maxWidth>
      <!--optional, max. jointing width-->
    </NormalJointScale>
  </JointScale>
```

```
  <DeleteJoint>
    <enabled>true</enabled>
    <!--optional, xs:string, support deleting jointing signal source-->
  </DeleteJoint>
</JointSignal>

<SupportGetPlayingPlan>
  <!--required, get plan in operation-->
  <enabled>true</enabled>
</SupportGetPlayingPlan>

<InputStreamV40>
  <!--required, local source V40 extension configuration-->
  <ColorMode opt="SelfDefine,Sharp,Ordinary,Soft"/>
  <!--required, color mode-->
</InputStreamV40>

<InputStreamCut>
  <!--required, signal source clipping-->
  <CutTop min="" max=""/>
  <!--required, top clipping pixel range-->
  <CutBottom min="" max=""/>
  <!--required, bottom clipping pixel range-->
  <CutLeft min="" max=""/>
  <!--required, left clipping pixel range-->
  <CutRight min="" max=""/>
  <!--required, right clipping pixel range-->
</InputStreamCut>

<WindowCap>
  <!--optional, window capability-->
  <isSupportPhysicsResolutionWnd>
    <!--optional, xs:boolean,"true", whether to support opening window in physical resolution-->
  </isSupportPhysicsResolutionWnd>
  <wndWidthAlignUint>
    <!--dep,xs:integer, physical resolution window width alignment unit-->
  </wndWidthAlignUint>
  <wndHeightAlignUint>
    <!--dep,xs:integer, physical resolution window height alignment unit-->
  </wndHeightAlignUint>
  <isSupportDecodeOSD>
    <!--optional, xs:boolean,"true", whether to support decoding OSD configuration-->
  </isSupportDecodeOSD>
</WindowCap>

<ResetOutputDisplayPosition>
  <!--required, unbind video wall output channel-->
  <enabled>
    <!--required, xs:boolean-->
  </enabled>
</ResetOutputDisplayPosition>
<InputBoardCfgList>
```

```
  <!--optional, input board parameter configuration capability-->
  <InputBoardCfg>
   <slotNo>
    <!--required, xs:integer, slot No.-->
   </slotNo>
   <fullFrameEnable>
    <!--required, xs:Boolean, whether to enable full frame rate fluent video-->
   </fullFrameEnable>
  </InputBoardCfg>
 </InputBoardCfgList>

 <isSupportLEDArea>
  <!--optional, xs:boolean,"true", whether to support LED area-->
 </isSupportLEDArea>

 <SubStreamAutoSwitchCap>
  <!--optional, sub-stream auto switch-->
  <subWndWidth min="" max="">
   <!--required, xs:integer, sub window width-->
  </subWndWidth>
  <subWndHeight min="" max="">
   <!--required, xs:integer, sub window height-->
  </subWndHeight>
 </SubStreamAutoSwitchCap>

 <streamFailedMode opt="noSignal,lastFrame"/>
 <!--optional, xs:string, window streaming failure display mode: noSignal- display "no network video signal",
lastFrame- display last frame-->

 <isSupportWallConference>
  <!--optional, xs:boolean,"true", whether to support meeting video wall-->
 </isSupportWallConference>

 <isSupportBaseMapCycleSwitch>
  <!--optional, xs:boolean,"true", whether the plan supports changing background picture
(NET_DVR_SWITCH_BASEMAP)-->
 </isSupportBaseMapCycleSwitch>

 <isSupportEDIDResolution>
  <!--optional, xs:boolean,"true", whether to support EDID resolution-->
 </isSupportEDIDResolution>
</WallAbility>
```

A.5.6 XML_WallWindowCap

XML message about video wall window capability

```
<WallWindowCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
 <windowMode optional="1,4,9,16,25,36">  <!--optional, xs:integer--></windowMode>
 <CycleCap/><!--optional-->
 <isSupportWinTopBottom><!--optional, xs:boolean--></isSupportWinTopBottom>
```

