Perface

Thanks for using our product. we'll do our best to provide the best service for you. This handbook maybe contains technology mistakes or text errors. The content will be updated regularly without prior notice and the new updated part will be added in the new version.

Client SDK Instructions

Summary

Client SDK is a matching product of embedded hard disk video recorder(DVR), network hard disk video recorder(NVR), video server(DVS) and IPCamera, mostly used in remote device access and remote control software development.All functions below is mainly for DVR and NVR, when you integrate IPC with SDK ,just take IPC as one channel DVR device, and IPC SDK instruction page lists which functions are supported for IPC integration. The version of this product is 1.0,and its main function as the list below:

Function Module	Function Details
Device basic info/Parameters config	device serial number,firmware version number,firmware version compiling date,core version number,hardware version number.
	video formats,device ID,device name
	device interface language,start password protection,screen protection,VGA parameters setting
Living preview	local preview,remote preview,channel hidden,picture segmentation,picture polling
	channel name,record state,alarm state(sensor alarm,network address conflict,harddisk error,disk full,system time)
	electronic zoom in,picture in picture
	living sound,volume adjust,mute control
Preview video/Parameters setting	color adjust,adapt color to different time slot
	character overlay:channel

	name,timestamp,user-defined information
	area keep out
Video & Audio record/Parameters settings	video&audio record/parameters settings
	record of alarming:record time before alarming,record time after alarming,different encoding parameters setting
	record switch:video switch,audio switch,video & audio binding relationship
	data expiration,redundancy record,group record,whether circulation cover
	record plan:record on time, record when sensor alarming,record when motion detection,record when video kept out alarming
	manual record,remote control of manual record
Capture/Parameters setting	picture measurement,picture quality,capture interval, capture numbers
	manual capture,picture retrieval,picture diaplay
Alarm handle/Parameters setting	alarm type:sensor alarm,motion detection alarm,video kept out alarm,video lost alarm,smart analytics alarm
	detection plant, detection time-lag,sensor device type,sensor device name, motion detection area & sensitivity
	alarm output:relay alarm,buzzer alarm,large picture caution,send

	emails,send up to center
	alarm record:trigger record(specify channel),record log(alarm starts information,alarm ends information)
	action with alarm:PTZ preset positions,PTZ cruise line, PTZ track
	alarm from email:channel information,alarm type,attached picture(specify channel,picture numbers & time interval)
Alarm output/Parameters setting	alarm from relay:switch,response schedule,response time-lag,alarm name
	alarm from buzzer:switch,response schedule,response time-lag
	manual alarm,remotely manual alarm
Past video retrieval/Playback/Backup	search by time:data distribution charter, distinguish different record types
	search by event:event list, filter event type
	search by file :file list(no partitioning record segment),lock/delete file
	search by picture:picture diaplaydisplay, locked/delete pictures, picture backup(save as)
	start playback:specify the start time(specify a group of channels) specify event,specify file
	playback control:stop ,speed,fast backward,single frame play,reposition,exist playback,store capture(harddisk),select area for backup

	manual backup:backup by start time & stop time,backup by the specified file,montage backup,DVR & AVI formats,combination backup(multi-channels backup in one file)
	automatically backup:specify time and condition,backup in external memory,backup in network service
Network/Parameters setting	network address setting:static address,dynamic address,PPPoE
	ports-settings:HTTP port,data port,alarm port,etc.
	multicast address setting
	DDNS setting:customization demand,send up the period setting,etc.
	parameters setting of network substream encoding:resolution,frame rate,encoded mode,picture quality,code stream limit,whether self-adaption(picture quality & fluency)
	network linking setting:register user number limit,video channel number limit,whether release mainstream,black and white list
	check network state,prompt network state live(normal connection,no connection,conflict),check users online,push-off users online
Mail functionary/Config	send emails when alarm,combine emails in a while,emails whether with attachment,specify to send emails, send emails manually

PTZ control/Parameters setting	parameters setting of serial port ,preset position setting,cruise line setting,track setting
	PTZ control:eight directions,stop,aperture,focus,zoom in,rate(128),lamplight,windshield wiper,automatically scan line
	control mode:mouse 3D control,mouse control through dialog box,front panel,remote-controller,professional keyboard,remote control
	protocol
Configuration management	local configure,remote configuration,configuration import & export,recover default configuration
Disc management/Health check	format disc,delete data,specify disc group,set disc attribute(read only, read-write,redundancy,backup)
System maintain	firmware upgrade,device health checked, remote upgrade,FTP upgrade
Other function	user management,permission setting
	log record/retrieval/export
	FTP setting

Client SDK Instructions

IPC SDK instruction

All supported functions for IPC:live preview, capture, sensor alarm, motion alarm, config import and export, getting or setting config, getting device time.

supported config for IPC:

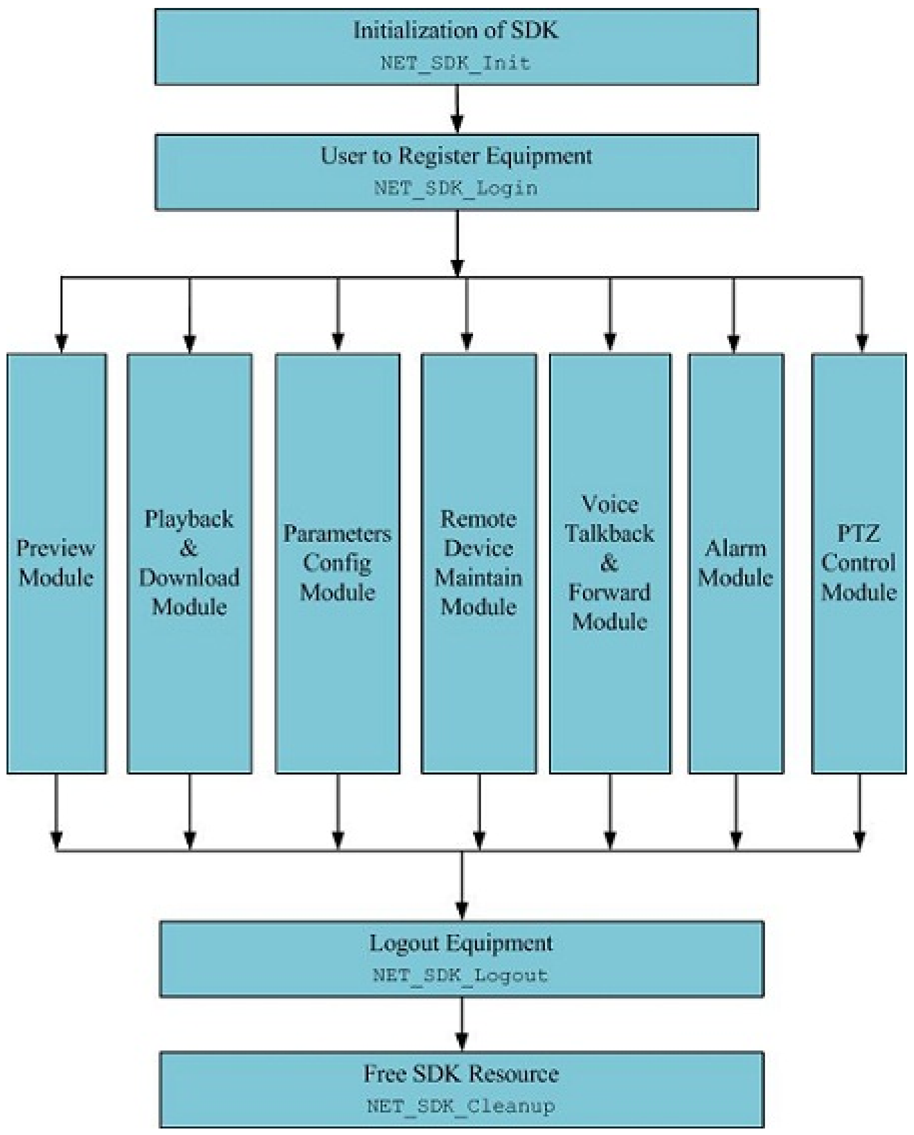
- DD_CONFIG_ITEM_DEVICE_INFO
- DD_CONFIG_ITEM_SYSTEM_BASIC
- DD_CONFIG_ITEM_DATE_TIME
- DD_CONFIG_ITEM_DAYLIGHT_INFO
- DD_CONFIG_ITEM_NETWORK_IP
- DD_CONFIG_ITEM_NETWORK_ADVANCE
- DD_CONFIG_ITEM_DDNS_SERVER_INFO
- DD_CONFIG_ITEM_ACCOUNT
- DD_CONFIG_ITEM_SENSOR_SETUP
- DD_CONFIG_ITEM_SENSOR_SCHEDULE
- DD_CONFIG_ITEM_SENSOR_ALARM_OUT
- DD_CONFIG_ITEM_SENSOR_TO_RECORD
- DD_CONFIG_ITEM_MOTION_SETUP
- DD_CONFIG_ITEM_MOTION_SCHEDULE
- DD_CONFIG_ITEM_MOTION_ALARM_OUT
- DD_CONFIG_ITEM_RELAY_SETUP
- DD_CONFIG_ITEM_NETWORK_SMTP
- DD_CONFIG_ITEM_PTZ_PRESET
- DD_CONFIG_ITEM_PTZ_SETUP
- DD_CONFIG_ITEM_VIDEO_COLOR
- DD_CONFIG_ITEM_ENCODE_MASK_MAJOR
- DD_CONFIG_ITEM_ENCODE_MASK_MINOR
- DD_CONFIG_ITEM_ENCODE_SCHEDULE
- DD_CONFIG_ITEM_ENCODE_NETWORK

Programming Guide

This section mainly introduces functions in SDK with flow charts and text brief description.

Client SDK Instructions

main flow of calling SDK interface



It has seven function modules,each function module has four necessary the same flow:initialize SDK,user to login,user to logout and free SDK resource.

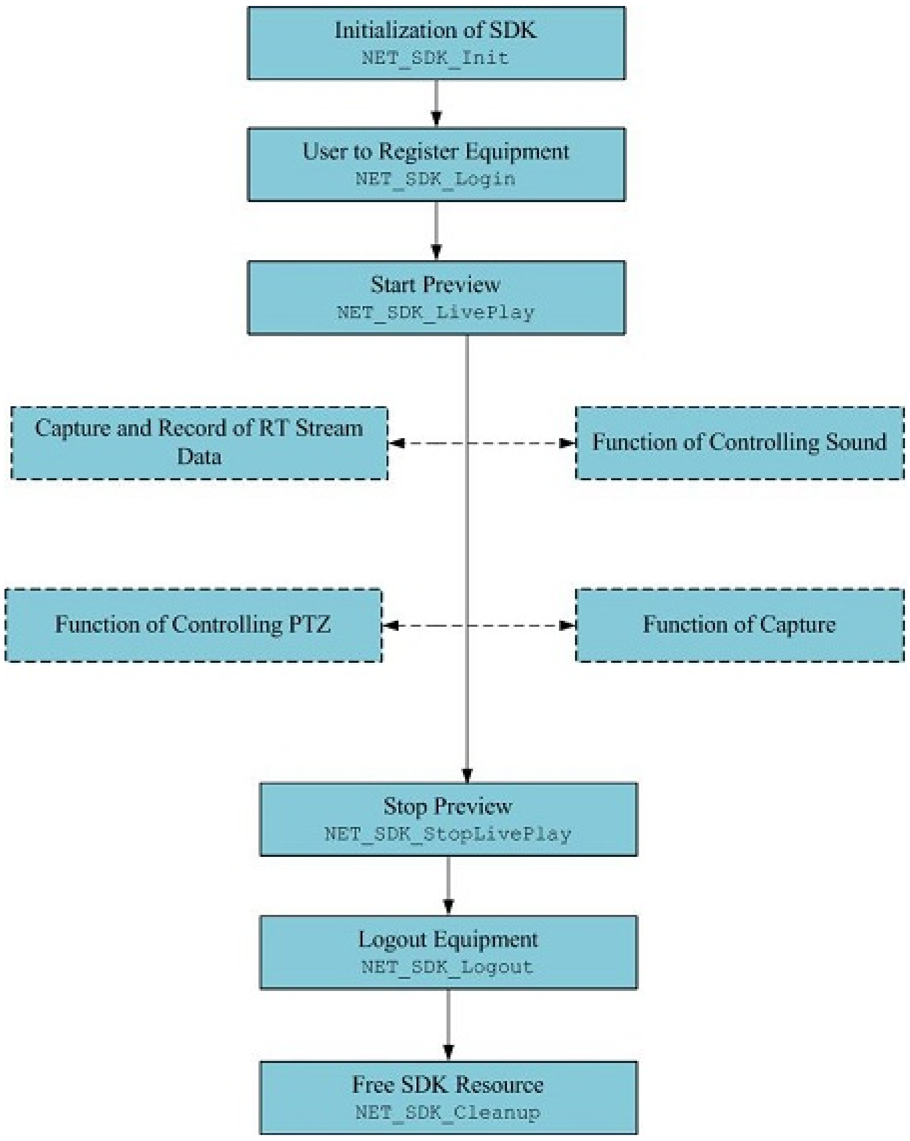
- Initialization of SDK interface:initialize whole network SDK system,preassign internal memory etc.

- Interface of setting connection timeout:this part is optional,user can set SDK network connection timeout time according to their need.If not call this interface,adopt the default in SDK.
- Set callback function of receiving exception information:most module function in SDK is realized with asynchronous mode,so we provide this interface to receive exception in preview,alarming, playback and talkback module.User can call this callback function after initialization of SDK to receive and dispose exception from all module on application layer.
- Get device IP address from analysis server:this interface provides the function of getting device IP address from analysis server in condition of only knowing device name and serial number.
- Interface of user register:realize function of register,under successful register,returned ID is the unique identification for other functions.SDK permits maximum of register user count is 512.For device,this version permits 32 register username,meanwhile permits 128 users to register at the same time.
- Preview module:get real time code stream from front server,decode to display and play control function etc.
- Play and download module:playback or download from front server by time remotely ,later encode and store,meanwhile support resuming from break point.
- Parameter configuration module:set and obtain parameter of front server,including device parameter,network parameter,channel compression parameter,port parameter,alarm parameter,exception parameter,exchange information and user configuration parameter etc.
- Remote device maintenance module:close device,reboot device,recover default setting, format harddisk remotely,remote upgrade,import&export of configuration file etc.

- Audio talkback module:audio data talkback and obtain from front server,audio encoding method can be appointed.
- Alarm module:dispose alarm signal uploaded from front server.
- PTZ control module:including basic operation on PTZ,preset point,cruise and track control.SDK divides PTZ control into two methods:control by the returned handle from image preview;preview without limit,user to control PTZ through register ID.

Client SDK Instructions

flow of preview module



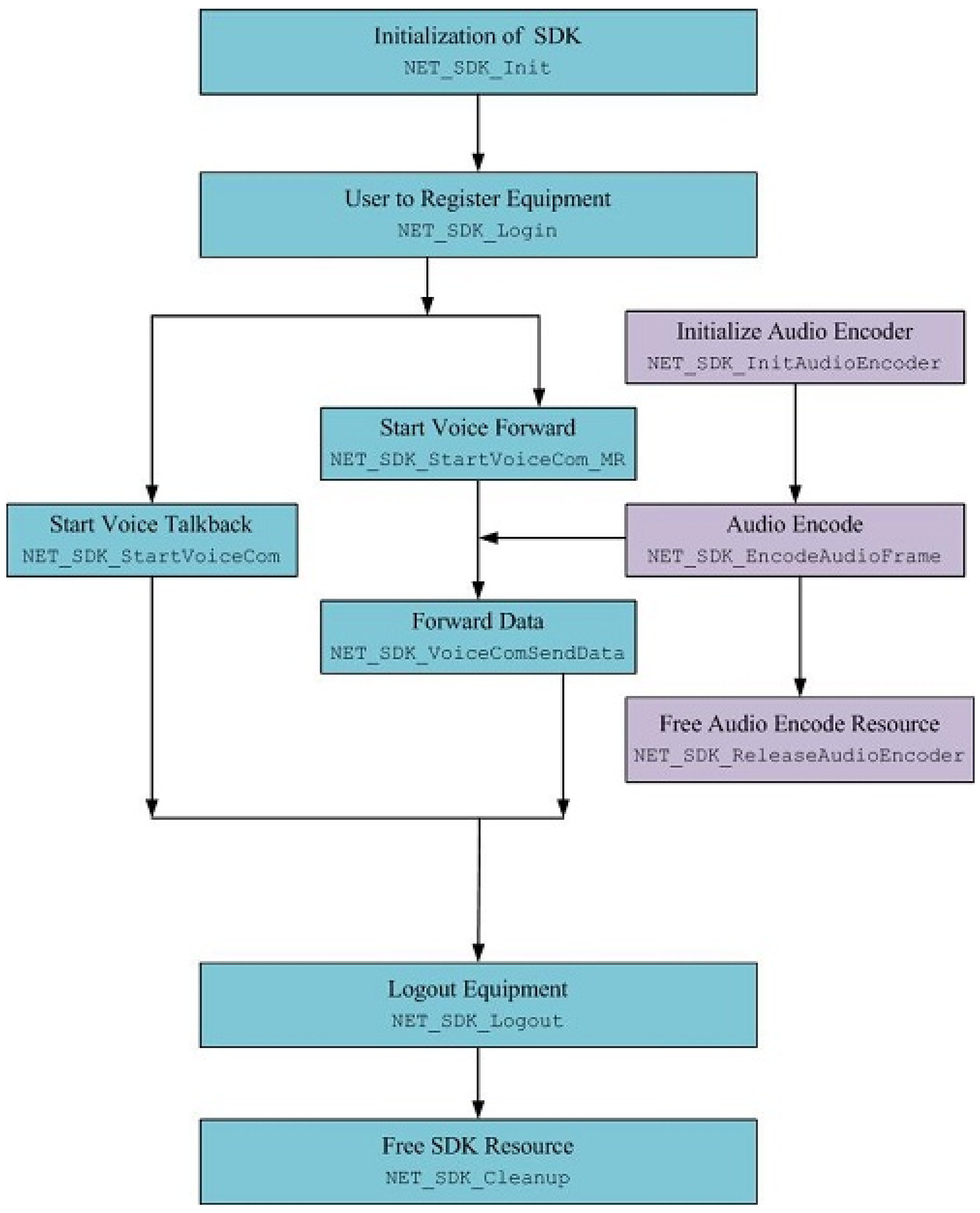
The part marked with dotted line is relative to preview module,call it in condition of open preview,the two modules are collocated and they both have their own functions.

- Volume control function mainly means monopolization and volume control.

- Real time data capture and record module mainly means data callback and local record for later dispose.
- Capture function mainly means capturing current image in decoding and store them into .BMP file.
- PTZ module means PTZ control and operation function in condition of open preview,including PTZ preset point,cruise and track etc.

Client SDK Instructions

flow of talkback and forward module

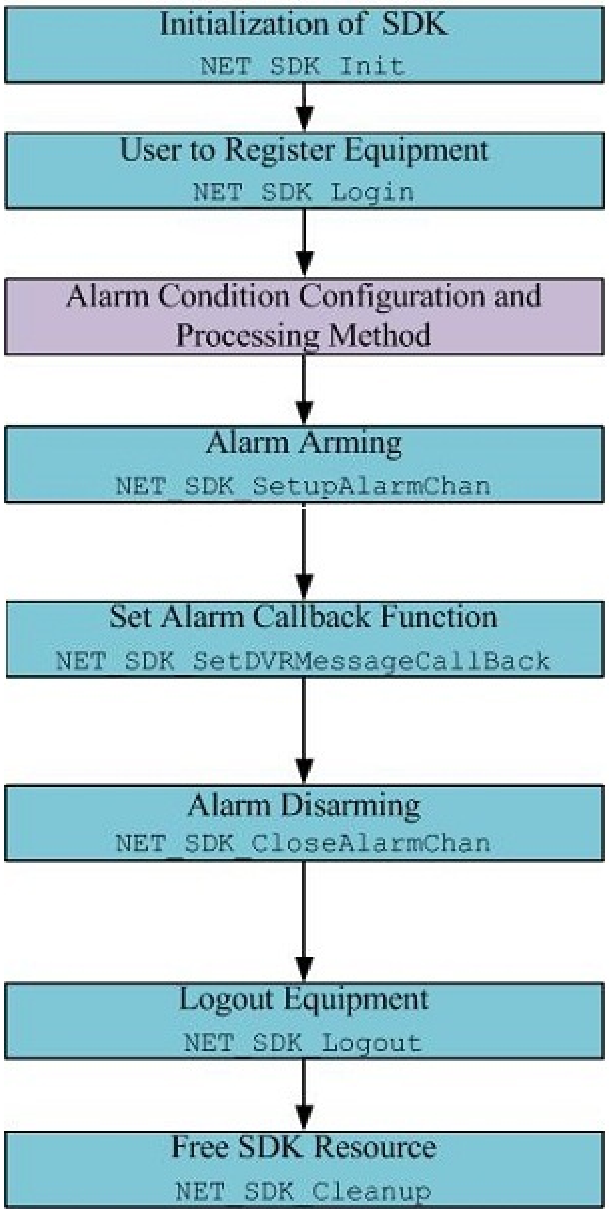


- Talkback function sends and receives audio between PC and device.Call NET_SDK_StartVoiceCom after device register,meanwhile user can get data which is sent by device or collected by PC through setting callback function.

- Firstly call NET_SDK_StartVoiceCom_MR to start talkback(connection with device has been done and wait for sending data). Second,prepare which data to send(need to encode),flow of encoding is the part in purple,if data has been compressed in appointed method omit the encoding part.Data resource can be from PC sound card or read from file,but must be compressed with our compression algorithm. After encoding operation we can get encoded data in fixed size,then call NET_SDK_VoiceComSendData to send the data to device. After sending all data,call NET_SDK_StopVoiceCom to stop forward connection with device.

Client SDK Instructions

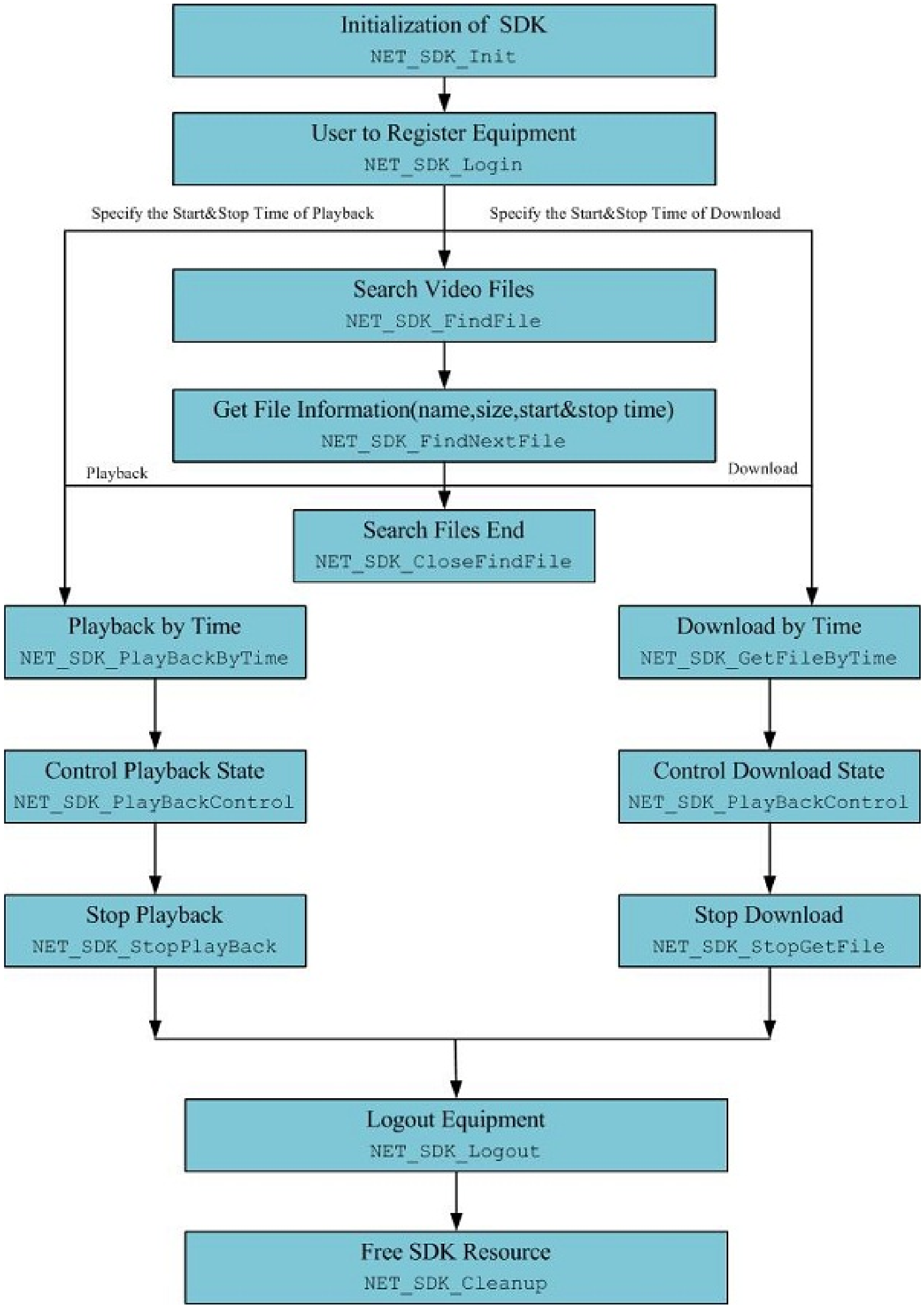
flow of alarm module



- Alarm method in this version is arming:SDK connects device actively, sends command of uploading alarm,then device alarms and sends to SDK. From the flow chart we can know that arming method needs user register first.The part in purple is a necessary conditon for realizing alarm information upload, mainly finishes relative alarm condition and process configuration,the interface of parameter configuration is NET_SDK_GetDVRConfig and NET_SDK_SetDVRConfig.Configurated struct for single quantity alarm is NET_SDK_AlarmInfo,if these parameters configuration has been done, next step is setting alarm callback function NET_SDK_SetDVRMessageCallBack,after above steps,set arming alarm NET_SDK_SetupAlarmChan. Cancel arming interface should be called after the whole alarm uploading progress.