```
  <isSupportPlayBack><!--optional, xs:boolean--></isSupportPlayBack>
  <isSupportPicCapture><!--optional, xs:boolean--></isSupportPicCapture>
  <isSupportDecodeDelay><!--optional, xs:boolean--></isSupportDecodeDelay>
  <isSupportSubWndAmplify><!--optional, xs:boolean--></isSupportSubWndAmplify>
  <isSupportFullFrame><!--optional, xs:boolean--></isSupportFullFrame>
  <isSptMutiScreenGetSubStream><!--optional, xs:boolean, whether it supports auto switching to sub-stream in multi-screen mode--></isSptMutiScreenGetSubStream>
  <isSptCatchStreamAlarmHint><!--optional, xs:boolean, whether it supports configuring parameters for streaming alarm notification--></isSptCatchStreamAlarmHint>
</WallWindowCap>
```

A.6 Device Network SDK Errors

The errors that may occur during the device network SDK integration are listed here for reference. You can search for the error descriptions according to the error codes or names returned by a specific API (NET_DVR_GetLastError or NET_DVR_GetErrorMsg).

General Errors

Error Name	Error Code	Error Description
NET_DVR_NOERROR	0	No error.
NET_DVR_PASSWORD_ERROR	1	Incorrect user name or password.
NET_DVR_NOENOUGHPRI	2	No permission.
NET_DVR_NOINIT	3	Uninitialized.
NET_DVR_CHANNEL_ERROR	4	Incorrect channel No.
NET_DVR_OVER_MAXLINK	5	No more device can be connected.
NET_DVR_VERSIONNOMATCH	6	Version mismatches.
NET_DVR_NETWORK_FAIL_CONNECT	7	Connecting to device failed. The device is offline or network connection timed out.
NET_DVR_NETWORK_SEND_ERROR	8	Sending data to device failed.
NET_DVR_NETWORK_RECV_ERROR	9	Receiving data from device failed.
NET_DVR_NETWORK_RECV_TIMEOUT	10	Receiving data from device timed out.
NET_DVR_NETWORK_ERRORDATA	11	The data sent to the device is illegal, or the data received from the device error. E.g. The input data is not supported by the device for remote configuration.

Error Name	Error Code	Error Description
NET_DVR_ORDER_ERROR	12	API calling order error.
NET_DVR_OPERNOPERMIT	13	No permission for this operation.
NET_DVR_COMMANDTIMEOUT	14	Executing device command timed out.
NET_DVR_ERRORSERIALPORT	15	Incorrect serial port No. The specified serial port does not exist.
NET_DVR_ERRORALARMPORT	16	Alarm port No. error. The alarm input or output port of the specified device does not exist.
NET_DVR_PARAMETER_ERROR	17	Incorrect parameter. The input or output parameters of the SDK API is empty, or the parameter value or format is invalid.
NET_DVR_CHAN_EXCEPTION	18	Device channel is in exception status.
NET_DVR_NODISK	19	No HDD in the device.
NET_DVR_ERRORDISKNUM	20	Incorrect HDD No.
NET_DVR_DISK_FULL	21	HDD full.
NET_DVR_DISK_ERROR	22	HDD error.
NET_DVR_NOSUPPORT	23	Device does not support this function.
NET_DVR_BUSY	24	Device is busy.
NET_DVR_MODIFY_FAIL	25	Failed to edit device parameters.
NET_DVR_PASSWORD_FORMAT_ ERROR	26	Invalid password format.
NET_DVR_DISK_FORMATING	27	HDD is formatting. Failed to startup.
NET_DVR_DVRNORESOURCE	28	Insufficient device resources.
NET_DVR_DVROPRATEFAILED	29	Device operation failed.
NET_DVR_OPENHOSTSOUND_FAIL	30	Failed to collect local audio data or open audio output during two-way audio and broadcast.
NET_DVR_DVRVOICEOPENED	31	Two-way audio channel is occupied.
NET_DVR_TIMEINPUTERROR	32	Incorrect time input.
NET_DVR_NOSPECFILE	33	No video file for playback.

Error Name	Error Code	Error Description
NET_DVR_CREATEFILE_ERROR	34	Failed to create a file during local recording, saving picture, getting configuration file or downloading video file remotely.
NET_DVR_FILEOPENFAIL	35	Failed to open a file. The file does not exist or directory error.
NET_DVR_OPERNOTFINISH	36	Operation conflicted.
NET_DVR_GETPLAYTIMEFAIL	37	Failed to get the current played time.
NET_DVR_PLAYFAIL	38	Failed to play.
NET_DVR_FILEFORMAT_ERROR	39	Invalid file format.
NET_DVR_DIR_ERROR	40	File directory error.
NET_DVR_ALLOC_RESOURCE_ERROR	41	Allocating resources failed.
NET_DVR_AUDIO_MODE_ERROR	42	Invalid sound card mode error. The opened sound play mode and configured mode mismatched.
NET_DVR_NOENOUGH_BUF	43	Insufficient buffer for receiving data or saving picture.
NET_DVR_CREATESOCKET_ERROR	44	Failed to create SOCKET.
NET_DVR_SETSOCKET_ERROR	45	Failed to set SOCKET.
NET_DVR_MAX_NUM	46	No more registrations and live views can be connected.
NET_DVR_USERNOTEXIST	47	The user doest not exist. The user ID is logged out or unavailable.
NET_DVR_WRITEFLASHERROR	48	Writing FLASH error during device upgrade.
NET_DVR_UPGRADEFAIL	49	Failed to upgrade device. Network problem or language mismatches.
NET_DVR_CARDHAVEINIT	50	The decoding card is already initialized.
NET_DVR_PLAYERFAILED	51	Failed to call the function of player SDK.
NET_DVR_MAX_USERNUM	52	No more users can log in to.

Error Name	Error Code	Error Description
NET_DVR_GETLOCALIPANDMACFAIL	53	Failed to get the IP address or physical address of local PC.
NET_DVR_NOENCODEING	54	The decoding function of this channel is not enabled.
NET_DVR_IPMISMATCH	55	IP address mismatches.
NET_DVR_MACMISMATCH	56	MAC address mismatches.
NET_DVR_UPGRADELANGMISMATCH	57	The language of upgrade file mismatches.
NET_DVR_MAX_PLAYERPORT	58	No more channels can be started to play.
NET_DVR_NOSPACEBACKUP	59	Insufficient space to back up file.
NET_DVR_NODEVICEBACKUP	60	No backup device found.
NET_DVR_PICTURE_BITS_ERROR	61	Picture pixel bit mismatches. Only 24 bits are allowed.
NET_DVR_PICTURE_DIMENSION_ ERROR	62	Too large picture. The height*width should be less than 128x256.
NET_DVR_PICTURE_SIZ_ERROR	63	Too large picture. The picture size should be smaller than 100K.
NET_DVR_LOADPLAYERSDKFAILED	64	Failed to load the player(PlayCtrl.dll, SuperRender.dll, AudioRender.dll) to the current directory.
NET_DVR_LOADPLAYERSDKPROC_ ERROR	65	Failed to find the function in player SDK.
NET_DVR_LOADDSSDKFAILED	66	Failed to load the DS SDK to the current directory.
NET_DVR_LOADDSSDKPROC_ERROR	67	Failed to find the function in the DS SDK.
NET_DVR_DSSDK_ERROR	68	Failed to call the API in the hardware decoding library.
NET_DVR_VOICEMONOPOLIZE	69	The sound card is exclusive.
NET_DVR_JOINMULTICASTFAILED	70	Failed to join to multicast group.
NET_DVR_CREATEDIR_ERROR	71	Failed to create log file directory.
NET_DVR_BINDSOCKET_ERROR	72	Failed to bind socket.

Error Name	Error Code	Error Description
NET_DVR_SOCKETCLOSE_ERROR	73	Socket disconnected. Network disconnected or the destination is unreachable.
NET_DVR_USERID_ISUSING	74	Operation is executing. Failed to log out.
NET_DVR_SOCKETLISTEN_ERROR	75	Failed to listen.
NET_DVR_PROGRAM_EXCEPTION	76	Program exception.
NET_DVR_WRITEFILE_FAILED	77	Failed to write file during local recording, downloading file remotely or saving picture.
NET_DVR_FORMAT_READONLY	78	The HDD is read-only. Formatting is forbidden.
NET_DVR_WITHSAMEUSERNAME	79	The user name already exists.
NET_DVR_DEVICETYPE_ERROR	80	Device model mismatches when importing parameters.
NET_DVR_LANGUAGE_ERROR	81	Language mismatches when importing parameters.
NET_DVR_PARAVERSION_ERROR	82	Software version mismatches when importing parameters.
NET_DVR_IPCHAN_NOTALIVE	83	The external IP channel is offline live view.
NET_DVR_RTSP_SDK_ERROR	84	Failed to load StreamTransClient.dll.
NET_DVR_CONVERT_SDK_ERROR	85	Failed to load SystemTransform.dll.
NET_DVR_IPC_COUNT_OVERFLOW	86	No more IP channels can access to.
NET_DVR_MAX_ADD_NUM	87	No more video tags can be added.
NET_DVR_PARAMMODE_ERROR	88	Invalid parameter mode of image enhancement.
NET_DVR_CODESPITTER_OFFLINE	89	Code distributer is offline.
NET_DVR_BACKUP_COPYING	90	Device is backing up.
NET_DVR_CHAN_NOTSUPPORT	91	This operation is not supported by the channel.
NET_DVR_CALLINEINVALID	92	The height line is too concentrated, or the length line is not inclined enough.

Error Name	Error Code	Error Description
NET_DVR_CALCANCELCONFLICT	93	Cancel calibration conflict, if the rule and global actual size filter are configured.
NET_DVR_CALPOINTOUTRANGE	94	The calibration point is out of limitation.
NET_DVR_FILTERRECTINVALID	95	The size filter does not meet the requirement.
NET_DVR_DDNS_DEVOFFLINE	96	Device has not registered to DDNS.
NET_DVR_DDNS_INTER_ERROR	97	DDNS internal error.
NET_DVR_FUNCTION_NOT_SUPPORT_OS	98	This function is not supported by this Operating system.
NET_DVR_DEC_CHAN_REBIND	99	Decoding channel binding display output is limited.
NET_DVR_INTERCOM_SDK_ERROR	100	Failed to load the two-way audio SDK of the current directory.
NET_DVR_NO_CURRENT_UPDATEFILE	101	No correct upgrade packet.
NET_DVR_USER_NOT_SUCC_LOGIN	102	Login failed.
NET_DVR_USE_LOG_SWITCH_FILE	103	The log switch file is under using.
NET_DVR_POOL_PORT_EXHAUST	104	No port can be bound in the port pool.
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	Incorrect stream packaging format.
NET_DVR_IPPARA_IPID_ERROR	106	Incorrect IPID for IP access configuration.
NET_DVR_LOAD_HCPREVIEW_SDK_ERROR	107	Failed to load the live view component.
NET_DVR_LOAD_HCVOICETALK_SDK_ERROR	108	Failed to load the audio component.
NET_DVR_LOAD_HCALARM_SDK_ERROR	109	Failed to load the alarm component.
NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	Failed to load the playback component.

Error Name	Error Code	Error Description
NET_DVR_LOAD_HCDISPLAY_SDK_ERROR	111	Failed to load the display component.
NET_DVR_LOAD_HCINDUSTRY_SDK_ERROR	112	Failed to load application component.
NET_DVR_LOAD_HCGENERALCFGMGR_SDK_ERROR	113	Failed to load the general configuration management component.
NET_DVR_CORE_VER_MISMATCH	121	Component version and core version mismatched when loading the component singly.
NET_DVR_CORE_VER_MISMATCH_HCPREVIEW	122	Live view component version and core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCVOICETALK	123	Audio component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCALARM	124	Alarm component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCPLAYBACK	125	Playback component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCDISPLAY	126	Display component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCINDUSTRY	127	Application component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCGENERALCFGMGR	128	General configuration management component version and the core version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPREVIEW	136	Live view component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCVOICETALKy	137	Audio component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCALARM	138	Alarm component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPLAYBACK	139	Playback component version and SDK version mismatched.

Error Name	Error Code	Error Description
NET_DVR_COM_VER_MISMATCH_HCDISPLAY	140	Display component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCINDUSTRY	141	Application component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCGENERALCFGMGR	142	General configuration management component version and SDK version mismatched.
NET_DVR_ALIAS_DUPLICATE	150	Duplicated alias(for HiDDNS configuration).
NET_DVR_USERNAME_NOT_EXIST	152	User name does not exist (error code of network camera and network speed dome with version from 5.1.7 to 5.3.1).
NET_ERR_USERNAME_LOCKED	153	The user name is locked.
NET_DVR_INVALID_USERID	154	Invalid user ID.
NET_DVR_LOW_LOGIN_VERSION	155	The version is too low.
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	Failed to load libeay32.dl.l
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	Failed to load ssleay32.dll.
NET_ERR_LOAD_LIBICONV	158	Failed to load libiconv.dll.
NET_ERR_SSL_CONNECT_FAILED	159	Connecting to SSL failed.
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	Failed to connect to test server.
NET_DVR_NAS_SERVER_INVALID_DIR	166	Failed to load NAS server to the directory, Invalid directory, or incorrect user name and password.
NET_DVR_NAS_SERVER_NOENOUGH_PRI	167	Failed to load NAS server th the directory. No permission.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	The server uses domain name without configuring DNS, the domain name may be invalid.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_GATEWAY	169	No gateway configured. Sending email may be failed.

Error Name	Error Code	Error Description
NET_DVR_TEST_SERVER_PASSWORD_ERROR	170	Incorrect user name or password of test server.
NET_DVR_EMAIL_SERVER_CONNECT_EXCEPTION_WITH_SMTP	171	Interaction exception between device and SMTP server.
NET_DVR_FTP_SERVER_FAIL_CREATE_DIR	172	FTP server creating directory failed.
NET_DVR_FTP_SERVER_NO_WRITE_PIR	173	FTP server has no wirting permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.
NET_DVR_INSUFFICIENT_STORAGEPOOL_SPACE	175	Storage pool space is full.
NET_DVR_STORAGEPOOL_INVALID	176	Invalid cloud storage pool. No storage pool configured or incorrect storage pool ID.
NET_DVR_EFFECTIVENESS_REBOOT	177	Restart to take effect.
NET_ERR_ANR_ARMING_EXIST	178	The ANR arming connection already exists(the error will be returned when arming with ANR function if the private SDK protocol arming connection is established).
NET_ERR_UPLOADLINK_EXIST	179	The ANR uploading connection already exists(the error will be returned when EHome protocol and private SDK protocol do not support ANR at the same time).
NET_ERR_INCORRECT_FILE_FORMAT	180	The imported file format is incorrect.
NET_ERR_INCORRECT_FILE_CONTENT	181	The imported file content is incorrect.
NET_ERR_MAX_HRUDP_LINK	182	No more HRUDP can be connected to device.
NET_ERR_MAX_PORT_MULTIPLEX	183	Maximum number of multiplexed ports reaches.
NET_ERR_CREATE_PORT_MULTIPLEX	184	Creating port multiplier failed.
NET_DVR_NONBLOCKING_CAPTURE_NOTSUPPORT	185	Non-blocking picture capture is not supported.

Error Name	Error Code	Error Description
NET_SDK_ERR_FUNCTION_INVALID	186	Invalid function. The asynchronous mode is enabled.
NET_SDK_ERR_MAX_PORT_ MULTIPLEX	187	Maximum number of multiplex ports reached.
NET_DVR_INVALID_LINK	188	Link has not been created or the link is invalid.
NET_DVR_NAME_NOT_ONLY	200	This name already exists.
NET_DVR_OVER_MAX_ARRAY	201	The number of RAID reaches the upper-limit.
NET_DVR_OVER_MAX_VD	202	The number of virtual disk reaches the upper-limit.
NET_DVR_VD_SLOT_EXCEED	203	The virtual disk slots are full.
NET_DVR_PD_STATUS_INVALID	204	The physical disk for rebuilding RAID is error.
NET_DVR_PD_BE_DEDICATE_SPARE	205	The physical disk for rebuilding RAID is specified as hot spare.
NET_DVR_PD_NOT_FREE	206	The physical disk for rebuilding RAID is busy.
NET_DVR_CANNOT_MIG2NEWMODE	207	Failed to migrate the current RAID type to the new type.
NET_DVR_MIG_PAUSE	208	Migration is paused.
NET_DVR_MIG_ABOUTED	209	Migration is cancelled.
NET_DVR_EXIST_VD	210	Failed to delete RAID. Virtual disk exists in the RAID.
NET_DVR_TARGET_IN_LD_ FUNCTIONAL	211	Target physical disk is a part of the virtual disk and it is working normally.
NET_DVR_HD_IS_ASSIGNED_ ALREADY	212	The specified physical disk is allocated as virtual disk.
NET_DVR_INVALID_HD_COUNT	213	The number of physical disks and specified RAID level mismatched.
NET_DVR_LD_IS_FUNCTIONAL	214	The RAID is normal. Failed to rebuild.
NET_DVR_BGA_RUNNING	215	Background task is executing.

Error Name	Error Code	Error Description
NET_DVR_LD_NO_ATAPI	216	Failed to create virtual disk by ATAPI disk.
NET_DVR_MIGRATION_NOT_NEED	217	There is no need to migrate the RAID.
NET_DVR_HD_TYPE_MISMATCH	218	The physical disk type is not allowed.
NET_DVR_NO_LD_IN_DG	219	No virtual disk. Operation failed.
NET_DVR_NO_ROOM_FOR_SPARE	220	Insufficient disk space. Failed to allocate the disk as hot spare.
NET_DVR_SPARE_IS_IN_MULTI_DG	221	The disk is already allocated as the hot spare of one RAID.
NET_DVR_DG_HAS_MISSING_PD	222	No disk in the RAID.
NET_DVR_NAME_EMPTY	223	The name is empty.
NET_DVR_INPUT_PARAM	224	Incorrect input parameters.
NET_DVR_PD_NOT_AVAILABLE	225	The physical disk is not available.
NET_DVR_ARRAY_NOT_AVAILABLE	226	The RAID is not available.
NET_DVR_PD_COUNT	227	Incorrect number of physical disks.
NET_DVR_VD_SMALL	228	Insufficient virtual disk space.
NET_DVR_NO_EXIST	229	Not exist.
NET_DVR_NOT_SUPPORT	230	This operation is not supported.
NET_DVR_NOT_FUNCTIONAL	231	The RAID status is exception.
NET_DVR_DEV_NODE_NOT_FOUND	232	The device node of virtual disk does not exist.
NET_DVR_SLOT_EXCEED	233	No more slots are allowed.
NET_DVR_NO_VD_IN_ARRAY	234	No virtual disk exists in the RAID.
NET_DVR_VD_SLOT_INVALID	235	Invalid virtual disk slot.
NET_DVR_PD_NO_ENOUGH_SPACE	236	Insufficient physical disk space.
NET_DVR_ARRAY_NONFUNCTION	237	Only the RAID in normal status supports to be migrated.
NET_DVR_ARRAY_NO_ENOUGH_ SPACE	238	Insufficient RAID space.
NET_DVR_STOPPING_SCANNING_ ARRAY	239	Pulling disk out safely or rescanning.

Error Name	Error Code	Error Description
NET_DVR_NOT_SUPPORT_16T	240	Creating RAID with size larger than 16T is not supported.
NET_DVR_ERROR_DEVICE_NOT_ ACTIVATED	250	The device is not activated (login failed.)
NET_DVR_ERROR_RISK_PASSWORD	251	Risky password.
NET_DVR_ERROR_DEVICE_HAS_ ACTIVATED	252	The device is already activated.
NET_DVR_ID_ERROR	300	The configured ID is invalid.
NET_DVR_POLYGON_ERROR	301	Invalid polygon shape.
NET_DVR_RULE_PARAM_ERROR	302	Invalid rule parameters.
NET_DVR_RULE_CFG_CONFLICT	303	Configured information conflicted.
NET_DVR_CALIBRATE_NOT_READY	304	No calibration information.
NET_DVR_CAMERA_DATA_ERROR	305	Invalid camera parameters.
NET_DVR_CALIBRATE_DATA_UNFIT	306	Invalid inclination angle for calibration.
NET_DVR_CALIBRATE_DATA_ CONFILICT	307	Calibration error.
NET_DVR_CALIBRATE_CALC_FAIL	308	Failed to calculate calibration parameter values of camera.
NET_DVR_CALIBRATE_LINE_OUT_ RECT	309	The inputted calibration line exceeds the external sample rectangle.
NET_DVR_ENTER_RULE_NOT_READY	310	No region entrance is configured.
NET_DVR_AID_RULE_NO_INCLUDE_ LANE	311	No lane configured in the traffic event rull(especially for traffic jam or driving against the traffic).
NET_DVR_LANE_NOT_READY	312	Lane not configured.
NET_DVR_RULE_INCLUDE_TWO_WAY	313	Two different directions are contained in event rule.
NET_DVR_LANE_TPS_RULE_CONFLICT	314	Lane and data rule conflicted.
NET_DVR_NOT_SUPPORT_EVENT_ TYPE	315	This event type is not supported.
NET_DVR_LANE_NO_WAY	316	The lane has no direction.

Error Name	Error Code	Error Description
NET_DVR_SIZE_FILTER_ERROR	317	Invalid size of filter frame.
NET_DVR_LIB_FFL_NO_FACE	318	No face picture exists in the image inputted when positioning feature point.
NET_DVR_LIB_FFL_IMG_TOO_SMALL	319	The inputted image is too small when positioning feature point.
NET_DVR_LIB_FD_IMG_NO_FACE	320	No face picture exists in the image inputted when detecting single face picture.
NET_DVR_LIB_FACE_TOO_SMALL	321	Face picture is too small when building model.
NET_DVR_LIB_FACE_QUALITY_TOO_BAD	322	The face picture quality is too poor when building model.
NET_DVR_KEY_PARAM_ERR	323	The configured advanced parameter is incorrect.
NET_DVR_CALIBRATE_DATA_ERR	324	Calibration sample number error, or data value error, or the sample points are beyond the horizontal line.
NET_DVR_CALIBRATE_DISABLE_FAIL	325	Canceling calibration is not allowed for configured rules.
NET_DVR_VCA_LIB_FD_SCALE_OUTRANGE	326	The minimum width and height of maximum filter frame are twice or more larger than the maximum width and height of minimum filter frame.
NET_DVR_LIB_FD_REGION_TOO_LARGE	327	Too large detection region. The maximum region should be 2/3 of the image.
NET_DVR_TRIAL_OVERDUE	328	Trial period is ended.
NET_DVR_CONFIG_FILE_CONFLICT	329	Device type and configuration file conflicted.
NET_DVR_FR_FPL_FAIL	330	Failed to positioning face feature points.
NET_DVR_FR_IQA_FAIL	331	Failed to test face picture quality.

Error Name	Error Code	Error Description
NET_DVR_FR_FEM_FAIL	332	Failed to extract the face feature points.
NET_DVR_FPL_DT_CONF_TOO_LOW	333	The face detection validity is too low when positioning face feature points.
NET_DVR_FPL_CONF_TOO_LOW	334	The validity of feature points positionong is too low.
NET_DVR_E_DATA_SIZE	335	Data size mismatches.
NET_DVR_FR_MODEL_VERSION_ERR	336	Incorrect model version in face model library.
NET_DVR_FR_FD_FAIL	337	Failed to detect face in the face recognition library.
NET_DVR_FA_NORMALIZE_ERR	338	Failed to normalize face attribute.
NET_DVR_DOG_PUSTREAM_NOT_MATCH	339	Dongle type and camera type mismatched.
NET_DVR_DEV_PUSTREAM_NOT_MATCH	340	Camera version mismatches.
NET_DVR_PUSTREAM_ALREADY_EXISTS	341	This camera is already added to other channels of devices.
NET_DVR_SEARCH_CONNECT_FAILED	342	Failed to connect to face retrieval server.
NET_DVR_INSUFFICIENT_DISK_SPACE	343	Insufficient storage space.
NET_DVR_DATABASE_CONNECTION_FAILED	344	Failed to connect to database.
NET_DVR_DATABASE_ADM_PW_ERROR	345	Incorrect database user name and password.
NET_DVR_DECODE_YUV	346	Decoding failed.
NET_DVR_IMAGE_RESOLUTION_ERROR	347	Invalid picture resolution
NET_DVR_CHAN_WORKMODE_ERROR	348	Invalid channel working mode.
NET_ERROR_TRUNK_LINE	711	Sub system is configured as the trunk line.
NET_ERROR_MIXED_JOINT	712	Mixed joint is not supported.

Error Name	Error Code	Error Description
NET_ERROR_DISPLAY_SWITCH	713	Switch of display channel is not supported.
NET_ERROR_USED_BY_BIG_SCREEN	714	Decoded resource is occupied by the big screen.
NET_ERROR_USE_OTHER_DEC_ RESOURCE	715	Using resources of other sub system is not allowed.
NET_ERROR_SCENE_USING	717	The scene is being used.
NET_ERR_NO_ENOUGH_DEC_ RESOURCE	718	Insufficient resources for decoding.
NET_ERR_NO_ENOUGH_FREE_ SHOW_RESOURCE	719	Insufficient resources for display.
NET_ERR_NO_ENOUGH_VIDEO_ MEMORY	720	Insufficient video storage resources.
NET_ERR_MAX _VIDEO_NUM	721	Insufficient resources for multiple channels.
NET_ERR_WINDOW_COVER_FREE_ SHOW_AND_NORMAL	722	Windows cover free display output channel and normal output channel.
NET_ERR_FREE_SHOW_WINDOW_ SPLIT	723	Window division is not supported for free display windows.
NET_ERR_INAPPROPRIATE_ WINDOW_FREE_SHOW	724	For the windows whose number is not integral multiple of the number of output channels, free display is not supported.
NET_DVR_TRANSPARENT_WINDOW_ NOT_SUPPORT_SPLIT	725	For windows whose transparency configuration is enabled, window division is not supported.
NET_DVR_SPLIT_WINDOW_NOT_ SUPPORT_TRANSPARENT	726	For windows whose window division is enabled, transparency configuration is not supported.
NET_ERR_TERMINAL_BUSY	780	The terminal busy.
NET_DVR_FUNCTION_RESOURCE_ USAGE_ERROR	791	Failed to enable this function. The resources is occupied by other functions.

Error Name	Error Code	Error Description
NET_DVR_DEV_NET_OVERFLOW	800	Network traffic is out of the limitation.
NET_DVR_STATUS_RECORDFILE_ WRITING_NOT_LOCK	801	Failed to lock. The video file is recording.
NET_DVR_STATUS_CANT_FORMAT_ LITTLE_DISK	802	Failed to format HDD. The HDD space is too small.
NET_SDK_ERR_REMOTE_DISCONNEC	803	Failed to connect to the remote terminal.
NET_SDK_ERR_RD_ADD_RD	804	Spare server cannot be added to spare server.
NET_SDK_ERR_BACKUP_DISK_EXCEPT	805	Backup disk exception.
NET_SDK_ERR_RD_LIMIT	806	No more spare server can be added.
NET_SDK_ERR_ADDED_RD_IS_WD	807	The added spare server is a working server.
NET_SDK_ERR_ADD_ORDER_WRONG	808	Adding flow error.
NET_SDK_ERR_WD_ADD_WD	809	Working server cannot be added to working server.
NET_SDK_ERR_WD_SERVICE_EXCETP	810	CVR service exception (For N+1 mode, it refers to CVR working server exception).
NET_SDK_ERR_RD_SERVICE_EXCETP	811	Spare CVR server exception.
NET_SDK_ERR_ADDED_WD_IS_RD	812	The added working server is spare server.
NET_SDK_ERR_PERFORMANCE_LIMIT	813	The performance reaches the upper-limit.
NET_SDK_ERR_ADDED_DEVICE_EXIST	814	This device already exists.
NET_SDK_ERR_INQUEST_RESUMING	815	Inquest resuming.
NET_SDK_ERR_RECORD_BACKUPING	816	Inquest video backing up.
NET_SDK_ERR_DISK_PLAYING	817	Playing.
NET_SDK_ERR_INQUEST_STARTED	818	Inquest started.
NET_SDK_ERR_LOCAL_OPERATING	819	Locally operating.
NET_SDK_ERR_INQUEST_NOT_START	820	Inquest is not started.

Error Name	Error Code	Error Description
NET_SDK_ERR_CHAN_AUDIO_BIND	821	The channel is not bound or binding two-way audio failed.
NET_DVR_N_PLUS_ONE_MODE	822	Ddevice is in N+1 mode. Cloud storage is not supported.
NET_DVR_CLOUD_STORAGE_OPENED	823	Cloud storage mode is enbaled.
NET_DVR_ERR_OPER_NOT_ALLOWED	824	Operation failed. The device is in N+0 taken over status.
NET_DVR_ERR_NEED_RELOCATE	825	The device is in N+0 taken over status. Get re-positioning information and try again.
NET_SDK_ERR_IR_PORT_ERROR	830	IR output error.
NET_SDK_ERR_IR_CMD_ERROR	831	IR output port command number error
NET_SDK_ERR_NOT_INQUESTING	832	Device is not in inquest status.
NET_SDK_ERR_INQUEST_NOT_ PAUSED	833	Device is not in paused status.
NET_DVR_CHECK_PASSWORD_ MISTAKE_ERROR	834	Incorrect verification code.
NET_DVR_CHECK_PASSWORD_NULL_ ERROR	835	Verification code is required.
NET_DVR_UNABLE_CALIB_ERROR	836	Failed to calibrate.
NET_DVR_PLEASE_CALIB_ERROR	837	Calibration first.
NET_DVR_ERR_PANORAMIC_CAL_ EMPTY	838	Panoramic calibration is empty in Flash.
NET_DVR_ERR_CALIB_FAIL_ PLEASEAGAIN	839	Calibration failed, please try again.
NET_DVR_ERR_DETECTION_LINE	840	Rule line configuration error. Please try again and make sure the line is within the red region.
NET_DVR_EXCEED_FACE_IMAGES_ ERROR	843	No more face pictures can be added.
NET_DVR_ANALYSIS_FACE_IMAGES_ ERROR	844	Picture recognition failed.

Error Name	Error Code	Error Description
NET_ERR_ALARM_INPUT_OCCUPIED	845	A<-1 alarm number is used for triggering vehicle capture.
NET_DVR_FACELIB_DATABASE_ERROR	846	Database version in face picture library mismatched.
NET_DVR_FACELIB_DATA_ERROR	847	Face picture library data error.
NET_DVR_FACE_DATA_ID_ERROR	848	Invalid face data PID.
NET_DVR_FACELIB_ID_ERROR	849	Invalid face picture library ID.
NET_DVR_EXCEED_FACE_LIBARY_ ERROR	850	No more face picture libraries can be established..
NET_DVR_PIC_ANALYSIS_NO_ TARGET_ERROR	851	No target recognized in the picture.
NET_DVR_SUBPIC_ANALYSIS_ MODELING_ERROR	852	Sub picture modeling failed.
NET_DVR_PIC_ANALYSIS_NO_ RESOURCE_ERROR	853	No VCA engine supports picture secondary recognition.
NET_DVR_ANALYSIS_ENGINES_NO_ RESOURCE_ERROR	854	No VCA engine.