Client SDK Instructions

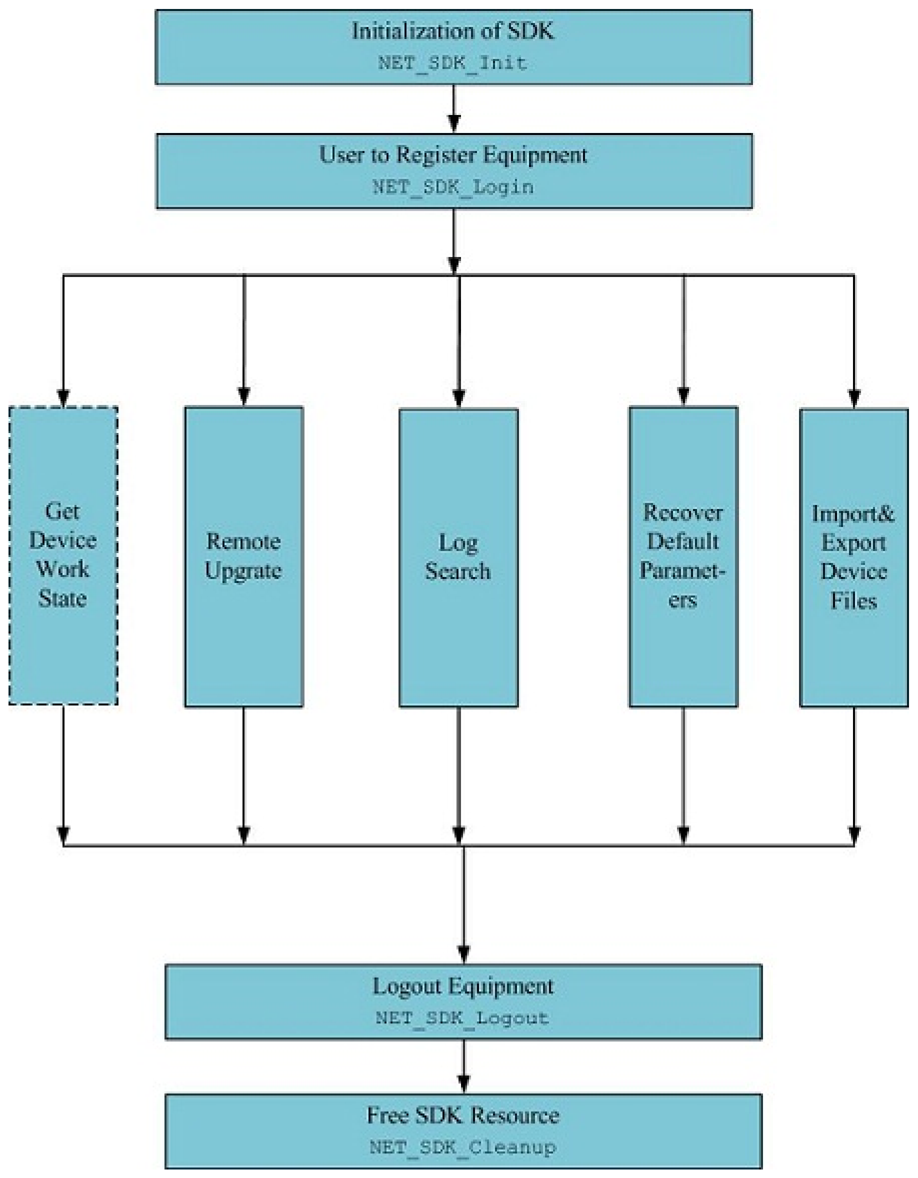
flow of playback and download by time module


Initialization of SDK
NET_SDK_Init
User to Register Equipment
NET_SDK_Login
Specify the Start&Stop Time of Playback
Specify the Start&Stop Time of Download
Search Video Files
NET_SDK_FindFile
Get File Information(name,size,start&stop time)
NET_SDK_FindNextFile
Playback
Download
Search Files End
NET_SDK_CloseFindFile
Playback by Time
NET_SDK_PlayBackByTime
Download by Time
NET_SDK_GetFileByTime
Control Playback State
NET_SDK_PlayBackControl
Control Download State
NET_SDK_PlayBackControl
Stop Playback
NET_SDK_StopPlayBack
Stop Download
NET_SDK_StopGetFile
Logout Equipment
NET_SDK_Logout
Free SDK Resource
NET_SDK_Cleanup

- When playback and downloading by time,user needn't call relative find file interface but just appoint the start time and stop time in interface. The start play command of control interface should be called after calling playback and downloading interface,then the nearest video will playback or download by appinted time range. Also user can call relative interface of finding record file to get the start time and stop time,and appoint the time parameter according to the returned time range,at last the start play command of control interface should be called too.

Client SDK Instructions

flow of remote device maintainance module

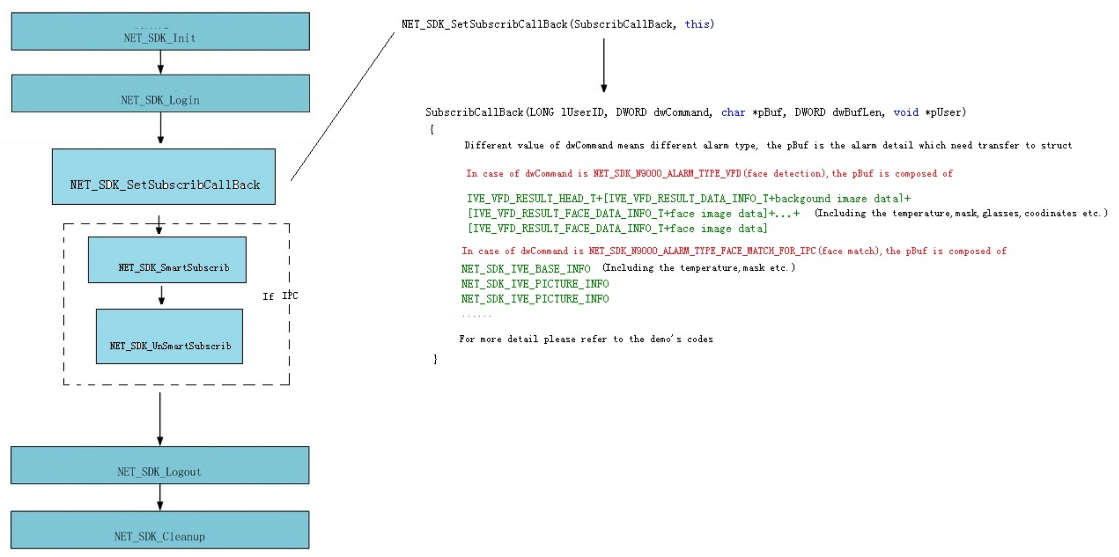


Remote device maintenance module includes getting device work status,remote upgrade,log search,recover device default parameter and import&export configuration file etc.

- Get device work status:get device current harddisk state,channel state,alarm import and export state,local display state and audio channel state,the part marked with dotted line is reserved temporarily.
- Remote upgrade:upgrade device and get the upgrade progress and state.
- Search log:search current device log information,including alarm,exception,operation and log with S.M.A.R.T information.
- Recover default device parameter:call NET_SDK_RestoreConfig to recover all default parameter setting.
- Import and export configuration file :export current all configuration information and store them or import the appointed configuration information.

Client SDK Instructions

flow of intelligent alarm

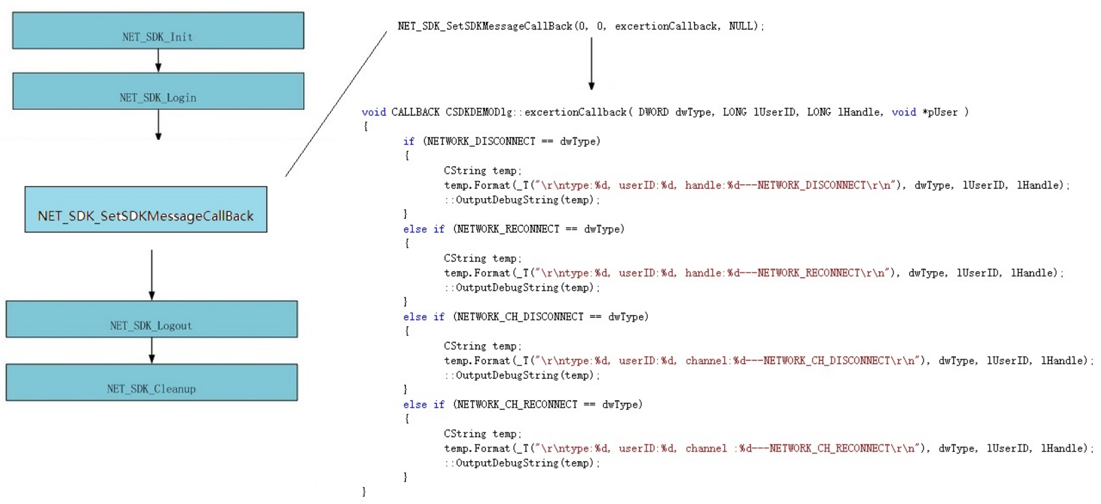


```
NET_SDK_SetSubscribCallBack(SubscribCallBack, this)

  SubscribCallBack(LONG lUserID, DWORD dwCommand, char *pBuf, DWORD dwBufLen, void *pUser)
   {
        Different value of dwCommand means different alarm type, the pBuf is the alarm detail which need transfer to struct

        In case of dwCommand is NET_SDK_N9000_ALARM_TYPE_VFD(face detection),the pBuf is composed of

        IVE_VFD_RESULT_HEAD_T+[IVE_VFD_RESULT_DATA_INFO_T+backgound image data]+
        [IVE_VFD_RESULT_FACE_DATA_INFO_T+face image data]+...+  (Including the temperature,mask,glasses,coodinates etc.)
        [IVE_VFD_RESULT_FACE_DATA_INFO_T+face image data]

       In case of dwCommand is NET_SDK_N9000_ALARM_TYPE_FACE_MATCH_FOR_IPC(face match),the pBuf is composed of
       NET_SDK_IVE_BASE_INFO  (Including the temperature,mask etc.)
       NET_SDK_IVE_PICTURE_INFO
       NET_SDK_IVE_PICTURE_INFO
       ......

       For more detail please refer to the demo's codes
   }
```

-

Client SDK Instructions

flow of exception

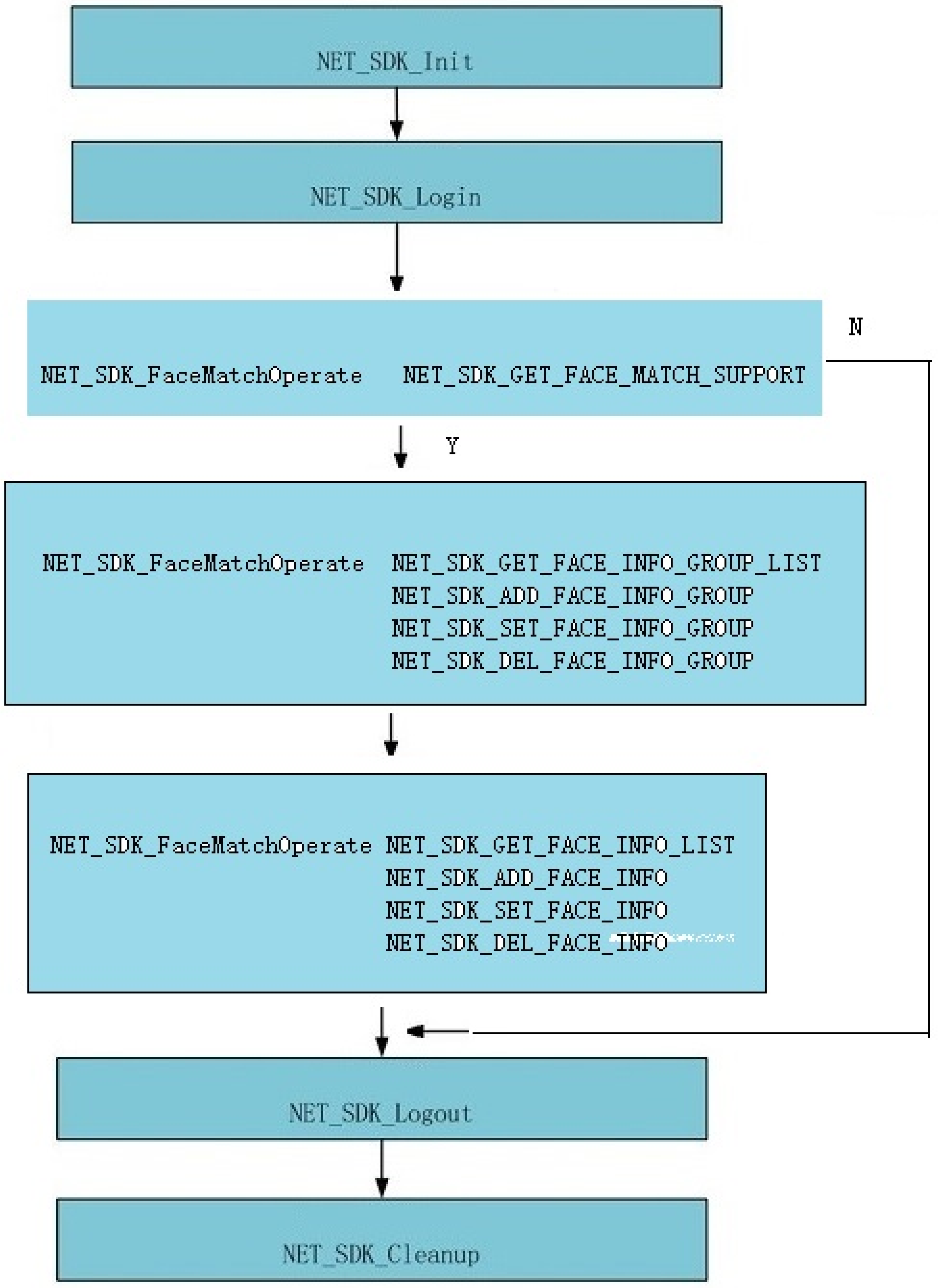


```
NET_SDK_SetSDKMessageCallBack(0, 0, excertionCallback, NULL);

void CALLBACK CSDKDEMODlg::excertionCallback( DWORD dwType, LONG lUserID, LONG lHandle, void *pUser )
{
	if (NETWORK_DISCONNECT == dwType)
	{
		CString temp;
		temp.Format(_T("\r\ntype:%d, userID:%d, handle:%d---NETWORK_DISCONNECT\r\n"), dwType, lUserID, lHandle);
		::OutputDebugString(temp);
	}
	else if (NETWORK_RECONNECT == dwType)
	{
		CString temp;
		temp.Format(_T("\r\ntype:%d, userID:%d, handle:%d---NETWORK_RECONNECT\r\n"), dwType, lUserID, lHandle);
		::OutputDebugString(temp);
	}
	else if (NETWORK_CH_DISCONNECT == dwType)
	{
		CString temp;
		temp.Format(_T("\r\ntype:%d, userID:%d, channel:%d---NETWORK_CH_DISCONNECT\r\n"), dwType, lUserID, lHandle);
		::OutputDebugString(temp);
	}
	else if (NETWORK_CH_RECONNECT == dwType)
	{
		CString temp;
		temp.Format(_T("\r\ntype:%d, userID:%d, channel :%d---NETWORK_CH_RECONNECT\r\n"), dwType, lUserID, lHandle);
		::OutputDebugString(temp);
	}
}
```

-

Client SDK Instructions

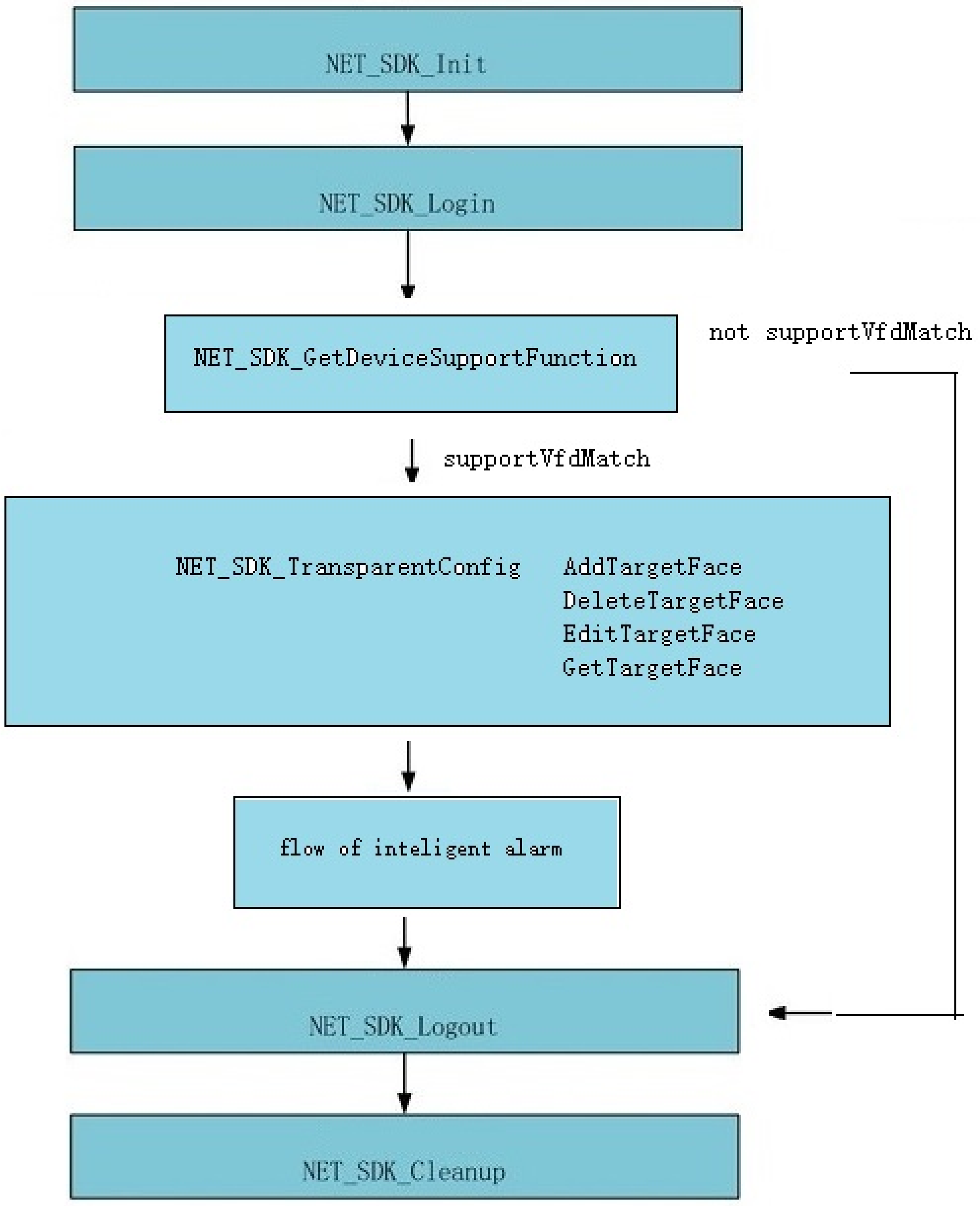
flow of config N9000's face album target


NET_SDK_Init
NET_SDK_Login
NET_SDK_FaceMatchOperate NET_SDK_GET_FACE_MATCH_SUPPORT
N
Y
NET_SDK_FaceMatchOperate NET_SDK_GET_FACE_INFO_GROUP_LIST
NET_SDK_ADD_FACE_INFO_GROUP
NET_SDK_SET_FACE_INFO_GROUP
NET_SDK_DEL_FACE_INFO_GROUP
NET_SDK_FaceMatchOperate NET_SDK_GET_FACE_INFO_LIST
NET_SDK_ADD_FACE_INFO
NET_SDK_SET_FACE_INFO
NET_SDK_DEL_FACE_INFO
NET_SDK_Logout
NET_SDK_Cleanup

-

Client SDK Instructions

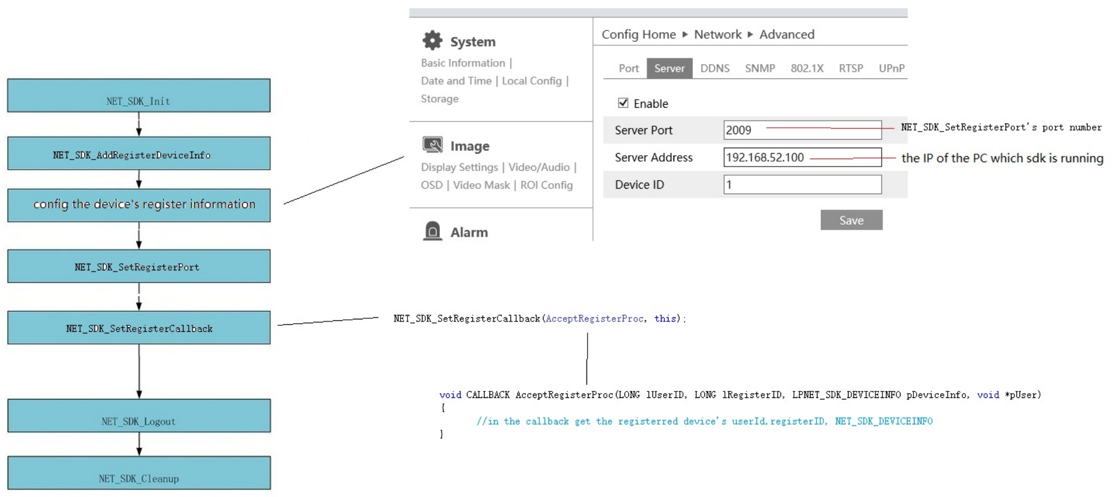
flow of config IPC's face album target



-

Client SDK Instructions

flow of accept device's register


NET_SDK_Init
NET_SDK_AddRegisterDeviceInfo
config the device's register information
NET_SDK_SetRegisterPort
NET_SDK_SetRegisterCallback
NET_SDK_Logout
NET_SDK_Cleanup
System
Basic Information | Date and Time | Local Config | Storage
Image
Display Settings | Video/Audio | OSD | Video Mask | ROI Config
Alarm
Config Home ▸ Network ▸ Advanced
Port
Server
DDNS
SNMP
802.1X
RTSP
UPnP
Save

☑ Enable

Server Port	2009
Server Address	192.168.52.100
Device ID	1

NET_SDK_SetRegisterPort's port number

the IP of the PC which sdk is running

```
NET_SDK_SetRegisterCallback(AcceptRegisterProc, this);
```

```
void CALLBACK AcceptRegisterProc(LONG lUserID, LONG lRegisterID, LPNET_SDK_DEVICEINFO pDeviceInfo, void *pUser)
{
        //in the callback get the registerred device's userId,registerID, NET_SDK_DEVICEINFO
}
```

-

Interface Definition

This section mainly introduces all the interface definitions involved in SDK , one single page corresponding to one interface definition,
and has a brief description for parameters and return values.
Further more the structures' definitions that involved in interface definitions are added,also one single page corresponding to one structure definition.

Client SDK Instructions

Macro Definition

macro definition	value of macro definition	meaning
DD_MAX_CAMERA_NUM	128	the maximum number of inserted camera shooting device
DD_MAX_CAMERA_NUM_BYTE_LEN	16	the maximum byte length of inserted camera shooting device
DD_MAX_SERIAL_NUMBER_LEN	64	length of serial number
DD_MAX_VERSION_BUF_LEN	64	length of version buffer
DD_MAX_NAME_LEN	64	length of user name
DD_MAX_NAME_BUF_LEN	132	buffer length of user name
DD_MAX_CAMERA_NAME_LEN	64	the maximum

		name length of inserted camera shooting device
DD_MAX_CAMERA_NAME_BUF_LEN	132	the maximum name buffer length of inserted camera device
DD_MAX_URL_LEN	256	the maximum length of input URL
DD_MAX_URL_BUF_LEN	260	the maximum buffer length of input URL
DD_MAX_COLOR_CFG_NUM	3	number of stream for controlling color
DD_MAX_TEXT_LEN	64	the maximum length of input text
DD_MAX_TEXT_BUF_LEN	132	the maximum buffer length of input text
DD_MAX_VIDEO_COVER_NUM	3	the

		maximum number of video override
DD_MAX_USER_NAME_LEN	64	the maximum length of user name
DD_MAX_USER_NAME_BUF_LEN	132	buffer length of user name
DD_MAX_PASSWORD_LEN	128	the maximum length of password
DD_MAX_PASSWORD_BUF_LEN	132	the maximum buffer length of password
DD_MAX_PPPOE_ACCOUNT_LEN	128	the maximum length of PPPOE dialling number
DD_MAX_PPPOE_ACCOUNT_BUF_LEN	132	buffer length of PPPOE dialling number
DD_MAX_DDNS_ACCOUNT_LEN	128	the maximum length of

		DDNS number
DD_MAX_DDNS_ACCOUNT_BUF_LEN	132	the maximum length of DDNS number
DD_MAX_EMAIL_RECEIVE_ADDR_NUM	3	number of address for receiving emails
DD_MAX_MOTION_AREA_WIDTH_NUM	1920/16	value of the width of motion area
DD_MAX_MOTION_AREA_HIGHT_NUM	((1080/16) + 3) / 4	value of the height of motion area
DD_MAX_PRESET_NUM	128	number of PTZ preset points
DD_MAX_CRUISE_NUM	32	number of PTZ cruise
DD_MAX_TRACK_NUM	1	number of PTZ track
DD_MAX_ACCOUNT_NUM	64	the maximum number of user
DD_MAX_BUF_SIZE	512*1024	size of buffer

Client SDK Instructions

NET_SDK_Init

initialize SDK,before calling other functions in SDK

```
BOOL NET_SDK_Init(
);
```

Return Values

TRUE means success; FALSE means failure.To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Cleanup

Client SDK Instructions

NET_SDK_SetConnectTime

set network connection timeout time and connection times

```
BOOL NET_SDK_SetConnectTime(
DWORD     dwWaitTime
DWORD     dwTryTimes
);
```

Parameters

dwWaitTime
[in] timeout time,in millisecond,its value is greater than 300. actual maximum timeout value is **connect** timeout value(**connect** timeout value depends on different system)the excess part is invalid,default value is 5 seconds.

dwTryTimes
[in] reconnection times(kept),default value is 3

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

SDK default timeout value is 5 seconds.

Client SDK Instructions

NET_SDK_SetReconnect

set reconnection function

```
BOOL NET_SDK_SetReconnect(
DWORD     dwInterval,
BOOL      bEnableRecon
);
```

Parameters

dwInterval
[in] reconnection interval,in millisecond,default value is 30 seconds

bEnableRecon
[in] whether to reconnect,0- no reconnect,1- reconnect,default value is 1

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface can control preview, transparent channel and reconnection functions in arming at the same time.When it isn't called SDK enables the three functions acquiescently, and the reconnection time interval is 5 seconds.