NET_DVR_ANALYSIS_ENGINES_ USAGE_EXCEED_ERROR	855	Overload. The engine CPU reached 100%.
NET_DVR_EXCEED_HUMANMISINFO_ FILTER_ENABLED_ERROR	856	No more false alarm channel can be enabled.
NET_DVR_NAME_ERROR	857	Name error.
NET_DVR_NAME_EXIST_ERROR	858	The name already exists.
NET_DVR_FACELIB_PIC_IMPORTING_ ERROR	859	The pictures is importing to face picture library.
NET_DVR_PIC_FORMAT_ERROR	864	Invalid picture format.
NET_DVR_PIC_RESOLUTION_ INVALID_ERROR	865	Invalid picture resolution.
NET_DVR_PIC_SIZE_EXCEED_ERROR	866	The picture size is too large.
NET_DVR_PIC_ANALYSIS_TARGRT_ NUM_EXCEED_ERROR	867	Too many targets in the picture.
NET_DVR_ANALYSIS_ENGINES_ LOADING_ERROR	868	Initializing analysis engine.

Error Name	Error Code	Error Description
NET_DVR_ANALYSIS_ENGINES_ ABNORMA_ERROR	869	Analysis engine exception.
NET_DVR_ANALYSIS_ENGINES_ FACELIB_IMPORTING	870	Analysis engine is importing pictures to face picture library.
NET_DVR_NO_DATA_FOR_ MODELING_ERROR	871	No data for modeling.
NET_DVR_FACE_DATA_MODELING_ ERROR	872	Device is modeling picture. Concurrent processing is not supported.
NET_ERR_FACELIBDATA_OVERLIMIT	873	No more face picture can be added to the device (the data of imported face picture library)
NET_DVR_ANALYSIS_ENGINES_ ASSOCIATED_CHANNEL	874	Channel is linked to the analysis engine.
NET_DVR_ERR_CUSTOMID_LEN	875	The minimum length of upper layer custom ID is 32 bytes.
NET_DVR_ERR_CUSTOMFACELIBID_ REPEAT	876	The applied custom face picture library ID is duplicated
NET_DVR_ERR_CUSTOMHUMANID_ REPEAT	877	The applied custom person ID is duplicated.
NET_DVR_ERR_URL_DOWNLOAD_ FAIL	878	URL download failed.
NET_DVR_ERR_URL_DOWNLOAD_ NOTSTART	879	URL download has not started.
NET_DVR_CFG_FILE_SECRETKEY_ ERROR	880	The security verification key of configuration file is error.
NET_DVR_THERMOMETRY_REGION_ OVERSTEP_ERROR	883	Invalid thermometry region
NET_DVR_ERR_TOO_SHORT_ CALIBRATING_TIME	894	Too short time for calibration.
NET_DVR_ERR_AUTO_CALIBRATE_ FAILED	895	Auto calibration failed.
NET_DVR_ERR_VERIFICATION_FAILED	896	Verification failed.
NET_DVR_NO_TEMP_SENSOR_ERROR	897	No temperature sensor.

Error Name	Error Code	Error Description
NET_DVR_PUPIL_DISTANCE_ OVERSIZE_ERROR	898	The pupil distance is too large.
NET_ERR_WINCHAN_IDX	901	Window channel index error.
NET_ERR_WIN_LAYER	902	Window layer number error(the count of window layers on a single screen exceeds the max number).
NET_ERR_WIN_BLK_NUM	903	Window block number error(the count of screens that single window overlays exceeds the max number).
NET_ERR_OUTPUT_RESOLUTION	904	The output resolution error.
NET_ERR_LAYOUT	905	Layout index error.
NET_ERR_INPUT_RESOLUTION	906	The input resolution is not supported.
NET_ERR_SUBDEVICE_OFFLINE	907	The sub-device is off-line.
NET_ERR_NO_DECODE_CHAN	908	There is no free decoding channel.
NET_ERR_MAX_WINDOW_ABILITY	909	The upper limit of window number.
NET_ERR_ORDER_ERROR	910	Calling order error.
NET_ERR_PLAYING_PLAN	911	Be playing plan.
NET_ERR_DECODER_USED	912	Decoder board is being used.
NET_ERR_OUTPUT_BOARD_DATA_ OVERFLOW	913	Output board data overflow
NET_ERR_SAME_USER_NAME	914	Duplicate user name
NET_ERR_INVALID_USER_NAME	915	Invalid user name
NET_ERR_MATRIX_USING	916	Input matrix is in use.
NET_ERR_DIFFERENT_CHAN_TYPE	917	Different channel type (the type of matrix output channel mismatches that of the controller input channel)
NET_ERR_INPUT_CHAN_BINDED	918	Input channel has been bound by other matrix
NET_ERR_BINDED_OUTPUT_CHAN_ OVERFLOW	919	The matrix output channels in use exceeded the number bound by matrix and controller

Error Name	Error Code	Error Description
NET_ERR_MAX_SIGNAL_NUM	920	Number of input signals reached upper limit
NET_ERR_INPUT_CHAN_USING	921	Input channel is in use
NET_ERR_MANAGER_LOGON	922	Administrator has logged in, operation failed
NET_ERR_USERALREADY_LOGON	923	The user has logged in, operation failed
NET_ERR_LAYOUT_INIT	924	Scene is initializing, operation failed
NET_ERR_BASEMAP_SIZE_NOT_MATCH	925	Base image size does not match
NET_ERR_WINDOW_OPERATING	926	Window is in other operation, operation failed
NET_ERR_SIGNAL_UPLIMIT	927	Number of signal source window reached upper limit
NET_ERR_WINDOW_SIZE_OVERLIMIT	943	The window size exceeds the limit.
NET_ERR_MAX_WIN_OVERLAP	951	The number of windows overlap has reached the maximum limit.
NET_ERR_STREAMID_CHAN_BOTH_VALID	952	stream ID and channel number are both valid.
NET_ERR_NO_ZERO_CHAN	953	The device has no zero channel.
NEED_RECONNECT	955	Need redirection (for transcoding system)
NET_ERR_NO_STREAM_ID	956	The stream ID does not exist.
NET_DVR_TRANS_NOT_START	957	The transcoding has not been started.
NET_ERR_MAXNUM_STREAM_ID	958	The number of stream ID has reached the maximum limit.
NET_ERR_WORKMODE_MISMATCH	959	The work mode does not match with the requirement.
NET_ERR_MODE_IS_USING	960	It Has been working in current mode.
NET_ERR_DEV_PROGRESSING	961	The device is in processing
NET_ERR_PASSIVE_TRANSCODING	962	It is in transcoding.

Error Name	Error Code	Error Description
NET_DVR_ERR_WINDOW_SIZE_ PLACE	975	Wrong window position.
NET_DVR_ERR_RGIONAL_ RESTRICTIONS	976	Screen distance exceeds the limit.
NET_DVR_ERR_CLOSE_WINDOWS	984	Operation failed. Close the window first.
NET_DVR_ERR_MATRIX_LOOP_ ABILITY	985	Beyond the cycle decoding capacity.
NET_DVR_ERR_MATRIX_LOOP_TIME	986	Invalid cycle decoding time.
NET_DVR_ERR_LINKED_OUT_ABILITY	987	No more linked camera can be added.
NET_ERR_RESOLUTION_NOT_ SUPPORT_ODD_VOUT	990	The resolution is not supported (odd No.).
NET_ERR_RESOLUTION_NOT_ SUPPORT_EVEN_VOUT	991	The resolution is not supported (even No.).
NET_ERR_UnitConfig_Failed	998	Unit configuration failed.
XML_ABILITY_NOTSUPPORT	1000	Getting capability node is not supported
XML_ANALYZE_NOENOUGH_BUF	1001	Not enough output memory
XML_ANALYZE_FIND_LOCALXML_ ERROR	1002	Failed to find related local xml
XML_ANALYZE_LOAD_LOCALXML_ ERROR	1003	Loading local xml error
XML_NANLYZE_DVR_DATA_FORMAT_ ERROR	1004	Device capability data format error
XML_ANALYZE_TYPE_ERROR	1005	Capability set type error
XML_ANALYZE_XML_NODE_ERROR	1006	XML capability node format error
XML_INPUT_PARAM_ERROR	1007	Input capability XML node value error
XML_VERSION_MISMATCH	1008	XML version does not match
NET_ERR_TRANS_CHAN_START	1101	Transparent channel has been open, operation failed
NET_ERR_DEV_UPGRADING	1102	Device is upgrading

Error Name	Error Code	Error Description
NET_ERR_MISMATCH_UPGRADE_PACK_TYPE	1103	Upgrade pack type does not match
NET_ERR_DEV_FORMATTING	1104	Device is formatting
NET_ERR_MISMATCH_UPGRADE_PACK_VERSION	1105	Upgrade pack version does not match
NET_ERR_PT_LOCKED	1106	PT is locked.
NET_DVR_ERR_ILLEGAL_VERIFICATION_CODE	1111	Illegal verification code. Change the verification code.
NET_DVR_ERR_LACK_VERIFICATION_CODE	1112	No verification code. Enter the verification code.
NET_DVR_ERR_FORBIDDEN_IP	1113	The IP address cannot be configured.
NET_DVR_ERR_HTTP_BKN_EXCEED_ONE	1125	Up to one channel's ANR function can be enabled.
NET_DVR_ERR_FORMATTING_FAILED	1131	Formatting HDD failed.
NET_DVR_ERR_ENCRYPTED_FORMATTING_FAILED	1132	Formatting encrypted HDD failed.
NET_DVR_ERR_WRONG_PASSWORD	1133	Verifying password of SD card failed. Incorrect password.
NET_ERR_SEARCHING_MODULE	1201	Searching peripherals.
NET_ERR_REGISTERING_MODULE	1202	Registering external module
NET_ERR_GETTING_ZONES	1203	Getting arming region parameter
NET_ERR_GETTING_TRIGGERS	1204	Getting trigger
NET_ERR_ARMED_STATUS	1205	System is in arming status
NET_ERR_PROGRAM_MODE_STATUS	1206	System is in programming mode
NET_ERR_WALK_TEST_MODE_STATUS	1207	System is in pacing measuring mode
NET_ERR_BYPASS_STATUS	1208	Bypass status
NET_ERR_DISABLED_MODULE_STATUS	1209	Function not enabled
NET_ERR_NOT_SUPPORT_OPERATE_ZONE	1210	Operation is not supported by arming region
NET_ERR_NOT_SUPPORT_MOD_MODULE_ADDR	1211	Module address cannot be modified

Error Name	Error Code	Error Description
NET_ERR_UNREGISTERED_MODULE	1212	Module is not registered
NET_ERR_PUBLIC_SUBSYSTEM_ ASSOCIATE_SELF	1213	Public sub system associate with its self
NET_ERR_EXCEEDS_ASSOCIATE_ SUBSYSTEM_NUM	1214	Number of associated public sub system reached upper limit
NET_ERR_BE_ASSOCIATED_BY_ PUBLIC_SUBSYSTEM	1215	Sub system is associated by other public sub system
NET_ERR_ZONE_FAULT_STATUS	1216	Arming region is in failure status
NET_ERR_SAME_EVENT_TYPE	1217	Same event type exists in enable event trigger alarm output and disable event trigger alarm output
NET_ERR_ZONE_ALARM_STATUS	1218	Arming region is in alarm status
NET_ERR_EXPANSION_BUS_SHORT_ CIRCUIT	1219	Extension bus short-circuit
NET_ERR_PWD_CONFLICT	1220	Password conflict, e.g., lock password is identical with duress password
NET_ERR_DETECTOR_GISTERED_BY_ OTHER_ZONE	1221	Detector has been registered by other arming regions
NET_ERR_DETECTOR_GISTERED_BY_ OTHER_PU	1222	Detector has been registered by other hosts
NET_ERR_DETECTOR_DISCONNECT	1223	Detector offline
NET_ERR_CALL_BUSY	1224	Device in call
NET_ERR_FILE_NAME	1357	File name error, empty or invalid
NET_ERR_BROADCAST_BUSY	1358	Device in broadcast
NET_DVR_ERR_LANENUM_EXCEED	1400	Over the number of lanes.
NET_DVR_ERR_PRAREA_EXCEED	1401	Recognition area is too large.
NET_DVR_ERR_LIGHT_PARAM	1402	Signal lamp access parameters error.
NET_DVR_ERR_LANE_LINE_INVALID	1403	Lane configuration error.
NET_DVR_ERR_STOP_LINE_INVALID	1404	Stop line configuration error.
NET_DVR_ERR_LEFTORRIGHT_LINE_ INVALID	1405	Turn left / right boundary configuration error.
NET_DVR_ERR_LANE_NO_REPEAT	1406	Overlay lane number repetition.

Error Name	Error Code	Error Description
NET_DVR_ERR_PRAREA_INVALID	1407	The polygon does not meet the requirements.
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	Video detection of traffic light signal exceeds the maximum number of.
NET_DVR_ERR_SUBLIGHT_NUM_ INVALID	1409	Video detection of traffic signal lamp lights are not legitimate
NET_DVR_ERR_LIGHT_AREASIZE_ INVALID	1410	The size of the video detection of traffic light input signal lamp is not valid.
NET_DVR_ERR_LIGHT_COLOR_ INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_ INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABLITY	1413	Lack of IO ablity.
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_ WRITE	1417	Setting param flash write error.
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_ TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.

Error Name	Error Code	Error Description
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_ NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_ COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.
NET_DVR_ERR_RADAR_ANGLE	1434	Radar angle error.
NET_DVR_ERR_RADAR_SPEED_ VALID_TIME	1435	Radar speed valid time error.
NET_DVR_ERR_RADAR_LINE_ CORRECT	1436	Radar line correct error.
NET_DVR_ERR_RADAR_CONST_ CORRECT	1437	Radar const correct error.
NET_DVR_ERR_RECORD_PARAM	1438	Record param error.
NET_DVR_ERR_LIGHT_WITHOUT_ COLOR_AND_DIRECTION	1439	Light number and other param error.
NET_DVR_ERR_LIGHT_WITHOUT_ DETECTION_REGION	1440	Light number and detection region error.
NET_DVR_ERR_RECOGNIZE_ PROVINCE_PARAM	1441	Plate recognize Province param error.
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO Speed TimeOut Param error.
NET_DVR_ERR_NTP_TIMEZONE	1443	NTP TimeZone Param error.
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	NTP Interval Time error.
NET_DVR_ERR_NETWORK_CARD_ NUM	1445	Network Card Num error.
NET_DVR_ERR_DEFAULT_ROUTE	1446	Default Route error.
NET_DVR_ERR_BONDING_WORK_ MODE	1447	Banding Work Mode error.
NET_DVR_ERR_SLAVE_CARD	1448	Slave Card error.
NET_DVR_ERR_PRIMARY_CARD	1449	Primary Card error.

Error Name	Error Code	Error Description
NET_DVR_ERR_DHCP_PPOE_WORK	1450	DHCP and PPOE not Meanwhile start.
NET_DVR_ERR_NET_INTERFACE	1451	Net Interface invalid.
NET_DVR_ERR_MTU	1452	Invalid MTU parameters.
NET_DVR_ERR_NETMASK	1453	Netmask address invalid.
NET_DVR_ERR_IP_INVALID	1454	IP address invalid.
NET_DVR_ERR_MULTICAST_IP_ INVALID	1455	Multicast IP address invalid.
NET_DVR_ERR_GATEWAY_INVALID	1456	Gateway address invalid.
NET_DVR_ERR_DNS_INVALID	1457	DNS Param invalid.
NET_DVR_ERR_ALARMHOST_IP_ INVALID	1458	AlarmHost IP invalid.
NET_DVR_ERR_IP_CONFLICT	1459	IP address Conflict.
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP not support Multi Network segment.
NET_DVR_ERR_NETPORT	1461	NetPort error.
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPoE is not supported.
NET_DVR_ERR_DOMAINNAME_ NOSUPPORT	1463	Not Support Domain Name.
NET_DVR_ERR_NO_SPEED	1464	Speed Not Enabled.
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO Status invalid.
NET_DVR_ERR_BURST_INTERVAL_ INVALID	1466	Burst Interval invalid.
NET_DVR_ERR_RESERVE_MODE	1467	Reserve Mode invalid.
NET_DVR_ERR_LANE_NO	1468	Lane No error.
NET_DVR_ERR_COIL_AREA_TYPE	1469	Coil Area Type error.
NET_DVR_ERR_TRIGGER_AREA_ PARAM	1470	Trigger Area Param error.
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	Speed Limit Param error.
NET_DVR_ERR_LANE_PROTOCOL_ TYPE	1472	Lane Protocol Type error.
NET_DVR_ERR_INTERVAL_TYPE	1473	Capture Interval Type error.

Error Name	Error Code	Error Description
NET_DVR_ERR_INTERVAL_DISTANCE	1474	Capture Interval Distance error.
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	Rs485 Associate DevType error.
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	Rs485 Associate LaneNo error.
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	LaneNo Associate MulitRs485 error.
NET_DVR_ERR_LIGHT_DETECTION_REGION	1478	Light Detection Region error.
NET_DVR_ERR_DN2D_NOSUPPORT	1479	UnSupport Capture Frame 2D Noise Reduction.
NET_DVR_ERR_IRISMODE_NOSUPPORT	1480	UnSupport scene Mode.
NET_DVR_ERR_WB_NOSUPPORT	1481	UnSupport White Balance Mode.
NET_DVR_ERR_IO_EFFECTIVENESS	1482	IO Effectiveness invalid.
NET_DVR_ERR_LIGHTNO_MAX	1483	Access Detector Lights Red / Yellow Overrun.
NET_DVR_ERR_LIGHTNO_CONFLICT	1484	Access Detector Lights Red / Yellow Conflict.
NET_DVR_ERR_CANCEL_LINE	1485	Trigger straight line error.
NET_DVR_ERR_STOP_LINE	1486	Subject line area stop line error.
NET_DVR_ERR_RUSH_REDLIGHT_LINE	1487	Red light trigger lines error.
NET_DVR_ERR_IOOUTNO_MAX	1488	IO out port error.
NET_DVR_ERR_IOOUTNO_AHEADTIME_MAX	1489	IO out ahead time error.
NET_DVR_ERR_IOOUTNO_IOWORKTIME	1490	IO out inwork time error.
NET_DVR_ERR_IOOUTNO_FREQMULTI	1491	IO out frequency multiplication error.
NET_DVR_ERR_IOOUTNO_DUTYRATE	1492	IO out duty rate error.
NET_DVR_ERR_VIDEO_WITH_EXPOSURE	1493	IO out work mode error.

Error Name	Error Code	Error Description
NET_DVR_ERR_PLATE_BRIGHTNESS_WITHOUT_FLASHDET	1494	Plate enable in plate compensate mode on.
NET_DVR_ERR_RECOGNIZE_TYPE_PARAM	1495	Recognize Type error.
NET_DVR_ERR_PALTE_RECOGNIZE_AREA_PARAM	1496	Plate Recognize Area Param error.
NET_DVR_ERR_PORT_CONFLICT	1497	Port Conflict.
NET_DVR_ERR_LOOP_IP	1498	IP cannot be the loopback address.
NET_DVR_ERR_DRIVELINE_SENSITIVE	1499	Driveline sensitivity error.
NET_ERR_VQD_TIME_CONFLICT	1500	The time period conflict.
NET_ERR_VQD_PLAN_NO_EXIST	1501	The diagnostic plan of VQD dese not exist.
NET_ERR_VQD_CHAN_NO_EXIST	1502	The channel dese not exist.
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.

Error Name	Error Code	Error Description
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.
NET_DVR_ERR_CARSIGNSPEED_GREATERTHAN_LIMITSPEED	1613	Car sign speed greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_LIMITSPEED	1614	Big car sign speed limit greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_CARSIGNSPEED	1615	Big car sign speed limit is greater than the car sign speed limit value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_CARLIMITSPEED	1616	Big car speed limit value is greater than the car speed limit value.
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATERTHAN_CARLOWSPEEDLIMIT	1617	Big car low speed limit value is greater than the car low speed limit value.
NET_DVR_ERR_CARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1618	Car speed limit greater than exception high speed value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1619	Big car speed limit greater than exception high speed value.
NET_DVR_ERR_STOPLINE_MORETHAN_TRIGGERLINE	1620	Stopping more than straight lines trigger lines.
NET_ERR_TIME_OVERLAP	1900	Time periods overlap
NET_ERR_HOLIDAY_PLAN_OVERLAP	1901	Holiday plan overlap
NET_ERR_CARDNO_NOT_SORT	1902	Card number is not sorted
NET_ERR_CARDNO_NOT_EXIST	1903	Card number does not exist
NET_ERR_ILLEGAL_CARDNO	1904	Card number error
NET_ERR_ZONE_ALARM	1905	Arming region is in arming status (parameter cannot be modified)

Error Name	Error Code	Error Description
NET_ERR_ZONE_OPERATION_NOT_SUPPORT	1906	Arming region does not support the operation
NET_ERR_INTERLOCK_ANTI_CONFLICT	1907	Interlock and anti-passback configuration conflict
NET_ERR_DEVICE_CARD_FULL	1908	Card full (return after card reached 10,000)
NET_ERR_HOLIDAY_GROUP_DOWNLOAD	1909	Failed to download holiday group
NET_ERR_LOCAL_CONTROL_OFF	1910	Distributed access controller offline
NET_ERR_LOCAL_CONTROL_DISADD	1911	Distributed access controller is not added
NET_ERR_LOCAL_CONTROL_HASADD	1912	Distributed access controller is added
NET_ERR_LOCAL_CONTROL_DOORNO_CONFLICT	1913	Conflict with added distributed access controller
NET_ERR_LOCAL_CONTROL_COMMUNICATION_FAIL	1914	Distributed access controller communication failed
NET_ERR_OPERAND_INEXISTENCE	1915	Operation object does not exist (operation to door, alarm output, alarm input, return when the object is not added)
NET_ERR_LOCAL_CONTROL_OVER_LIMIT	1916	Distributed access controller exceeded device capability upper limit
NET_ERR_DOOR_OVER_LIMIT	1917	Door exceeded device capability upper limit
NET_ERR_ALARM_OVER_LIMIT	1918	Alarm input and output exceeded device capability upper limit
NET_ERR_LOCAL_CONTROL_ADDRESS_INCONFORMITY_TYPE	1919	Distributed access controller address does not match with type
NET_ERR_NOT_SUPPORT_ONE_MORE_CARD	1920	not support one person multi-card
NET_ERR_DELETE_NO_EXISTENCE_FACE	1921	The face picture does not exist.
NET_ERR_DOOR_SPECIAL_PASSWORD_REPEAT	1922	Repeated door door duress code, the super password, or the dismiss code.

Error Name	Error Code	Error Description
NET_ERR_AUTH_CODE_REPEAT	1923	Repeated device authentication code
NET_ERR_DEPLOY_EXCEED_MAX	1924	No more devices can be armed.
NET_ERR_NOT_SUPPORT_DEL_FP_BY_ID	1925	The fingerprint module does not support deleting fingerprint by finger ID.
NET_ERR_TIME_RANGE	1926	Invalid range of the effective period.
NET_ERR_CAPTURE_TIMEOUT	1927	Collection timed out.
NET_ERR_LOW_SCORE	1928	Low quality of collected data.
NET_ERR_OFFLINE_CAPTURING	1929	The device is collecting data offline and cannot respond.
NET_DVR_ERR_OUTDOOR_COMMUNICATION	1950	Communication exception with outdoor terminal
NET_DVR_ERR_ROOMNO_UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringing
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth

Error Name	Error Code	Error Description
NET_DVR_ERR_CONFLICT_TO_ LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.
NET_ERR_ATM_VCA_CHAN_IS_ RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_ OVERLAPED	2113	The VCA channel is already overlayed.
NET_ERR _FACE_CHAN_UNOVERLAP_ EACH_OTHER	2114	The face channels cannot be overlayed.
NET_DVR_SMD_ENCODING_ NORESOURSE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_ NORESOURSE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_ PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_ DIFFRENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_ PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_ WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_ WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number

Error Name	Error Code	Error Description
NET_DVR_ILLEGAL_CERITIFICATE_NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported
NET_DVR_ERR_CLUSTER_DEVICE_TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough
NET_DVR_ERR_CLUSTER_DEL_DEVICE_CM_PLAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity

Error Name	Error Code	Error Description
		for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.

Error Name	Error Code	Error Description
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.
NET_DVR_IOT_CHANNEL_DEVICE_NOT_EXIST	2187	Device of the IoT channel does not exist.
NET_DVR_INVALID_IOT_PROTOCOL_TYPE	2188	Invalid IoT protocol type
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD__FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.

Error Name	Error Code	Error Description
NET_DVR_ERR_NOTSUPPORT_ DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).
NET_DVR_ERR_THERMENABLE_ CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_ BASICPARAM is not turned on)
NET_DVR_ERR_PANORAMIC_LIMIT_ OPERATED	3004	Panoramic map and limit cannot be operated at same time
NET_DVR_ERR_SMARTH264_ROI_ OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_ OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_ OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_ CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_ REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_ HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_ NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_ RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error
NET_DVR_RTSP_ERROR_NO_URL	404	The assigned URL does not exist: when the server returns 404, SDK turns to this error code. E.g. the channel is not available, or the channel does not support sub stream
NET_DVR_RTSP_ERROR_FORCE_STOP	406	The user forces to exit midway
NET_DVR_RTSP_GETPORTFAILED	407	RTSP port getting error.
NET_DVR_RTSP_DESCRIBERROR	410	RTSP DECRIBE communicate error
NET_DVR_RTSP_ DESCRIBESENDTIMEOUT	411	Sending "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERROR	412	Failed to send "RTSP DECRIBE".
NET_DVR_RTSP_ DESCRIBERECVTIMEOUT	413	Receiving "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_ DESCRIBERECVDATALOST	414	Receiving data of "RTSP DECRIBE" error.
NET_DVR_RTSP_DESCRIBERECVERROR	415	Failed to receive "RTSP DECRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DECRIBE, the device returns the error code: 501 (failed to allocate the resource in the device)
NET_DVR_RTSP_SETUPERROR	420	(or 419), RTSP SETUP interaction error. Generally, it is that the address(URL) returned by the device is not accessible, or it is rejected by the server
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDERROR	422	Sending "RTSP SETUP" error.

Error Name	Error Code	Error Description
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECVDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVERROR	425	Failed to receive "RTSP SETUP".
NET_DVR_RTSP_OVER_MAX_CHAN	426	"RTSP SETUP" device returns the error that values 401 or 501. It exceeds the max connection number.
NET_DVR_RTSP_PLAYERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERROR	432	Sending "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	Receiving "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYRECVDATALOST	434	Receiving data of "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVERROR	435	Failed to receive "RTSP PLAY".
NET_DVR_RTSP_PLAYSERVERERR	436	"RTSP PLAY" device returns the error that values 401 or 501.
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN interaction error.
NET_DVR_RTSP_ TEARDOWNSENDTIMEOUT	441	Sending "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_ TEARDOWNSENDERROR	442	Sending "RTSP TEARDOWN" error.
NET_DVR_RTSP_ TEARDOWNRECVTIMEOUT	443	Receiving "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_ TEARDOWNRECVDATALOST	444	Receiving data of "RTSP TEARDOWN" error.
NET_DVR_RTSP_ TEARDOWNRECVERROR	445	Failed to receive "RTSP TEARDOWN".
NET_DVR_RTSP_ TEARDOWNSERVERERR	446	"RTSP TEARDOWN" device returns the error that values 401 or 501.

Software Decoding Library Related Errors

Error Name	Error Code	Error Description
NET_PLAYM4_NOERROR	500	No error.
NET_PLAYM4_PARA_OVER	501	Input parameter is invalid.
NET_PLAYM4_ORDER_ERROR	502	API calling order error.
NET_PLAYM4_TIMER_ERROR	503	Failed to create multimedia clock.
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.
NET_PLAYM4_DEC_AUDIO_ERROR	505	Failed to decode audio data.
NET_PLAYM4_ALLOC_MEMORY_ ERROR	506	Failed to allocate memory.
NET_PLAYM4_OPEN_FILE_ERROR	507	Failed to open the file.
NET_PLAYM4_CREATE_OBJ_ERROR	508	Failed to create thread event.
NET_PLAYM4_CREATE_DDRAW_ ERROR	509	Failed to create DirectDraw object.
NET_PLAYM4_CREATE_OFFSCREEN_ ERROR	510	Failed to create backstage cache for OFFSCREEN mode.
NET_PLAYM4_BUF_OVER	511	Buffer overflow, failed to input stream.
NET_PLAYM4_CREATE_SOUND_ ERROR	512	Failed to create audio equipment.
NET_PLAYM4_SET_VOLUME_ ERROR	513	Failed to set the volume.
NET_PLAYM4_SUPPORT_FILE_ONLY	514	This API can be called only for file playback mode.
NET_PLAYM4_SUPPORT_STREAM_ ONLY	515	This API can be called only when playing stream.
NET_PLAYM4_SYS_NOT_SUPPORT	516	Not support by the system. Decoder can only work on the system above Pentium 3.
NET_PLAYM4_FILEHEADER_ UNKNOWN	517	There is no file header.
NET_PLAYM4_VERSION_ INCORRECT	518	The version mismatch between decoder and encoder.

Error Name	Error Code	Error Description
NET_PLAYM4_INIT_DECODER_ERROR	519	Failed to initialize the decoder.
NET_PLAYM4_CHECK_FILE_ERROR	520	The file is too short, or the stream data is unknown.
NET_PLAYM4_INIT_TIMER_ERROR	521	Failed to initialize multimedia clock.
NET_PLAYM4_BLT_ERROR	522	BLT failure.
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	Failed to open video & audio stream file.
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	Failed to open video stream file.
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG compression error.
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	Don't support the version of this file.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	Extract video data failed.

Container Format Conversion Library Related Errors

Error Name	Error Code	Error Description
NET_CONVERT_ERROR_NOT_SUPPORT	581	This container format is not supported.