Client SDK Instructions

NET_SDK_Cleanup

before finish, the last step is to free SDK resource.

```
BOOL NET_SDK_Cleanup(
);
```

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Init

Client SDK Instructions

NET_SDK_DiscoverDevice

discover device automatically on LAN

```
long NET_SDK_DiscoverDevice(
NET_SDK_DEVICE_DISCOVERY_INFO            *pDeviceInfo,
long   bufNum;
long   waitSeconds;
);
```

Parameters

**pDeviceInfo*
: [in] an array witch is needed to asign values,its size is **bufNum** ,if descovered device num is more than,the returned size is just **bufNUm**

bufNum
: [in] size of the array

waitSeconds
: [in] time to discover devices, unit is second,this interface will be returned after **waitSeconds**

Return Values

Returned value is the number of discovered devices, if no device is found or discovering device gets error,the value is 0. To get error information, please refer to NET_SDK_GetLastError

See Also

NET_SDK_GetDeviceInfo

Client SDK Instructions

NET_SDK_DiscoverDeviceStart

start discover device on LAN(asynchronous, only for windows)

```
unsigned int NET_SDK_DiscoverDeviceStart(
IPTool_SearchDataCallBack              SearchCallBack,
IPTool_SearchDataCallBackEx              SearchCallBackEx,
void     *pParam,
unsigned int   SearchTypeMask,
int  nMaxRecordCount
);
```

Parameters

SearchCallBack
  search result callback during searching, result format is xml
SearchCallBackEx
  [in] search result callback during searching, result format is SEARCHED_DEVICE_INFO struct
**pParam*
  pointer to user data
SearchTypeMask
  search type，refer to the table：

type	value	meanning
_SEARCH_STANDARD	0x001	standard device
_SEARCH_ONVIF	0x002	onvif device
_SEARCH_UPNP	0x004	upnp device
_SEARCH_AIPSTAR	0x008	AIPSTAR device
_SEARCH_DAHUA	0x010	DAHUA device
_SEARCH_HIK	0x020	HIK device
_SEARCH_UNIVIEW	0x040	UNIVIEW device
_SEARCH_YCX	0x080	YCX device
_SEARCH_SPECO	0x100	SPECO device
_SEARCH_ALL	0xffff	all type

nMaxRecordCount
  max number of searched devices

Return Values

Returned value is the handle of searching. To get error information, please refer to NET_SDK_GetLastError

---

Client SDK Instructions

NET_SDK_DiscoverDeviceStop

stop asynchronous searching（only for windows）

```
void NET_SDK_DiscoverDeviceStop(
unsigned int   hSearch,
);
```

Parameters

hSearch
the handle of searching, the return value of NET_SDK_DiscoverDeviceStart

Client SDK Instructions

IPTool_SearchDataCallBack

call back of asynchronous searching

```
void *IPTool_SearchDataCallBack(
char*   hwaddr ,
char*   szDevIP ,
int    opt ,
const char*    szXmlData,
void *    pParam,
const char *  szRecvFromNIC
);
```

Parameters

hwaddr
: hard ware address

szDevIP
: IP

opt
: reserve

szXmlData
: the searched device's information with xml format

pParam
: reserve

szRecvFromNIC
: reserve

Return Values

No return value. To get error information, please refer to NET_SDK_GetLastError

Client SDK Instructions

IPTool_SearchDataCallBackEx

call back of asynchronous searching

```
void *IPTool_SearchDataCallBackEx(
char*    hwaddr ,
char*    szDevIP ,
int     opt ,
const SEARCHED_DEVICE_INFO *     pData ,
void *     pParam,
const char *   szRecvFromNIC
);
```

Parameters

hwaddr
  hard ware address
szDevIP
  IP
opt
  reserve
pData
  the searched device's information
pParam
  reserve
szRecvFromNIC
  reserve

Return Values

No return value. To get error information, please refer to NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetRegisterCallback

callback function when device receives DVR registered local port

```
 BOOL NET_SDK_SetRegisterCallback(
ACCEPT_REGISTER_CALLBACK           fRegisterCBFun
 void                 *pUser
 );
```

Parameters

fRegisterCBFun
: [in] callback information when device receives DVR registered local port

**pUser*
: [in] custom parameter passed by user

Return Values

TRUE means sccess; FALSE means failure. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_SetRegisterPort   ACCEPT_REGISTER_CALLBACK

Client SDK Instructions

NET_SDK_SetUnRegisterCallback

the function is a callback for the unregistered device,calls NET_SDK_AddRegisterDeviceInfo to notify SDK after receiving callback

```
BOOL NET_SDK_SetUnRegisterCallback(
 ACCEPT_UNREGISTER_CALLBACK          fUnRegisterCBFun
void                 *pUser
);
```

Parameters

fUnRegisterCBFun
  [in] callback information when device receives DVR registered local port
**pUser*
  [in] custom parameter passed by user

Return Values

TRUE means sccess; FALSE means failure. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_SetRegisterPort   ACCEPT_REGISTER_CALLBACK

Client SDK Instructions

NET_SDK_SetRegisterPort

local port when device receives DVR's register

```
BOOL NET_SDK_SetRegisterPort(
WORD   wRegisterPort
);
```

Parameters

wRegisterPort
[in] local port when device receives DVR's register

Return Values

TRUE means sccess; FALSE means failure. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_SetRegisterCallback   ACCEPT_REGISTER_CALLBACK

Client SDK Instructions

NET_SDK_AddRegisterDeviceInfo

add the informations of the devices which need to auto register to sdk

```
 BOOL NET_SDK_AddRegisterDeviceInfo(
REG_LOGIN_INFO   * pLoginInfo,
unsigned int deviceNum
 );
```

Parameters

REG_LOGIN_INFO
  [in] the pointer of the devices' information
deviceNum
  [in] number of the devices

Return Values

TRUE means sccess; FALSE means failure. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_SetRegisterCallback   *ACCEPT_REGISTER_CALLBACK*

Client SDK Instructions

ACCEPT_REGISTER_CALLBACK

callback information when device receives DVR's register

```
void ACCEPT_REGISTER_CALLBACK(
LONG            lUserID,
LONG            lRegisterID,
LPNET_SDK_DEVICEINFO            pDeviceInfo,
void            *pUser
);
```

Parameters

lUserID
: [in] connection ID,other interface accesses device through this ID

lRegisterID
: [in] device received DVR initiative register ID

pDeviceInfo
: [in] information of initiative register device

**pUser*
: [in] custom parameter passed by user

Return Values

None. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_SetRegisterCallback   NET_SDK_SetRegisterPort

Client SDK Instructions

NET_SDK_GetDeviceIPByName

Get the device Ip by the device name

```
BOOL NET_SDK_GetDeviceIPByName(
char          *sSerIP,
DWORD            wSerPort;
char          *sDvrName;
char          *sDvrIP
);
```

Parameters

**sSerIP*
  [in] the IP address of IPSever
wSerPort
  [in] the port of IPSever
**sDvrName*
  [in] the device name automatically reported to IP Server
**sDvrIP*
  [in] the device port automatically reported to IP Server

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_DiscoverDevice

Client SDK Instructions

NET_SDK_SetSDKMessageCallBack

callback function of SDK operation exception

```
BOOL  NET_SDK_SetSDKMessageCallBack(
UINT                    nMessage,
HWND                    hWnd,
EXCEPTION_CALLBACK      fExceptionCallBack,
void                    *pUser
);
```

Parameters

nMessage
[in] message

hWnd
[in] window handle of receiving exception message

fExceptionCallBack
[in] callback function of receiving exception message,callback current relevant information of exception

**pUser*
[in] user data

Callback Function

```
void (CALLBACK fExceptionCallBack)(
DWORD    dwType,
LONG     lUserID,
LONG     lHandle,
void     *pUser
);
```

Callback Function Parameters

dwType
   message type of exception or reconnection
lUserID
   login ID
lHandle
   relevant type handle of exception
pUser
   user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

---

Client SDK Instructions

EXCEPTION_CALLBACK

data callback during SDK exception

```
void *EXCEPTION_CALLBACK(
DWORD             dwType,
LONG              lUserID,
LONG              lHandle,
void              *pUser
);
```

Parameters

dwType
[in] type of exception or reconnection message,refer to NET_SDK_EXCEPTION_TYPE:

Type	Value	Description
NETWORK_DISCONNECT	0	Disconnection
NETWORK_RECONNECT	1	Reconnection
NETWORK_CH_DISCONNECT	2	Channel Disconnection
NETWORK_CH_RECONNECT	3	Channel Reconnection

lUserID
[in] login ID
lHandle
[in] corresponding type handle when exception
** pUser*
[in] pointer to user data

Return Values

None. To get error code, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetVideoEffect

Set the display parameter of the video.

```
BOOL NET_SDK_SetVideoEffect(
LONG       lUserID,
LONG       lChannel,
DWORD      dwBrightValue,
DWORD      dwContrastValue,
DWORD      dwSaturationValue,
DWORD      dwHueValue
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] return value of NET_SDK_CLIENTINFO,channel number starts from 0
dwBrightValue
[out] brightness,range[0,255]
dwContrastValue
[out] contrast,range[0,255]
dwSaturationValue
[out] saturation,range[0,255]
dwHueValue
[out] gray scale,range[0,255]

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_GetVideoEffect

Set the display parameter of the video.

```
BOOL NET_SDK_GetVideoEffect(
LONG        lUserID,
LONG        lChannel,
DWORD       *pBrightValue,
DWORD       *pContrastValue,
DWORD       *pSaturationValue,
DWORD       *pHueValue
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

lChannel
[in] return value of NET_SDK_CLIENTINFO,channel number starts from 0

pBrightValue
[out] pointer to brightness,range[0,255]

pContrastValue
[out] pointer to contrast,range[0,255]

pSaturationValue
[out] pointer to saturation,range[0,255]

pHueValue
[out] pointer to gray scale,range[0,255]

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_SetVideoEffect_Ex

Set the display parameter of the video.

```
BOOL NET_SDK_SetVideoEffect_Ex(
LONG      lUserID,
LONG      lChannel,
DWORD     dwBrightValue,
DWORD     dwContrastValue,
DWORD     dwSaturationValue,
DWORD     dwHueValue
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
lChannel
[in] the return value of NET_SDK_CLIENTINFO. The channel number starts from 0
dwBrightValue
[out] Brightness: the return value range of NET_SDK_SetVideoEffect_Ex : [minValue,maxValue]
dwContrastValue
[out] Contrast: the return value range of NET_SDK_SetVideoEffect_Ex: [minValue,maxValue]
dwSaturationValue
[out] Saturation: the return value range of NET_SDK_SetVideoEffect_Ex : [minValue,maxValue]
dwHueValue
[out] Hue: the return value range of NET_SDK_SetVideoEffect_Ex :[minValue,maxValue]

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Android Interface

The corresponding Android interface：NO

Client SDK Instructions

typedef struct

```
{
    unsigned int          minValue;                // the minimum valu
    unsigned int          maxValue;                // the maximum valu
    unsigned int          curValue;                // the current valu
    unsigned int          defaultValue;                // the default
}NET_SDK_IMAGE_EFFECT_T;
```

NET_SDK_GetVideoEffect_Ex

Get the display parameter of video.

```
BOOL NET_SDK_GetVideoEffect_Ex(
LONG      lUserID,
LONG      lChannel,
NET_SDK_IMAGE_EFFECT_T     *pBrightValue,
NET_SDK_IMAGE_EFFECT_T     *pContrastValue,
NET_SDK_IMAGE_EFFECT_T     *pSaturationValue,
NET_SDK_IMAGE_EFFECT_T     *pHueValue
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login().
lChannel
[in] the return value of NET_SDK_CLIENTINFO. The channel number starts from 0.
pBrightValue
[out] a pointer to brightness
pContrastValue
[out] a pointer to contrast
pSaturationValue
[out] a pointer to saturation
pHueValue
[out] a pointer to hue

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Android Interface

The corresponding Android interface：NO

Client SDK Instructions

NET_SDK_GetSDKBuildVersion

get version and build information of SDK

```
DWORD NET_SDK_GetSDKBuildVersion(
);
```

Return Values

Version and build information of SDK.Two high bytes are the version:25~32 bits are main version,17~24 bits are minor version;two low bytes are build information. Take 0x01000101 for example:version is 1.0,build number is 0101.

Client SDK Instructions

NET_SDK_GetSDKVersion

get information of SDK version

```
DWORD NET_SDK_GetSDKVersion(
);
```

Return Values

Version of SDK.Two high bytes are the version:25~32 bits are main version,17~24 bits are minor version;two low bytes are build information. Take 0x01000101 for example:version is 1.0,build number is 0101.

Client SDK Instructions

NET_SDK_SetLogToFile

enable writing log file

```
BOOL NET_SDK_SetLogToFile(
BOOL     bLogEnable,
char     *strLogDir,
BOOL     bAutoDel
);
```

Parameters

bLogEnable
[in] whether enable the function of writing log,default value is FALSE

strLogDir
[in] directory of log file,default directory is "C:\\SdkLog\\"

bAutoDel
[in] whether delete excess file count,default value is TRUE

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Directory of log file must be absolute path,and ends with "\\",for example:"C:\\SdkLog\\".Advise user to create file manually,if no specified path,adopt default path "C:\\SdkLog\\". This interface can be called many times to create log files, and support creating 10 files at most.When assign bAutoDel TRUE,system will delete excess file automatically.New directory will be valid in writing file when changing directory or next time to write file.

Client SDK Instructions

NET_SDK_GetErrorMsg

return the last error code message

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

Parameters

pErrorNo
[out] pointer to the value of error code

Return Values

return value is pointer to error code information. error message has two main types,error message of network communication library and error message of soft and hard decoding library,list the first type as follows:

error message of network communication library

type of errors	error value	i
NET_SDK_SUCCESS	0	no error
NET_SDK_PASSWORD_ERROR	1	user's name or pass
NET_SDK_NOENOUGH_AUTH	2	no right for this ope
NET_SDK_NOINIT	3	SDK is not initializ
NET_SDK_CHANNEL_ERROR	4	error of channel nu
NET_SDK_OVER_MAXLINK	5	the client connecte
NET_SDK_LOGIN_REFUSED	6	SDK login is refuse
NET_SDK_VERSION_NOMATCH	7	version doesn't mat
NET_SDK_NETWORK_FAIL_CONNECT	8	failed to connect to
NET_SDK_NETWORK_NOT_CONNECT	9	network isn't conne
NET_SDK_NETWORK_SEND_ERROR	10	failed to send data t
NET_SDK_NETWORK_RECV_ERROR	11	failed to receive the
NET_SDK_NETWORK_RECV_TIMEOUT	12	timeout when recei
NET_SDK_NETWORK_ERRORDATA	13	send illegal data to
NET_SDK_ORDER_ERROR	14	the called order err
NET_SDK_OPER_BY_OTHER	15	operation method is
NET_SDK_OPER_NOPERMIT	16	the privileged user
NET_SDK_COMMAND_TIMEOUT	17	DVR command tim
NET_SDK_ERROR_SERIALPORT	18	error of serial port
NET_SDK_ERROR_ALARMPORT	19	error of alarm port
NET_SDK_PARAMETER_ERROR	20	parameter error
NET_SDK_CHAN_EXCEPTION	21	server's channel is i
NET_SDK_NODISK	22	no hard disk
NET_SDK_ERROR_DISKNUM	23	hard disk no. error
NET_SDK_DISK_FULL	24	server hark disk is f

NET_SDK_DISK_ERROR	25	server hard disk err
NET_SDK_NOSUPPORT	26	server does not sup
NET_SDK_BUSY	27	server is busy
NET_SDK_MODIFY_FAIL	28	failed to modify in
NET_SDK_PASSWORD_FORMAT_ERROR	29	the password inputt
NET_SDK_DISK_FORMATING	30	hard disk is formatt
NET_SDK_DVR_NORESOURCE	31	DVR no resources
NET_SDK_DVR_OPRATE_FAILED	32	DVR failed to opea
NET_SDK_OPEN_HOSTSOUND_FAIL	33	failed open PC voic
NET_SDK_DVR_VOICEOPENED	34	server voice dialogu
NET_SDK_TIME_INPUTERROR	35	time input is not co
NET_SDK_NOSPECFILE	36	there is no appointe
NET_SDK_CREATEFILE_ERROR	37	failed to create a fil
NET_SDK_FILEOPENFAIL	38	faile to open a file
NET_SDK_OPERNOTFINISH	39	the last operation is
NET_SDK_GETPLAYTIMEFAIL	40	faile to get the curre
NET_SDK_PLAYFAIL	41	failed to play
NET_SDK_FILEFORMAT_ERROR	42	the file input forma
NET_SDK_DIR_ERROR	43	path error
NET_SDK_ALLOC_RESOURCE_ERROR	44	resources allotting e
NET_SDK_AUDIO_MODE_ERROR	45	display card mode e
NET_SDK_NOENOUGH_BUF	46	buffer is not enough
NET_SDK_CREATESOCKET_ERROR	47	establish SOCKET
NET_SDK_SETSOCKET_ERROR	48	set SOCKET error
NET_SDK_MAX_NUM	49	the max number
NET_SDK_USERNOTEXIST	50	user doest not exit
NET_SDK_WRITEFLASHERROR	51	wirte FLASH error

NET_SDK_UPGRADEFAIL	52	failed to upgrade D
NET_SDK_CARDHAVEINIT	53	the decode card is i
NET_SDK_PLAYERFAILED	54	player failed
NET_SDK_MAX_USERNUM	55	the max user no.
NET_SDK_GETLOCALIPANDMACFAIL	56	failed to get the IP a end or physical add
NET_SDK_NOENCODEING	57	the channel is not c
NET_SDK_IPMISMATCH	58	IP address not matc
NET_SDK_MACMISMATCH	59	MAC address not m
NET_SDK_UPGRADELANGMISMATCH	60	the language of upg
NET_SDK_MAX_PLAYERPORT	61	reach to the max pl
NET_SDK_NOSPACEBACKUP	62	no enough space to
NET_SDK_NODEVICEBACKUP	63	no backup device
NET_SDK_PICTURE_BITS_ERROR	64	the bits of picture n
NET_SDK_PICTURE_DIMENSION_ERROR	65	the dimension is ov
NET_SDK_PICTURE_SIZ_ERROR	66	the size of picture is
NET_SDK_LOADPLAYERSDKFAILED	67	failed to load player
NET_SDK_LOADPLAYERSDKPROC_ERROR	68	not find some funct
NET_SDK_LOADDSSDKFAILED	69	failed to load DsSD
NET_SDK_LOADDSSDKPROC_ERROR	70	not find some funct
NET_SDK_DSSDK_ERROR	71	failed to call functio
NET_SDK_VOICEMONOPOLIZE	72	voice card is monop
NET_SDK_JOINMULTICASTFAILED	73	failed join to multic
NET_SDK_CREATEDIR_ERROR	74	failed to create log
NET_SDK_BINDSOCKET_ERROR	75	failed to bind socke
NET_SDK_SOCKETCLOSE_ERROR	76	socket is closed
NET_SDK_USERID_ISUSING	77	the user ID is opera

NET_SDK_PROGRAM_EXCEPTION	78	sdk program except
NET_SDK_WRITEFILE_FAILED	79	write file failed
NET_SDK_FORMAT_READONLY	80	failed to format rea
NET_SDK_WITHSAMEUSERNAME	81	there is same userna
NET_SDK_DEVICETYPE_ERROR	82	device type no matc
NET_SDK_LANGUAGE_ERROR	83	language no match
NET_SDK_PARAVERSION_ERROR	84	soft version no mat
NET_SDK_FILE_SUCCESS	85	file has been create
NET_SDK_FILE_NOFIND	86	file isn't found
NET_SDK_NOMOREFILE	87	there is no more fil
NET_SDK_FILE_EXCEPTION	88	file exception
NET_SDK_TRY_LATER	89	Try again later
NET_SDK_DEVICE_OFFLINE	90	Device offline
NET_SDK_CREATEJPEGSTREAM_FAIL	91	Failed to create JPE
NET_SDK_USER_ERROR_NO_USER	92	No such user!
NET_SDK_USER_ERROR_USER_OR_PASSWORD_IS_NULL	93	No username or pas
NET_SDK_USER_ERROR_ALREDAY_LOGIN	94	The user has been l
NET_SDK_USER_ERROR_SYSTEM_BUSY	95	The device is busy.
NET_SDK_DEVICE_NOT_SUPPROT	96	The device don not
NET_SDK_USER_ERROR_SYSTEM_NO_READY	97	Do not complete ge
NET_SDK_CHANNEL_OFFLINE	98	Camera is offline.
NET_SDK_GETREADYINFO_FAIL	99	It fails to get device
NET_SDK_NORESOURCE	100	SDK resources is n
NET_SDK_DEVICE_QUERYSYSTEMCAPS_FAIL	101	The device fails to
NET_SDK_INBUFFER_TOSMALL	102	The input buffer ar
NET_SDK_NO_PASSWORD_STRENGTH	103	The password stren

Remarks

Get error number through function NET_SDK_GetErrorMsg

See Also

NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_GetLastError

return the last error code of operation

```
DWORD NET_SDK_GetLastError(
);
```

Return Values

return value is pointer to error code information. error message has two main types,error message of network communication library and error message of soft and hard decoding library,list the first type as follows:

error message of network communication library

type of errors	error value	
NET_SDK_SUCCESS	0	no error
NET_SDK_PASSWORD_ERROR	1	user's name or pa
NET_SDK_NOENOUGH_AUTH	2	no right for this o
NET_SDK_NOINIT	3	SDK is not initial
NET_SDK_CHANNEL_ERROR	4	error of channel n
NET_SDK_OVER_MAXLINK	5	the client connect
NET_SDK_LOGIN_REFUSED	6	SDK login is refu
NET_SDK_VERSION_NOMATCH	7	version doesn't m
NET_SDK_NETWORK_FAIL_CONNECT	8	failed to connect
NET_SDK_NETWORK_NOT_CONNECT	9	network isn't con
NET_SDK_NETWORK_SEND_ERROR	10	failed to send dat
NET_SDK_NETWORK_RECV_ERROR	11	failed to receive t
NET_SDK_NETWORK_RECV_TIMEOUT	12	timeout when rec
NET_SDK_NETWORK_ERRORDATA	13	send illegal data t
NET_SDK_ORDER_ERROR	14	the called order e
NET_SDK_OPER_BY_OTHER	15	operation method
NET_SDK_OPER_NOPERMIT	16	the privileged use
NET_SDK_COMMAND_TIMEOUT	17	DVR command ti
NET_SDK_ERROR_SERIALPORT	18	error of serial por
NET_SDK_ERROR_ALARMPORT	19	error of alarm por
NET_SDK_PARAMETER_ERROR	20	parameter error
NET_SDK_CHAN_EXCEPTION	21	server's channel i
NET_SDK_NODISK	22	no hard disk
NET_SDK_ERROR_DISKNUM	23	hard disk no. erro
NET_SDK_DISK_FULL	24	server hark disk i
NET_SDK_DISK_ERROR	25	server hard disk e
NET_SDK_NOSUPPORT	26	server does not su
NET_SDK_BUSY	27	server is busy
NET_SDK_MODIFY_FAIL	28	failed to modify i
NET_SDK_PASSWORD_FORMAT_ERROR	29	the password inpu
NET_SDK_DISK_FORMATING	30	hard disk is forma
NET_SDK_DVR_NORESOURCE	31	DVR no resource
NET_SDK_DVR_OPRATE_FAILED	32	DVR failed to op
NET_SDK_OPEN_HOSTSOUND_FAIL	33	failed open PC vo
NET_SDK_DVR_VOICEOPENED	34	server voice dialo
NET_SDK_TIME_INPUTERROR	35	time input is not c
NET_SDK_NOSPECFILE	36	there is no appoin
NET_SDK_CREATEFILE_ERROR	37	failed to create a
NET_SDK_FILEOPENFAIL	38	faile to open a fil
NET_SDK_OPERNOTFINISH	39	the last operation
NET_SDK_GETPLAYTIMEFAIL	40	faile to get the cu
NET_SDK_PLAYFAIL	41	failed to play

NET_SDK_FILEFORMAT_ERROR	42	the file input form
NET_SDK_DIR_ERROR	43	path error
NET_SDK_ALLOC_RESOURCE_ERROR	44	resources allotting
NET_SDK_AUDIO_MODE_ERROR	45	display card mod
NET_SDK_NOENOUGH_BUF	46	buffer is not enou
NET_SDK_CREATESOCKET_ERROR	47	establish SOCKE
NET_SDK_SETSOCKET_ERROR	48	set SOCKET erro
NET_SDK_MAX_NUM	49	the max number
NET_SDK_USERNOTEXIST	50	user doest not exi
NET_SDK_WRITEFLASHERROR	51	wirte FLASH err
NET_SDK_UPGRADEFAIL	52	failed to upgrade
NET_SDK_CARDHAVEINIT	53	the decode card is
NET_SDK_PLAYERFAILED	54	player failed
NET_SDK_MAX_USERNUM	55	the max user no.
NET_SDK_GETLOCALIPANDMACFAIL	56	failed to get the I end or physical a
NET_SDK_NOENCODEING	57	the channel is not
NET_SDK_IPMISMATCH	58	IP address not ma
NET_SDK_MACMISMATCH	59	MAC address not
NET_SDK_UPGRADELANGMISMATCH	60	the language of u
NET_SDK_MAX_PLAYERPORT	61	reach to the max
NET_SDK_NOSPACEBACKUP	62	no enough space
NET_SDK_NODEVICEBACKUP	63	no backup device
NET_SDK_PICTURE_BITS_ERROR	64	the bits of picture
NET_SDK_PICTURE_DIMENSION_ERROR	65	the dimension is
NET_SDK_PICTURE_SIZ_ERROR	66	the size of picture
NET_SDK_LOADPLAYERSDKFAILED	67	failed to load play
NET_SDK_LOADPLAYERSDKPROC_ERROR	68	not find some fun
NET_SDK_LOADDSSDKFAILED	69	failed to load DsS
NET_SDK_LOADDSSDKPROC_ERROR	70	not find some fun
NET_SDK_DSSDK_ERROR	71	failed to call func
NET_SDK_VOICEMONOPOLIZE	72	voice card is mon
NET_SDK_JOINMULTICASTFAILED	73	failed join to mult
NET_SDK_CREATEDIR_ERROR	74	failed to create lo
NET_SDK_BINDSOCKET_ERROR	75	failed to bind soc
NET_SDK_SOCKETCLOSE_ERROR	76	socket is closed
NET_SDK_USERID_ISUSING	77	the user ID is ope
NET_SDK_PROGRAM_EXCEPTION	78	sdk program exce
NET_SDK_WRITEFILE_FAILED	79	write file failed
NET_SDK_FORMAT_READONLY	80	failed to format re
NET_SDK_WITHSAMEUSERNAME	81	there is same user
NET_SDK_DEVICETYPE_ERROR	82	device type no ma
NET_SDK_LANGUAGE_ERROR	83	language no matc
NET_SDK_PARAVERSION_ERROR	84	soft version no m
NET_SDK_FILE_SUCCESS	85	file has been crea
NET_SDK_FILE_NOFIND	86	file isn't found

NET_SDK_NOMOREFILE	87	there is no more f
NET_SDK_FILE_EXCEPTION	88	file exception
NET_SDK_TRY_LATER	89	Try again later
NET_SDK_DEVICE_OFFLINE	90	Device offline
NET_SDK_CREATEJPEGSTREAM_FAIL	91	Failed to create JI
NET_SDK_USER_ERROR_NO_USER	92	No such user!
NET_SDK_USER_ERROR_USER_OR_PASSWORD_IS_NULL	93	No username or p
NET_SDK_USER_ERROR_ALREDAY_LOGIN	94	The user has beer
NET_SDK_USER_ERROR_SYSTEM_BUSY	95	The device is bus
NET_SDK_DEVICE_NOT_SUPPROT	96	The device don n
NET_SDK_USER_ERROR_SYSTEM_NO_READY	97	Do not complete
NET_SDK_CHANNEL_OFFLINE	98	Camera is offline
NET_SDK_GETREADYINFO_FAIL	99	It fails to get devi
NET_SDK_NORESOURCE	100	SDK resources is
NET_SDK_DEVICE_QUERYSYSTEMCAPS_FAIL	101	The device fails t
NET_SDK_INBUFFER_TOSMALL	102	The input buffer a
NET_SDK_NO_PASSWORD_STRENGTH	103	The password stre

Remarks

Get error number through NET_SDK_GetErrorMsg

See Also

NET_SDK_GetErrorMsg

Client SDK Instructions

NET_SDK_Login

user to register device

```
LONG NET_SDK_Login(
char                      *sDVRIP,
WORD                      wDVRPort,
char                      *sUserName,
char                      *sPassword,
NET_SDK_DEVICEINFO             lpDeviceInfo
);
```

Parameters

sDVRIP
  [in] device IP address
wDVRPort
  [in] number of device port
sUserName
  [in] user name for login
sPassword
  [in] user password
lpDeviceInfo
  [out] device information

Return Values

-1 means failure and other value is the returned ID from user.The ID is unique. Later operations on device realize through this ID. To get error information,please call NET_SDK_GetLastError

Remarks

Device permits 32 registered names,and supports 128 users to register at the same time.