Two Way Audio Library Related Errors

Error Name	Error Code	Error Description
NET_AUDIOINTERCOM_OK	600	No error.
NET_AUDIOINTECOM_ERR_NOTSUPORT	601	Not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMERY	602	Memory allocation error.
NET_AUDIOINTECOM_ERR_PARAMETER	603	Parameter error.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	API calling order error.
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	No audio device
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	Failed to open the audio device

Error Name	Error Code	Error Description
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	Context error.
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV file error.
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	The type of WAV parameter is invalid
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	Failed to encode data
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	Failed to decode data
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	Failed to play audio
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	Failed to denoise
NET_AUDIOINTECOM_ERR_UNKOWN	619	Unknown

QoS Stream Control Library Related Errors

Error Name	Error Code	Error Description
NET_QOS_ERR_SCHEDPARAMS_BAD_ MINIMUM_INTERVAL	678	Incorrect predefined minimum interval.
NET_QOS_ERR_SCHEDPARAMS_BAD_ FRACTION	679	Incorrect predefined score.
NET_QOS_ERR_SCHEDPARAMS_INVALID_ BANDWIDTH	680	Invalid predefined bandwidth.
NET_QOS_ERR_PACKET_TOO_BIG	687	The packet size is too large.
NET_QOS_ERR_PACKET_LENGTH	688	Invalid packet size.
NET_QOS_ERR_PACKET_VERSION	689	Incorrect packet versio information.
NET_QOS_ERR_PACKET_UNKNOW	690	Unknown packet.
NET_QOS_ERR_OUTOFMEM	695	Out of memory.
NET_QOS_ERR_LIB_NOT_INITIALIZED	696	The library is not initialized.
NET_QOS_ERR_SESSION_NOT_FOUND	697	No session found.
NET_QOS_ERR_INVALID_ARGUMENTS	698	Invalid parameters.
NET_QOS_ERROR	699	QoS Stream Control Library error.
NET_QOS_OK	700	No error.

NPQ (Network Protocol Quality) Related Error

Error Name	Error Code	Error Description
NET_ERR_NPQ_PARAM	8001	NPQ library: Incorrect parameter.
NET_ERR_NPQ_SYSTEM	8002	NPQ library: Operating system error.
NET_ERR_NPQ_GENRAL	8003	NPQ library: Internal error.
NET_ERR_NPQ_PRECONDITION	8004	NPQ library: Calling sequence error.
NET_ERR_NPQ_NOTSUPPORT	8005	NPQ library: This function is not supported.
NET_ERR_NPQ_NOTCALLBACK	8100	No data is called back.
NET_ERR_NPQ_LOADLIB	8101	Loading NPQ library failed.
NET_ERR_NPQ_STEAM_CLOSE	8104	The NPQ function of this stream is not enabled.
NET_ERR_NPQ_MAX_LINK	8110	No more streaming channel's NPQ function can be enabled.
NET_ERR_NPQ_STREAM_CFG_ CONFLICT	8111	The configured encoding parameters conflicted.

A.7 Response Codes of Text Protocol

The response codes returned during the text protocol integration is based on the status codes of HTTP. 7 kinds of status codes are predefined, including 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid Message Format), 6 (Invalid Message Content), and 7 (Reboot Required). Each kind of status code contains multiple sub status codes, and the response codes are in a one-to-one correspondence with the sub status codes.

StatusCode=1

SubStatusCode	Error Code	Description
ok	0x1	Operation completed.
riskPassword	0x10000002	Risky password.
armProcess	0x10000005	Arming process.

StatusCode=2

Sub Status Code	Error Code	Description
noMemory	0x20000001	Insufficient memory.
serviceUnavailable	0x20000002	The service is not available.
upgrading	0x20000003	Upgrading.
deviceBusy	0x20000004	The device is busy or no response.
reConnectIpc	0x20000005	The video server is reconnected.
transferUpgradePackageFailed	0x20000006	Transmitting device upgrade data failed.
startUpgradeFailed	0x20000007	Starting upgrading device failed.
getUpgradeProcessfailed.	0x20000008	Getting upgrade status failed.
certificateExist	0x2000000B	The Authentication certificate already exists.

StatusCode=3

Sub Status Code	Error Code	Description
deviceError	0x30000001	Hardware error.
badFlash	0x30000002	Flash operation error.
28181Uninitialized	0x30000003	The 28181 configuration is not initialized.
socketConnectError	0x30000005	Connecting to socket failed.
receiveError	0x30000007	Receive response message failed.
deletePictureError	0x3000000A	Deleting picture failed.
pictureSizeExceedLimit	0x3000000C	Too large picture size.
clearCacheError	0x3000000D	Clearing cache failed.
updateDatabasError	0x3000000F	Updating database failed.
searchDatabaseError	0x30000010	Searching in the database failed.

Sub Status Code	Error Code	Description
writeDatabaseError	0x30000011	Writing to database failed.
deleteDatabaseError	0x30000012	Deleting database element failed.
searchDatabaseElementError	0x30000013	Getting number of database elements failed.
cloudAutoUpgradeException	0x30000016	Downloading upgrade packet from cloud and upgrading failed.
HBPException	0x30001000	HBP exception.
UDEPException	0x30001001	UDEP exception
elasticSearchException	0x30001002	Elastic exception.
kafkaException	0x30001003	Kafka exception.
HBaseException	0x30001004	Hbase exception.
sparkException	0x30001005	Spark exception.
yarnException	0x30001006	Yarn exception.
cacheException	0x30001007	Cache exception.
trafficException	0x30001008	Monitoring point big data server exception.
faceException	0x30001009	Human face big data server exception.
SSDFileSystemIsError	0x30001013	SSD file system error (Error occurs when it is non-Ext4 file system)
insufficientSSDCapacityForFPD	0x30001014	Insufficient SSD space for person frequency detection
wifiException	0x3000100A	Wi-Fi big data server exception
structException	0x3000100D	Video parameters structure server exception.
calibrationTimeout	0x30002051	Calibration timed out.
captureTimeout	0x30006000	Data collection timed out.
lowScore	0x30006001	Low quality of collected data.
uploadingFailed	0x30007004	Uploading failed.

StatusCode=4

Sub Status Code	Error Code	Description
notSupport	0x40000001	Not supported.
lowPrivilege	0x40000002	No permission.
badAuthorization	0x40000003	Authentication failed.
methodNotAllowed	0x40000004	Invalid HTTP method.
notSetHdiskRedund	0x40000005	Setting spare HDD failed.
invalidOperation	0x40000006	Invalid operation.
notActivated	0x40000007	Inactivated.
hasActivated	0x40000008	Activated.
certificateAlreadyExist	0x40000009	The certificate already exists.
operateFailed	0x4000000F	Operation failed.
USBNotExist	0x40000010	USB device is not connected.
upgradePackageMorethan2GB	0x40001000	Up to 2GB upgrade package is allowed to be uploaded.
IDNotexist	0x40001001	The ID does not exist.
synchronizationError	0x40001003	Synchronization failed.
synchronizing	0x40001004	Synchronizing.
importError	0x40001005	Importing failed.
importing	0x40001006	Importing.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAlloweLog	0x40001009	Accessing to backup node is not allowed.
exportingError	0x4000100A	Exporting failed.
exporting	0x4000100B	Exporting.
exportEnded	0x4000100C	Exporting stopped.
exported	0x4000100D	Exported.
IPOccupied	0x4000100E	The IP address is already occupied.
IDAlreadyExists	0x4000100F	The ID already exists.

Sub Status Code	Error Code	Description
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoEror	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.

Sub Status Code	Error Code	Description
deviceMOdelMismatched	0x40001027	Adding failed. Device model mismatches.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	No valid data analysis server.
unitNameDuplicate	0x4000102A	Duplicated data analysis server name.
deleteUnitFirst	0x4000102B	Delete the added data analysis server of the node first.
getLocalInfoFailed	0x4000102C	Getting the server information failed.
getClientAddedNodeFailed	0x4000102D	Getting the added node information of data analysis server failed.
taskExit	0x4000102E	The task already exists.
taskInitError	0x4000102F	Initializing task failed.
taskSubmitError	0x40001030	Submiting task failed.
taskDelError	0x40001031	Deleting task failed.
taskPauseError	0x40001032	Pausing task failed.
taskContinueError	0x40001033	Starting task failed.
taskSeverNoCfg	0x40001035	Full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The picture server is not configured.
taskStreamError	0x40001037	Streaming information exception.
taskRecSDK	0x40001038	History recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Invalid VCA rule.
taskNoRun	0x4000103C	The task is not executed.
unitLinksNoStorageNode	0x4000103D	No node is linked with the data analysis server. Configure the node first.
searchFailed	0x4000103E	Searching video files failed.
searchNull	0x4000103F	No video clip.
userScheOffline	0x40001040	The task scheduler service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type mismatches.

Sub Status Code	Error Code	Description
userExist	0x40001043	The user already exists.
userCannotDelAdmin	0x40001044	The administrator cannot be deleted.
userInexistence	0x40001045	The user name does not exist.
userCannotCreatAdmin	0x40001046	The administrator cannot be created.
monitorCamExceed	0x40001048	Up to 3000 cameras can be added.
monitorCunitOverLimit	0x40001049	Adding failed. Up to 5 lower-levels are supported by the control center.
monitorReginOverLimit	0x4000104A	Adding failed. Up to 5 lower-levels are supported by the area.
monitorArming	0x4000104B	The camera is already armed. Disarm the camera and try again.
monitorSyncCfgNotSet	0x4000104C	The system parameters are not configured.
monitorFdSyncing	0x4000104E	Synchronizing. Try again after completing the synchronization.
monitorParseFailed	0x4000104F	Parsing camera information failed.
monitorCreatRootFailed	0x40001050	Creating resource node failed.
deleteArmingInfo	0x40001051	The camera is already . Disarm the camera and try again.
cannotModify	0x40001052	Editing is not allowed. Select again.
cannotDel	0x40001053	Deletion is not allowed. Select again.
deviceExist	0x40001054	The device already exists.
IPErrorConnectFailed	0x40001056	Connection failed. Check the network port.
cannotAdd	0x40001057	Only the capture cameras can be added.
serverExist	0x40001058	The server already exists.
fullTextParamError	0x40001059	Incorrect full-text search parameters.
storParamError	0x4000105A	Incorrect storage server parameters.
picServerFull	0x4000105B	The storage space of picture storage server is full.

Sub Status Code	Error Code	Description
NTPUnconnect	0x4000105C	Connecting to NTP server failed. Check the parameters.
storSerConnectFailed	0x4000105D	Connecting to storage server failed. Check the network port.
storSerLoginFailed	0x4000105E	Logging in to storage server failed. Check the user name and password.
searchSerConnectFailed	0x4000105F	Connecting to full-text search server failed. Check the network port.
searchSerLoginFailed	0x40001060	Logging in to full-text search server failed. Check the user name and password.
kafkaConnectFailed	0x40001061	Connecting to Kafka failed. Check the network port.
mgmtConnectFailed	0x40001062	Connecting to system failed. Check the network port.
mgmtLoginFailed	0x40001063	Logging in to system failed. Check the user name and password.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster…Do not operate.
clusterUninstall	0x4000106F	Reseting cluster…Do not operate.
clusterInstall	0x40001070	Creating cluster failed.

Sub Status Code	Error Code	Description
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The master node and slave node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.
clusterNodeUnadd	0x40001074	The specified master(slave) node is not added.
clusterNodeOffline	0x40001075	The task scheduler server is offline.
nodeNotCurrentIP	0x40001076	The analysis node of the current IP address is required when adding master and slave nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.
needTwoMgmtNode	0x40001078	Two management nodes are required when adding master and slave nodes.
ipConflict	0x40001079	The virtual IP address and data analysis server's IP address conflicted.
ipUsed	0x4000107A	The virtual IP address has been occupied.
cloudAlalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSame NetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of master/slave node.
getNodeDispatchInfoFailed	0x4000107D	Getting node scheduler information failed.
unableModifyManagementNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for master and slave cluster.
armingFull	0x40001080	No more device can be armed.
armingNoFind	0x40001081	The arming information does not exist.
disArming	0x40001082	Disarming failed.
getArmingError	0x40001084	Getting arming information failed.
refreshArmingError	0x40001085	Refreshing arming information failed.
ArmingPlateSame	0x40001086	The license plate number is repeatedly armed.
ArmingParseXLSError	0x40001087	Parsing arming information file failed.

Sub Status Code	Error Code	Description
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateAarmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blacklist library can be armed.
noMoreCamerasAdded	0x4000108D	No more camera can be armed.
noMoreArmingListAddedWithCamera	0x4000108E	No more library can be linked to the camera.
noMoreArmingPeriodAdded	0x4000108F	No more time period can be added to the arming schedule.
armingPeriodsOverlapped	0x40001090	The time periods in the arming schedule are overlapped.
noArmingAlarmInfo	0x40001091	The alarm information does not exist.
armingAlarmUnRead	0x40001092	Getting number of unread alarms failed.
getArmingAlarmError	0x40001093	Getting alarm information failed.
searchByPictureTimedOut	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.
comparisonUnfinished	0x4000109A	The comparison is not completed.
facePictureModelInvalid	0x4000109B	Invalid face model.

Sub Status Code	Error Code	Description
duplicateLibraryName.	0x4000109C	The library name already exists.
noRecord	0x4000109D	No record found.
countingRecordsFailed.	0x4000109E	Calculate the number of records failed.
getHumanFaceFrameFailed	0x4000109F	Getting face thumbnail from the picture failed.
modelingFailed.	0x400010A0	Modeling face according to picture URL failed.
1V1FacePictureComparisonFailed	0x400010A1	Comparison 1 VS 1 face picture failed.
libraryArmed	0x400010A2	The blacklist library is armed.
licenseExeedLimit	0x400010A3	Dongle limited.
licenseExpired	0x400010A4	Dongle expired.
licenseDisabled	0x400010A5	Unavailable dongle.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	Session expired .
beyondConcurrentLimit	0x400010A8	Out of concurrent limit.
stopSync	0x400010A9	Synchronization stopped.
getProgressFaild	0x400010AA	Getting progress failed.
uploadExtraCaps	0x400010AB	No more files can be uploaded.
timeRangeError	0x400010AC	Time period error.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Adding to the cluster failed. The device is already added to other cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Searching task failed.
modifyTimeRuleFailed	0x400010B2	The task already exists. Editing time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Editing VAC rule is not allowed.
queryHistoryVideoFaile d	0x400010B4	Searching history video failed.

Sub Status Code	Error Code	Description
addDeviceFailed	0x400010B5	Adding device failed.
addVideoFailed	0x400010B6	Adding video files failed.
deleteAllVideoFailed	0x400010B7	Deleting all video files failed.
createVideoIndexFailed	0x400010B8	Indexing video files failed.
videoCheckTypeFailed	0x400010B9	Verifying video files types failed.
configStructuredAddressFailed	0x400010BA	Configuring IP address of structured server failed.
configPictureServerAddressFailed	0x400010BB	Configuring IP address of picture storaged server failed.
storageServiceIPNotExist	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFailed	0x400010BE	Synchronizing slave database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of slave server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopbacl address is not supported by the master or slave cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blacklists.
blackListQueryExported	0x400010C8	The matched blacklists are exported.
blackListQueryStopExporting	0x400010C9	Exporting matched blacklists is stopped.

Sub Status Code	Error Code	Description
blackListAlarmQueryExporting	0x400010CA	Exporting matched blacklist alarms.
blackListAlarmQueryExported	0x400010CB	The matched blacklists alarms are exported.
blackListAlarmQueryStopExporting	0x400010CC	Exporting matched blacklist alarms is stopped.
getBigDataCloudAnalysisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalysisFailed	0x400010CE	Configuring big data cloud analytic failed.
submitMapSearchFailed	0x400010CF	Submitting search by picture task failed.
controlRelationshipNotExist	0x400010D0	The relation does not exist.
getHistoryAlarmInfoFailed	0x400010D1	Getting history alarm information failed.
getFlowReportFailed	0x400010D2	Getting people counting report failed.
addGuardFailed	0x400010D3	Adding arming configuration failed.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.

Sub Status Code	Error Code	Description
transcodingServerOffline	0x400010DD	Transcoding server is offline.
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sesssionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO 8601TimeFormatError	0x400010ED	Invalid ISO8601 time format.
clusterDissolutionFailed	0x400010EE	Deleting cluster failed.
getServiceNodeInfoFailed	0x400010EF	Getting service node information failed.
getUPSInfoFailed	0x400010F0	Getting UPS configuration information failed.
getDataStatisticsReportFailed	0x400010F1	Getting data statistic report failed.
getDisplayConfigInfoFailed	0x400010F2	Getting display configuration failed.

Sub Status Code	Error Code	Description
namingAnalysisBoardNotAllowed	0x400010F3	Renaming analysis board is not allowed.
onlyDrawRegionsOfConvexPolygon	0x400010F4	Only drawing convex polygon area is supported.
bigDataServerResponseInfoParseFailed	0x400010F5	Parsing response message of big data service failed.
bigDataServerReturnFailed	0x400010F6	No response is returned by big data service.
microVideoReturnFailed	0x400010F7	No response is returned by micro video cloud service.
transcodingServerReturnFailed	0x400010F8	No response is returned by transcoding service.
UPSServerReturnFailed	0x400010F9	No response is returned by UPS monitoring service.
forwardingServer ReturnFailed	0x400010FA	No response is returned by forwarding service.
storageServer ReturnFailed	0x400010FB	No response is returned by storage service.
cloudAnalysisServerReturnFailed	0x400010FC	No response is returned by cloud analytic service.
modelEmpty	0x400010FD	No model is obtained.
mainAndBackupNodeCannotModifyManagementNetworkInterfaceIP	0x400010FE	Editing the management interface IP address of master node and backup node is not allowed.
IDTooLong	0x400010FF	The ID is too long.
pictureCheckFailed	0x40001100	Detecting picture failed.
pictureModelingFailed	0x40001101	Modeling picture failed.
setCloudAnalsisDefaultProvinceFailed	0x40001102	Setting default province of cloud analytic service failed.
InspectionAreasNumberExceedLimit	0x40001103	No more detection regions can be added.
picturePixelsTooLarge	0x40001105	The picture resolution is too high.
picturePixelsTooSmall	0x40001106	The picture resolution is too low.

Sub Status Code	Error Code	Description
storageServiceIPEmpty	0x40001107	The storage server IP address is required.
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysiTimedOut	0x40001109	Analysis time out.
high-performanceModeDisabled.	0x4000110A	Please enable high-performance mode.
configuringUPSMonitoringServerTimedOut	0x4000110B	Configurating the UPS monitoring server time out. Check IP address.
cloudAnalysisRequestInformationBuildFailed	0x4000110C	Creating request message of cloud analytic service failed.
cloudAnalysisResponseInformationParseFailed	0x4000110D	Parsing response message of cloud analytic service failed.
allCloudAnalysisInterfaceFailed	0x4000110E	Calling API for cloud analytic service failed.
cloudAnalysisModelCompareFailed	0x4000110F	Model comparison of cloud analytic service failed.
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.

Sub Status Code	Error Code	Description
numberReachedLlimit	0x40001119	Number reached the limit.
wait1~3MinutesGetIPAfterSetupDHCP	0x4000111A	Wait 1 to 3 minutes to get IP address after configuring DHCP.
plannedTimeMustbeHalfAnHour	0x4000111B	Schedule must be half an hour.
oneDeviceCannotBuildCluster	0x4000111C	Creating master and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodeDoesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.
getVideoSourceTypeFailed	0x4000112A	Getting video source type failed.
smartRulesBuildFailed	0x4000112B	Creating VAC rule failed.
smartRulesParseFailed	0x4000112C	Parsing VAC rule failed.

Sub Status Code	Error Code	Description
timeRulesBuildFailed	0x4000112D	Creating time rule failed.
timeRulesParseFailed	0x4000112E	Parsing time rule failed.
monitoInfoInvalid	0x4000112F	Invalid camera information.
addingFailedVersionMismatches	0x40001130	Adding failed. The device version mismatches.
theInformationReturnedAfterCloudAnalysisIsEmpty	0x40001131	No response is returned by the cloud analytic service.
selectingIpAddressOfHostAndSpareNodeFailedCheckTheStatus	0x40001132	Setting IP address for master node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
theIndexCodeDoesNotExist	0x40001138	The index code does not exist.
theControlCenterIdDoesNotExist	0x40001139	The control center ID does not exist.
theAreaIdDoesNotExist	0x4000113A	The area ID does not exist.
theArmingLinkageIdDoesNotExist	0x4000113C	The arming relationship ID does not exist.
theListLibraryIdDoesNotExist	0x4000113D	The list library ID does not exist.
invalidCityCode	0x4000113E	Invalid city code.
synchronizingThePasswordOfSpareServerFailed	0x4000113F	Synchronizing backup system password failed.
editingStreamingTypeIsNotSupported	0x40001140	Editing streaming type is not supported.

Sub Status Code	Error Code	Description
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching scheduled task to temporary task is not supported.
switchingTemporaryTaskToScheduledTaskIsNotSupported	0x40001142	Switching temporary task to scheduled task is not supported.
theTaskIsNotDispatchedOrItIsUpdating	0x40001143	The task is not dispatched or is updating.
thisTaskDoesNotExist	0x40001144	This task does not exist in the cloud analytic serice.
duplicatedSchedule	0x40001145	Schedule period cannot be overlapped.
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	The continuous schedule periods with same algorithm type should be merged.
invalidStreamingTimeRange	0x40001147	Invalid streaming time period.
invalidListLibraryType	0x40001148	Invalid list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of search results should be larger than 0.
invalidValueRangeOfSimilarity	0x4000114A	Invalid similarity range.
invalidSortingType	0x4000114B	Invalid sorting type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	No more lists can be added to one device.
InvalidRecipientAddressFormat	0x4000114D	Invalid address format of result receiver.
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	Insert the dongle before creating cluster.
theURLIsTooLong	0x4000114F	No schedule configured for the task.
noScheduleIsConfiguredForTheTask	0x40001150	No schedule configured for the task.
theDongleIsExpiried	0x40001151	Dongle has expired.

Sub Status Code	Error Code	Description
dongleException	0x40001152	Dongle exception.
invalidKey	0x40001153	Invalid authorization service key.
decryptionFailed	0x40001154	Decrypting authorization service failed.
encryptionFailed	0x40001155	Encrypting authorization service failed.
AuthorizeServiceResponseError	0x40001156	Authorization service response exception.
incorrectParameter	0x40001157	Authorization service parameters error.
operationFailed	0x40001158	Operating authorization service error.
noAnalysisResourceOrNoDataInTheListLibrary	0x40001159	No cloud analytic resources or no data in the list library.
calculationException	0x4000115A	Calculation exception.
allocatingList	0x4000115B	Allocating list.
thisOperationIsNotSupportedByTheCloudAnalytics	0x4000115C	This operation is not supported by the cloud analytic serice.
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic serice is interrupted.
theServiceIsNotReady	0x4000115E	The service is not ready.
searchingForExternalApiFailed	0x4000115F	Searching external interfaces failed.
noOnlineNode	0x40001160	No node is online.
noNodeAllocated	0x40001161	No allocated node.
noMatchedList	0x40001162	No matched list.
allocatingFailedTooManyFacePictureLists	0x40001163	Allocation failed. Too many lists of big data service.
searchIsNotCompletedSearchAgain	0x40001164	Current searching is not completed. Search again.
allocatingListIsNotCompleted	0x40001165	Allocating list is not completed.
searchingForCloudAnalyticsResultsFailed	0x40001166	Searching cloud analytic serice overtime.

Sub Status Code	Error Code	Description
noDataOfTheCurrentLibraryFound	0x40001167	No data in the current library. Make sure there is data in the Hbase.
noFacePictureLibraryIsArmed	0x40001168	No face picture library is armed for big data service.
noAvailableDataSlicingVersionInformationArmFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicinDataFailedNoArmedFacePictureLibrary	0x4000116B	Slicing failed. No arming information in the face big data.
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	Generating sliced file failed. Slice again.
NonprimaryNodeIsProhibitedFromSlcingData	0x4000116D	Slicing is not allowed by the backup node.
NoReadyNodeToClusterServers	0x4000116E	Creating the cluster failed. No ready node.
NodeManagementServiceIsOffline	0x4000116F	The node management server is offline.
theCamera(s)OfTheControlCenterAreAlreadyArmed.DisarmThemFirst	0x40001170	Some cameras in control center are already armed. Disarm them and try again.
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	Some cameras in this area are already armed. Disarm them and try again.
configuringHigh-frequencyPeopleDetectionFailed	0x40001172	Configuring high frequency people detection failed.