See Also

NET_SDK_Logout

Client SDK Instructions

NET_SDK_Logout

user to logout

```
BOOL NET_SDK_Logout(
LONG    lUserID
);
```

Parameters

lUserID
[in] user ID,return value of NET_SDK_Login

Return Values

TRUE means success; FALSE means failure.To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_LoginEx

user to register device

```
NET_SDK_API LONG CALL_METHOD NET_SDK_LoginEx(
char                       *sDVRIP,
WORD                       wDVRPort,
char                       *sUserName,
char                       *sPassword,
NET_SDK_DEVICEINFO              lpDeviceInfo,
NET_SDK_CONNECT_TYPE              eConnectType,
const char                 *sDevSN
);
```

Parameters

sDVRIP
  [in] device IP address or P2P Server address
wDVRPort
  [in] number of device port or P2P Server port
sUserName
  [in] user name for login
sPassword
  [in] user password
lpDeviceInfo
  [out] device informantion
eConnectType
  [in] Connect type,refer to NET_SDK_CONNECT_TYPE:

Type	Value	Description
NET_SDK_CONNECT_TCP	0	TCP connection
NET_SDK_CONNECT_NAT	1	NAT1.0 connection
NET_SDK_CONNECT_NAT20	2	NAT2.0 connection

sDevSN
  [in] Serial number of P2P. If TCP is connected, ignore this parameter

Return Values

-1 means failure and other value is the returned ID from user.The ID is unique. Later operations on device realize through this ID. To get error information,please callNET_SDK_GetLastError

Remarks

Device permits 32 registered names,and supports 128 users to register at the same time.

See Also

NET_SDK_Logout

Client SDK Instructions

NET_SDK_SetNat2Addr

set the p2p2.0 address

```
LONG NET_SDK_SetNat2Addr(
char                    *sServerAddr,
WORD                      wDVRPort,
);
```

Parameters

sServerAddr
  [in] device IP address （it is the p2p server address（c2020.autonat.com））

wDVRPort
  [in] p2p port

Return Values

TRUE means success; FALSE means failure.To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_LoginEx

Client SDK Instructions

NET_SDK_MakeKeyFrame

one key frame from main code stream dynamically

```
BOOL NET_SDK_MakeKeyFrame(
LONG     lUserID,
LONG     lChannel
);
```

Parameters

lUserID
    [in] return value of NET_SDK_Login
lChannel
    [in] channel number,start from 0

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is set to reset I frame.According to the preview parameter(NET_SDK_CLIENTINFO) type main code stream or sub code stream calls NET_SDK_MakeKeyFrame or NET_SDK_MakeKeyFrameSub to realize resetting I frame.

See Also

NET_SDK_MakeKeyFrameSub

Client SDK Instructions

NET_SDK_MakeKeyFrameEx

one key frame from main code stream dynamically

```
BOOL NET_SDK_MakeKeyFrameEx(
LONG     lUserID,
LONG     lChannel,
unsigned int     streamType
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

lChannel
[in] channel number,start from 0

streamType
[in] stream type

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is set to reset I frame.According to the preview parameter(NET_SDK_CLIENTINFO) type main code stream or sub code stream calls NET_SDK_MakeKeyFrame or NET_SDK_MakeKeyFrameSub to realize resetting I frame.

See Also

NET_SDK_MakeKeyFrameSub

Client SDK Instructions

NET_SDK_MakeKeyFrameSub

one key frame from sub code stream dynamically

```
BOOL NET_SDK_MakeKeyFrameSub(
LONG     lUserID,
LONG     lChannel
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login
lChannel
  [in] channel number starts from 0

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

This interface is set to reset I frame.According to the preview parameter(NET_SDK_CLIENTINFO) type main code stream or sub code stream calls NET_SDK_MakeKeyFrame or NET_SDK_MakeKeyFrameSub to realize resetting I frame.

See Also

NET_SDK_MakeKeyFrame

Client SDK Instructions

NET_SDK_LivePlay

real time preview

```
LONG NET_SDK_LivePlay(
LONG                       lUserID,
LPNET_SDK_CLIENTINFO       lpClientInfo,
LIVE_DATA_CALLBACK         fLiveDataCallBack,
void                       * pUser
);
```

Parameters

lUserID
: [in] return value of NET_SDK_Login()

lpClientInfo
: [in] preview parameter

fLiveDataCallBack
: [in] preview data callback parameter,default value is NULL

** pUser*
: [in] pointer to user,default value is NULL

Return Values

-1 means failure and other value is parameter of handle of function NET_SDK_StopLivePlay. To get error information,please call NET_SDK_GetLastError

Remarks

After calling this interface successfully,if need to capture real time code stream data,call NET_SDK_SetLiveDataCallBack to register the callback function for capturing code stream data,and access the code stream data in the callback function.

See Also

NET_SDK_StopLivePlay

Client SDK Instructions

NET_SDK_StopLivePlay

stop preview

```
BOOL NET_SDK_StopLivePlay(
LONG    lLiveHandle
);
```

Parameters

lLiveHandle
: [in] handle for preview,return value of NET_SDK_LivePlay

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Client SDK Instructions

DRAW_FUN_CALLBACK

data callback when preview

```
void *DRAW_FUN_CALLBACK(
LONG          lLiveHandle,
HDC           hDC,
void          * pUser
);
```

Parameters

lLiveHandle
   [in] preview interface handle
hDC
   [in] device context handle
**pUser*
   [in] pointer to user information

Return Values

None. To get error code,please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_LivePlay_Ex

Real-time Preview

```
LONG NET_SDK_LivePlayEx(
LONG                        lUserID,
  LPNET_SDK_CLIENTINFO      lpClientInfo,
LIVE_DATA_CALLBACK_EX       fLiveDataCallBack,
void                        * pUser
);
```

Parameters

lUserID
  [in] the return value of NET_SDK_Login()
lpClientInfo
  [in] preview parameters
fLiveDataCallBack
  [in] preview data callback parameters, the default value is NULL .
* *pUser*
  [in] user pointer, the default value is NULL.

Return Values

-1 means failure; other values is the handles of the functions, like NET_SDK_StopLivePlay. To get error information, please call NET_SDK_GetLastError

Remarks

Having sucessfully called up this interface, call up the callback function of stream data captured through the registeration of the interface

(NET_SDK_SetLiveDataCallBackEx) to get the real-time stream data.

See Also

NET_SDK_StopLivePlay NET_SDK_GetLivePlayerIndex

Client SDK Instructions

NET_SDK_SupportStreamNum

get the number of streams which support live

```
unsigned int NET_SDK_SupportStreamNum(
LONG                         lUserID,
LONG                         lChannel
);
```

Parameters

lUserID
  [in] the return value of NET_SDK_Login()
lChannel
  [in] the index of channel，from 0

Return Values

return the number of streams which support live。To get error information, please callNET_SDK_GetLastError

Client SDK Instructions

NET_SDK_RegisterDrawFun

overlapping picture of character and image when preview or playback

```
BOOL NET_SDK_RigisterDrawFun(
LONG                 lLiveHandle,
DRAW_FUN_CALLBACK    fDrawFun,
void                 *pUser
);
```

Parameters

lLiveHandle
    [in] return value of NET_SDK_LivePlay
fDrawFun
    [in] callback function for image
pUser
    [in] pointer to user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is mainly used to register callback function,and get current surface device context. User can draw or write on this DC just like on the window's client DC,but this DC is Off-Screen surface DC in player DirectDraw,not window's client DC .

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_SetPlayerBufNumber

set frame buffer area count of broadcast library

```
BOOL NET_SDK_SetPlayerBufNumber(
LONG       lLiveHandle,
DWORD      dwBufNum
);
```

Parameters

lLiveHandle
   [in] return value of NET_SDK_LivePlay
dwBufNum
   [in] the maximum frame count in buffer area when playing one frame by one frame,range[1,50],default value is 15 in SDK

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is called to adjust network delay and play fluency.value of dwBufNum more large,fluency more well,relative delay more large and vice versa. But when network isn't enough well,lost frame makes effect on fluency of broadcast.If current stream is complex,assign buffer frame count equal to or greater than 6 to ensure audio and video effect. This function follows NET_SDK_LivePlay closely,because it's invalid after playing image .

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_OpenSound

open sound in monopolistic sound card mode

```
BOOL NET_SDK_OpenSound(
LONG    lLiveHandle
);
```

Parameters

lLiveHandle
  [in] return value of NET_SDK_LivePlay

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

If current mode is sharing,calling this interface returns failure.The monopolistic mode just opens one channel to playback, when in turn to open multichannel,just opens the last one.

See Also

NET_SDK_LivePlay   NET_SDK_CloseSound

Client SDK Instructions

NET_SDK_Volume

adjust play volume

```
BOOL NET_SDK_Volume(
LONG          lLiveHandle,
WORD          wVolume
);
```

Parameters

lLiveHandle
: [in] return value of NET_SDK_LivePlay

wVolume
: [in] volumeless, range[0,0xffff]

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_CloseSound

close sound under monopolistic sound card mode

```
BOOL NET_SDK_CloseSound(
);
```

Return Values

TRUE means success; FALSE means failure. to get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_OpenSound

Client SDK Instructions

NET_SDK_SetLiveDataCallBack

set preview data callback

```
BOOL NET_SDK_SetLiveDataCallBack(
LONG               lLiveHandle,
LIVE_DATA_CALLBACK fLiveDataCallBack,
void               *pUser
);
```

Parameters

lLiveHandle
  [in] return value of NET_SDK_LivePlay
fLiveDataCallBack
  [in] callback function of code stream
pUser
  [in] user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This function includes start and stop operation on user dispose captured data by SDK,when assigning callback function fLiveDataCallBack other value except NULL, start callback and disposing data,when assigning NULL,stop callback and disposing data.The first package in callback is a file head with 40 bytes,for the later usage in decoding,the next package which to call is compressed code stream.

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_SaveLiveData

save real time preview data

```
BOOL NET_SDK_SaveLiveData(
LONG    lLiveHandle,
char    *sFileName
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
**sFileName*
[in] pointer to file directory

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay   NET_SDK_StopSaveLiveData

Client SDK Instructions

NET_SDK_SetLiveDataCallBackEx

Set the callback of preview data.

```
BOOL NET_SDK_SetLiveDataCallBackEx(
LONG                lLiveHandle,
  LIVE_DATA_CALLBACK_EX fLiveDataCallBack,
void                *pUser
);
```

Parameters

lLiveHandle
  [in] the return value of NET_SDK_LivePlay()
fLiveDataCallBack
  [in] callback function of stream data
pUser
  [in] user data

Return Values

TRUE means success; FALSE means failure. To get error code, please call NET_SDK_GetLastError

Remarks

This function includes start and stop operation on user dispose captured data by SDK,when assigning callback function fLiveDataCallBack other value except NULL, start callback and disposing data,when assigning NULL,stop callback and disposing data.The first package in callback is a file head with 40 bytes,for the later usage in decoding,the next package which to call is compressed code stream.

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_StopSaveLiveData

stop data capture

```
BOOL NET_SDK_StopSaveLiveData(
LONG    lLiveHandle
);
```

Parameters

lLiveHandle
  [in] return value of NET_SDK_LivePlay

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LivePlay   NET_SDK_SaveLiveData

Client SDK Instructions

YUV_DATA_CALLBACK

YUV data cabllback after decoding the captured real-time data

```
void YUV_DATA_CALLBACK(
POINTERHANDLE                   lLiveHandle,
  DECODE_FRAME_INFO           frameInfo,
void                        * pUser
);
```

Parameters

lLiveHandle
: [in] the handle of real-time preview

frameInfo
: [in] YUV data after decoding the frame of real-time preview data

**pUser*
: [in] the pointer of user information

Return Values

No return value. To get error information, please call NET_SDK_GetLastError

Android Interface

Client SDK Instructions

NET_SDK_SetYUVCallBack

Configure the YUV data cabllback after decoding video.

```
BOOL NET_SDK_SetYUVCallBack(
 LONG              lLiveHandle,
   YUV_DATA_CALLBACK fYuvCallBack,
 void             *pUser
 );
```

Parameters

lLiveHandle
   [in] the return value of NET_SDK_LivePlay()
fYuvCallBack
   [in] YUV stream data callback function
pUser
   [in] user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This function includes starting and stopping processing data captured by SDK . When the callback function (fYuvCallback) set to any vaule except null, it means callback and processing data; when it set to null, it means stopping callback and processing data. The callback data are the YUV data after decoding.

See Also

NET_SDK_LivePlay

Android Interface

Client SDK Instructions

LIVE_DATA_CALLBACK

data callback when capture data in living preview

```
void LIVE_DATA_CALLBACK(
LONG                           lLiveHandle,
NET_SDK_FRAME_INFO             frameInfo,
BYTE                           *pBuffer,
void                           * pUser
);
```

Parameters

lLiveHandle
[in] real time preview handle
frameInfo
[in] preview data frame information in real time
**pBuffer*
[in] pointer to buffer area
**pUser*
[in] pointer to user information

Return Values

None. To get error code, please call NET_SDK_GetLastError

Client SDK Instructions

LIVE_DATA_CALLBACK_Ex

Data callback when capturing the real-time preview data.

```
void LIVE_DATA_CALLBACK_Ex(
LONG                          lLiveHandle,
UINT                         dataType,
BYTE                          *pBuffer,
UINT                         dataLen,
void                        * pUser
);
```

Parameters

lLiveHandle
   [in] real-time preview handle
dataType
   [out] data frame type, see DD_FRAME_TYPE
**pBuffer*
   [out] a pointer to the buffer, the contents is
   NET_SDK_FRAME_INFO + FrameData
   dataLen
   [out] the data length of the buffer
**pUser*
   [out] a pointer to a user

Return Values

No return value. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_CapturePicture

capture data one frame by one frame and store to be BMP file.