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	Searching detection event logs of high-frequency people detection failed.
gettingDetailsOfSearchedHigh-	0x40001174	Getting the search result details of high frequency alarms failed.

Sub Status Code	Error Code	Description
frequencyPeopleDetectionLogsFailed.		
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCameraIsNotArmed	0x40001177	Disarming failed. The camera is not armed.
noDataReturned	0x40001178	No response is returned by the big data service.
preallocFailure	0x40001179	Pre-allocating algorithm resource failed.
overDogLimit	0x4000117A	Configuration failed. No more resources can be pre-allocated.
analysisServicesDoNotSupport	0x4000117B	Not supported.
commandAndDispatchServiceError	0x4000117C	Scheduling service of cloud analytic serice error.
engineModuleError	0x4000117D	Engine module of cloud analytic serice error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic serice error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic serice error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic serice error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic serice error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic serice error.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForListLibrary	0x40001183	High frequency alarm is already armed for blacklist library.
creatingListLibraryFaile d	0x40001184	Creating list library failed.
invalidIdentiryKeyOfListLibrary	0x40001185	Invalid identity key of list library.

Sub Status Code	Error Code	Description
noMoreDevicesCanBeArmed	0x40001186	No more camera can be added.
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting high frequency alarm failed.
invalidSearchConfition	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The high frequency alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
SMRDiskNotSupportRaid	0x40001291	SMR disk does not support RAID.
OnlySupportHikAndCustomProtocol	0x400012A3	IPv6 camera can only be added via Device Network SDK or custom protocols.
vehicleEnginesNoResource	0x400012A6	Insufficient vehicle engine resources.

Sub Status Code	Error Code	Description
noMoreRunningRulesAdded	0x400012A9	Maximum number of running rules reached.
noMoreGroupRulesAdded	0x400012AA	Maximum number of people gathering rules reached.
noMoreFailDownRulesAdded	0x400012AB	Maximum number of people falling down rules reached.
noMorePlayCellphoneRulesAdded	0x400012AC	Maximum number of playing cellphone rules reached.
ruleEventTypeDuplicate	0x400012C8	Event type duplicated.
noMoreRetentionRulesAdded	0x400015AD	Maximum number of people retention rules reached.
noMoreSleepOnDutyRulesAdded	0x400015AE	Maximum number of sleeping on duty rules reached.
polygonNotAllowCrossing	0x400015C2	Polygons are not allowed to cross.
AITargetBPCaptureFail	0x400019C5	Capturing reference picture for AI target comparison failed.
AITargetBPToDSPFail	0x400019C6	Sending reference picture to DSP for AI target comparison failed.
AITargetBPDuplicateName	0x400019C7	Duplicated name of reference picture for AI target comparison.
audioFileNameWrong	0x400019D0	Incorrect audio file name.
audioFileImportFail	0x400019D1	Importing audio file failed.
alreadyRunning	0x40002026	The application program is running.
notRunning	0x40002027	The application program is stopped.
packNotFound	0x40002028	The software packet does not exist.
alreadyExist	0x40002029	The application program already exists.
noMemory	0x4000202A	Insufficient memory.
invalLicense	0x4000202B	Invalid License.
noClientCertificate	0x40002036	The client certificate is not installed.
noCACertificate	0x40002037	The CA certificate is not installed.

Sub Status Code	Error Code	Description
authenticationFailed	0x40002038	Authenticating certificate failed. Check the certificate.
clientCertificateExpired	0x40002039	The client certificate is expired.
clientCertificateRevocation	0x4000203A	The client certificate is revoked.
CACertificateExpired	0x4000203B	The CA certificate is expired.
CACertificateRevocation	0x4000203C	The CA certificate is revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit	0x4000203F	No more user can log in.
HDMIResolutionIllegal	0x40002040	The HDMI video resolution cannot be larger than that of main and sub stream.
hdFormatFail	0x40002049	Formatting HDD failed.
formattingFailed	0x40002056	Formatting HDD failed.
encryptedFormattingFailed	0x40002057	Formatting encrypted HDD failed.
wrongPassword	0x40002058	Verifying password of SD card failed. Incorrect password.
audioIsPlayingPleaseWait	0x40002067	Audio is playing. Please wait.
twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress. Please wait.
calibrationPointNumFull	0x40002069	The maximum number of calibration points reached.
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.

Sub Status Code	Error Code	Description
HBDLibNumOverLimit	0x40002092	The number of human body picture libraries reaches the upper limit
theShieldRegionError	0x40002093	Saving failed. The shielded area should be the ground area where the shielded object is located.
theDetectionAreaError	0x40002094	Saving failed. The detection area should only cover the ground area.
invalidLaneLine	0x40002096	Saving failed. Invalid lane line.
enableITSFunctionOfThisChannelFirst	0x400020A2	Enable ITS function of this channel first.
noCloudStorageServer	0x400020C5	No cloud storage server
NotSupportWithVideoTask	0x400020F3	This function is not supported.
incorrectConsolePassword	0x40002106	Saving failed. Incorrect console command.
noDetectionArea	0x400050df	No detection area
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarming failed.
clearAlarmFailed	0x40008002	Clearing alarm failed.
bypassFailed	0x40008003	Bypass failed.
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Opening relay failed.
outputsCloseFailed	0x40008006	Closing relay failed.
registerTimeOut	0x40008007	Registering timed out.
registerFailed	0x40008008	Registering failed.
addedByOtherHost	0x40008009	The peripheral is already added by other security control panel.
alreadyAdded	0x4000800A	The peripheral is already added.
armedStatus	0x4000800B	The partition is armed.
bypassStatus	0x4000800C	Bypassed.
zoneNotSupport	0x4000800D	This operation is not supported by the zone.

Sub Status Code	Error Code	Description
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.
masterSlaveIsEnable	0x4000802c	The master-slave relationship has taken effect, the slave radar does not support this operation.
forceTrackNotEnabled	0x4000802d	Mandatory tracking is disabled.
isNotSupportZoneConfigByLocalArea	0x4000802e	This area does not support the zone type.
alarmLineCross	0x4000802f	Trigger lines are overlapped.
zoneDrawingOutOfRange	0x40008030	The drawn zone is out of detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm trigger line is out of detection range.
hasTargetInWarningArea	0x40008032	The warning zone already contains targets. Whether to enable mandatory arming?

Sub Status Code	Error Code	Description
radarMoudleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
noFlash	/	The platform mismatches.

StatusCode=5

Sub Status Code	Error Code	Description
badXmlFormat	0x50000001	Invalid XML format.

StatusCode=6

Sub Status Code	Error Code	Description
badParameters	0x60000001	Invalid parameter.
badHostAddress	0x60000002	Invalid host IP address.
badXmlContent	0x60000003	Invalid XML content.
badIPv4Address	0x60000004	Invalid IPv4 address.

Sub Status Code	Error Code	Description
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
connectSreverFail	0x60000009	Connecting to server failed.
conflictDomainName	0x6000000A	Domain name conflicted.
badPort	0x6000000B	Port number conflicted.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	Invalid sub-net mask.
badVersion	0x6000000F	Version mismatches.
badDevType	0x60000010	Device type mismatches.
badLanguage	0x60000011	Language mismatches.
incorrentUserNameOrPassword	0x600000012	Incorrect user name or password.
invalidStoragePoolOfCloudServer	0x600000013	Invalid storage pool. The storage pool is not configured or incorrect ID.
noFreeSpaceOfStoragePool	0x600000014	Storage pool is full.
riskPassword	0x600000015	Risky password.
UnSupportCapture	0x600000016	Capturing in 4096*2160 or 3072*2048 resolution is not supported when H.264+ is enabled.
userPwdLenUnder8	0x60000023	At least two kinds of characters, including digits, letters, and symbols, should be contained in the password.
userPwdNameSame	0x60000025	Duplicated password.
userPwdNameMirror	0x60000026	The password cannot be the reverse order of user name.

Sub Status Code	Error Code	Description
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutofDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library
humanLibraryNumError	0x60000091	No more human body picture library can be added
humanImagesNumError	0x60000092	No more human body picture can be added
noHumanInThePicture	0x60000093	Modeling failed, no human body in the picture
analysisEnginesNoResourceError	0x60001000	No analysis engine.

Sub Status Code	Error Code	Description
analysisEnginesUsageExcced	0x60001001	The engine usage is overloaded.
PicAnalysisNoResourceError	0x60001002	No analysis engine provided for picture secondary recognition.
analysisEnginesLoadingError	0x60001003	Initializing analysis engine.
analysisEnginesAbnormaError	0x60001004	Analysis engine exception.
analysisEnginesFacelibImportin g	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChan nel	0x60001006	The analysis engine is linked to channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	HDD error.
diskFull	0x6000100a	HDD full.
facelibDataProcessing	0x6000100b	Handling face picture library data.
capturePackageFailed	0x6000100c	Capturing packet failed.
capturePackageProcessing	0x6000100d	Capturing packet.
noSupportWithPlaybackAbstra ct	0x6000100e	This function is not supported. Playback by video synopsis is enabled.
insufficientNetworkBandwidth	0x6000100f	Insufficient network bandwidth.
tapeLibNeedStopArchive	0x60001010	Stop the filing operation of tape library first.
identityKeyError	0x60001011	Incorrect interaction command.
identityKeyMissing	0x60001012	The interaction command is lost.

Sub Status Code	Error Code	Description
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChanNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.
collisionTimeOverflow	0x60001022	No more comparison duration can be configured.
invalidTaskID	0x60001023	Invalid task ID.
eventNotSupport	0x60001024	Event subscription is not supported.
invalidEZVIZSecretKey	0x60001034	Invalid verification code for Hik-Connect.
needDoubleVerification	0x60001042	Double verification required

Sub Status Code	Error Code	Description
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
CurrentAudioFileAIRuleInUseAlreadyDelete	0x60001136	The AI rule linkage related to current audio file has been deleted.
TransitionUseEmmc	0x60002000	Starting device failed. The EMMC is overused.
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.
AdaptiveStreamAndVariableBitrateEnabled	0x60002002	Stream self-adptive and variable bitrate function cannot be enabled at the same time.
noSafetyHelmetRegion	0x60002023	The hard hat detection area is not configured (if users save their settings without configuring the arming area, they should be prompted to configure one).

Sub Status Code	Error Code	Description
unclosedSafetyHelmet	0x60002024	The hard hat detection is enabled (If users save their settings after deleting the arming area, they should be prompted to disable hard hat detection first and then delete the arming area).
width/ heightRatioOfPictureError	0x6000202C	The width/height ratio of the uploaded picture should be in the range from 1:2 to 2:1.
PTZNotInitialized	0x6000202E	PTZ is not initialized.
PTZSelfChecking	0x6000202F	PTZ is self-checking.
PTZLocked	0x60002030	PTZ is locked.
advancedParametersError	0x60002031	Auto-switch interval in advanced parameters cannot be shorter than parking tolerance for illegal parking detection in speed dome rule settings.
resolutionError	0x60005003	Invalid resolution
deployExceedMax	0x60006018	The arming connections exceed the maximum number.
detectorTypeMismatch	0x60008000	The detector type mismatched.
nameExist	0x60008001	The name already exists.
uploadImageSizeError	0x60008016	The size of the uploaded picture is larger than 5 MB.
laneAndRegionOverlap	/	The lanes are overlapped.
unitConfigurationNotInEffect	/	Invalid unit parameter.
ruleAndShieldingMaskConflict	/	The line-rule region overlaps with the shielded area.
wholeRuleInShieldingMask	/	There are complete temperature measurement rules in the shielded area.

Sub Status Code	Error Code	Description
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log HDD in the HDD group cannot be set to read-only.
LogDiskNotSetReDundancyInGroupMode	0x60001101	The log HDD in the HDD group cannot be set to redundancy.

StatusCode=7

SubStatusCode	Error Code	Description
rebootRequired	0x70000001	Reboot to take effect.