```
BOOL NET_SDK_CapturePicture(
LONG    lLiveHandle,
char    *sPicFileName
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay()
sPicFileName
[in] directory of storing .BMP image,the length of directory is less than or equals to 256 bytes

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is used to capture present frame of decoding to be .BMP image.

See Also

NET_SDK_LivePlay   NET_SDK_CapturePicture_Other

Client SDK Instructions

NET_SDK_CapturePicture_Other

capture data one frame by one frame and store to be BMP file.

```
BOOL NET_SDK_CapturePicture_Other(
LONG    lUserID,
LONG    lChannel,
char    *sPicFileName
);
```

Parameters

lUserID
  [in] user ID
lChannel
  [in] channel number
sPicFileName
  [in] directory of storing .BMP image,the length of directory is less than or equals to 256 bytes

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is used to capture certain user ID and channel frame of decoding to be .BMP image.

See Also

NET_SDK_CapturePicture

Client SDK Instructions

NET_SDK_CaptureJPEGData_V2

Capture data in JPEG format.

```
BOOL NET_SDK_CaptureJPEGData_V2(
LONG          lUserID,
LONG          lChannel,
char          *sJpegPicBuffer,
DWORD       dwPicSize,
LPDWORD    lpSizeReturned
);
```

Parameters

lUserID
  [in] user ID
lChannel
  [in] channel number
sJpegPicBuffer
  [out] JPEG data buffer
dwPicSize
  [in] sJpegPicBuffer buffer size
lpSizeReturned
  [out] JPEG data size

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is used to capture jpeg data,do not need to open the stream. Only applicable to specific firmware.

See Also

NET_SDK_CaptureJPEGFile_V2

Client SDK Instructions

NET_SDK_CaptureJPEGFile_V2

Capture data in JPEG format and store to be JPEG file.

```
BOOL NET_SDK_CaptureJPEGFile_V2(
LONG    lUserID,
LONG    lChannel,
char    *sPicFileName
);
```

Parameters

lUserID
[in] user ID
lChannel
[in] channel number
**sPicFileName*
[in] save the file path

Return Values

TRUE means success; FALSE means failure. To get error information,please callNET_SDK_GetLastError

Remarks

This interface is used to capture jpeg data and store,do not need to open the stream. Only applicable to specific firmware

See Also

NET_SDK_CaptureJPEGData_V2

Client SDK Instructions

NET_SDK_CaptureJPEGPicture

Capture data in JPEG format and store to be JPEG file.

```
BOOL NET_SDK_CaptureJPEGFile_V2(
LONG    lUserID,
LONG    lChannel,
LPNET_SDK_JPEGPARA lpJpegPara,
char *sJpegPicBuffer ,
DWORD dwPicSize,
LPDWORD lpSizeReturned
);
```

Parameters

lUserID
    [in] user ID
lChannel
    [in] channel number
**LPNET_SDK_JPEGPARA*
    [in] a command type
sJpegPicBuffer
    [in]JPEG data buffer
dwPicSize
    [in]sJpegPicBuffer buffer size
lpSizeReturned
    [in]JPEG data size

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is used to capture jpeg data and store,do not need to open the stream. Only applicable to specific firmware

See Also

NET_SDK_CaptureJPEGData_V2

Android Interface

The corresponding Android interface：no

NET_SDK_RemoteSnap

Caputre images by controlling devices remotely. The images are saved in the device end.(only N9000 availabe）

```
BOOL NET_SDK_RemoteSnap(
LONG    lUserID,
int   lChannel,
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
lChannel
[in] Channel number（starting with 0）

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

typedef struct _net_sdk_image_sreach

```
{
    DWORD                           dwChannel;//Snapshot channel (starting with 0)
    DD_TIME                         StartTime; //time
    DD_TIME                         StopTime; //time
    DWORD                           pageIndex;//Page number
    DWORD                           pageSize;//Page
    IMAGE_SORT_TYPE sort; //return
    unsigned char       resv[8];
}NET_SDK_IMAGE_SREACH;

typedef struct _net_sdk_image_
{
    DWORD                           dwChannel; //Snapshot cahnnel
    DWORD                           dwImageType; //Snapshot type IMAGE_EVENT_TYPE
    DD_TIME                         captureTime;//Snapshot time
    DWORD                           totalNum;//Totol number
    unsigned char       resv[8];
}NET_SDK_IMAGE;
```

NET_SDK_SearchPictures

Get the image list of the remote device.(only N9000 device availabe)

```
BOOL NET_SDK_SearchPictures(
LONG lUserID,
NET_SDK_IMAGE_SREACH sInSreachImage,
LONG lInImageBufferSize,
NET_SDK_IMAGE *pOutImageInfo,
LONG *pOutImageNum
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login()
sInSreachImage
[in] Search condition
lInImageBufferSize
[in] the space of pOutImageInfo applied for
pOutImageInfo
[out] Return image information
pOutImageNum
[out] the number of returning to image information

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

This interface can realize JPEG snapshot and save the files without enabling stream. It is only available for specified devices.

See Also

NET_SDK_CaptureJPEGData_V2

Client SDK Instructions

//Image Type

```
enum IMAGE_MODE
{
    IMAGE_MODE_JPG,
    IMAGE_MODE_PNG,
    IMAGE_MODE_BMP,
};
typedef struct _net_sdk_image_info
{
    unsigned int        imageSize;
    IMAGE_MODE          imageMode; //Image format
    unsigned char       resv[8];
}NET_SDK_IMAGE_INFO;
```

NET_SDK_DownLoadPicture

Uploading remote images（only N9000 devices availabe）

```
 BOOL NET_SDK_DownLoadPicture(
LONG lUserID,
NET_SDK_IMAGE captureImage,
NET_SDK_IMAGE_INFO *pOutImageInfo,
char* pOutBuffer,
int outBufferSize
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login()
captureImage
  [in] NET_SDK_SearchPictures returns a data of the searched image list
pOutImageInfo
  [out] return the uploaded image information
pOutBuffer
  [out] return the image data（when the current image exists and the space requested is larger than or equal to the image size, it takes effect)
outBufferSize
  [int] buffer area size of pOutBuffer requesting

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Two conditions to return successfully：

1、When the image exsits and the size of outBufferSize is less than the image size，the return will be successful and the upper application can decode the image size from pOutImageInfo parameters.
2、When the image exsits and the size of outBufferSize is larger than or equal to the downloaded image size, the return value of pOutBuffer is image data.

Client SDK Instructions

NET_SDK_FindFile

find record file by time

```
LONG NET_SDK_FindFile(
LONG                  lUserID,
LONG                  lChannel,
DD_TIME               lpStartTime,
DD_TIME               lpStopTime
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login()
lChannel
  [in] channel number,start from 0
lpStartTime
  [in] the start time of file
lpStopTime
  [in] the stop time of file

Return Values

-1 means failure and other value is a parameter of function NET_SDK_FindClose. To get error information,please call NET_SDK_GetLastError

Remarks

This interface specifies the type and time range of finding record file,after calling the interface,call NET_SDK_FindNextFile to get file information.

See Also

NET_SDK_FindNextFile NET_SDK_FindClose

Client SDK Instructions

NET_SDK_FindNextFile

get file information one by one

```
LONG NET_SDK_FindNextFile(
LONG                      lFindHandle,
NET_SDK_REC_FILE          lpFindData
);
```

Parameters

lFindHandle
    [in] handle of finding file,return value of NET_SDK_FindFile()
lpFindData
    [out] pointer to store file information

Return Values

-1 means failure and other value is current state information. To get error information,please call NET_SDK_GetLastError

Remarks

Before calling this interface,call NET_SDK_FindFile to get current handle for finding

See Also

NET_SDK_FindFile

Client SDK Instructions

NET_SDK_FindClose

close finding filename, free resource

```
BOOL NET_SDK_FindClose(
LONG   lFindHandle
);
```

Parameters

lFindHandle
[in] return handle of finding file in function NET_SDK_FindFile

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindFile

Client SDK Instructions

NET_SDK_FindRecDate

find record file by data

```
LONG NET_SDK_FindRecDate(
LONG        lUserID
);
```

Parameters

lUserID
    [in] user ID

Return Values

-1 means failure and other value is the return information of finding. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindRecDateClose   NET_SDK_FindNextRecDate

Client SDK Instructions

NET_SDK_FindNextRecDate

get record file one by one

```
LONG NET_SDK_FindNextRecDate(
LONG           lFindHandle,
DD_DATE        *lpRecDate
);
```

Parameters

lFindHandle
    [in] handle of finding
**lpRecDate*
    [in] date of record

Return Values

-1 means failure and other value is the return value of finding. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindRecDate   NET_SDK_FindRecDateClose

Client SDK Instructions

NET_SDK_FindRecDateClose

close finding record file by date,free resource

```
BOOL NET_SDK_FindRecDateClose(
LONG       lFindHandle
);
```

Parameters

lFindHandle
    [in] handle of finding

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindRecDate   NET_SDK_FindNextRecDate

Client SDK Instructions

NET_SDK_FindEvent

find record file by event

```
LONG NET_SDK_FindEvent(
LONG                    lUserID,
LONG                    lChannel,
DWORD                   dwRecType,
DD_TIME                 lpStartTime,
DD_TIME                 lpStopTime
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login()
lChannel
  [in] channel number,start from 0
dwRecType
  [in] type of event
lpStartTime
  [in] pointer to the start time of file
lpStopTime
  [in] pointer to the stop time of file

Return Values

-1 means failure and other value is parameter of function NET_SDK_FindNextEvent. To get error information,please call NET_SDK_GetLastError

Remarks

This interface specifies information of finding record file(by event),after calling the interface,call

NET_SDK_FindNextEvent to get file information. The record file which is found by event aims at the start time and stop time,so it only supports playback by time.

See Also

NET_SDK_FindNextEvent NET_SDK_FindEventClose

Client SDK Instructions

NET_SDK_FindNextEvent

get information of found file one by one

```
LONG NET_SDK_FindNextEvent(
LONG                        lFindHandle,
NET_SDK_REC_EVENT           *lpRecEvent
);
```

Parameters

lFindHandle
    [in] handle of finding file,return value of NET_SDK_FindEvent()
**lpRecEvent*
    [out] pointer to store file information

Return Values

-1 means failure and other value is current state information. To get error information,please call NET_SDK_GetLastError

Remarks

Before calling this interface,call NET_SDK_FindEvent to get current handle for finding.The record file found by event aims at the start time and stop time,so only supports playback by time.

See Also

NET_SDK_FindEvent

Client SDK Instructions

NET_SDK_FindEventClose

close finding file by event,free resource

```
BOOL NET_SDK_FindEventClose(
LONG       lFindHandle
);
```

Parameters

lFindHandle
  [in] handle of query

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindEvent   NET_SDK_FindNextEvent

Client SDK Instructions

NET_SDK_FindTime

find record file by time

```
LONG NET_SDK_FindTime(
LONG        lUserID,
LONG        lChannel,
DD_TIME     * lpStartTime,
DD_TIME     * lpStopTime
);
```

Parameters

lUserID
  [in] user ID
lChannel
  [in] channel number,start from 0
** lpStartTime*
  [in] pointer to the start time
** lpStopTime*
  [in] pointer to the stop time

Return Values

-1 means failure and other value is the result of finding. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindNextTime   NET_SDK_FindTimeClose

Client SDK Instructions

NET_SDK_FindNextTime

get record file one by one

```
LONG NET_SDK_FindNextTime(
LONG                lFindHandle,
NET_SDK_REC_TIME    *lpRecTime
);
```

Parameters

lFindHandle
  [in] handle of finding
**lpRecTime*
  [in] time of record

Return Values

-1 means failure and other value is found information. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindTime   NET_SDK_FindTimeClose

Client SDK Instructions

NET_SDK_FindTimeClose

close finding record file by time,free resource

```
BOOL NET_SDK_FindTimeClose(
LONG        lFindHandle
);
```

Parameters

lFindHandle
    [in] handle of finding

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FindTime   NET_SDK_FindNextTime

Client SDK Instructions

NET_SDK_PlayBackByTime

Play back the record file by time and the main stream is requested.

```
LONG NET_SDK_PlayBackByTime(
LONG                 lUserID,
LONG                 *pChannels,
LONG                 channelNum,
DD_TIME              *lpStartTime,
DD_TIME              *lpStopTime,
HWND                 *hWnd
);
```

Parameters

lUserID
: [in] return value of NET_SDK_Login

pChannels
: [in] channel number,start from 0 array for playback

channelNum
: [in] quatity of channels in array **pChannels**

**lpStartTime*
: [in] pointer to the start time of the file

**lpStopTime*
: [in] pointer to the stop time of the file

**hWnd*
: [in] window handle for playback,if null,SDK still can receive code stream data,but can't decode to display.

Return Values

-1 means failure and other value is parameter of NET_SDK_StopPlayBack. To get error information,please call NET_SDK_GetLastError

Remarks

This interface specifies which record file to play,after finishing calling this interface,call NET_SDK_SetPlayDataCallBack to register callback function and dispose the captured code stream data by itself.

See Also

NET_SDK_PlayBackControl NET_SDK_StopPlayBack NET_SDK_SetLiveDataCallBack NET_SDK_Login

Client SDK Instructions

NET_SDK_PlayBackControl

control the state of playback

```
BOOL NET_SDK_PlayBackControl(
LONG      lPlayHandle,
DWORD     dwControlCode,
DWORD     dwInValue,
DWORD     *lpOutValue
);
```

Parameters

lPlayHandle
[in] play handle,return value of NET_SDK_PlayBackByTime
dwControlCode
[in] command of controlling the state of playback,refer to NET_SDK_PLAYCTRL_TYPE:

Type	Description
NET_SDK_PLAYCTRL_PAUSE	pause
NET_SDK_PLAYCTRL_FF	fast forward
NET_SDK_PLAYCTRL_REW	rewind
NET_SDK_PLAYCTRL_RESUME	resume
NET_SDK_PLAYCTRL_STOP	stop
NET_SDK_PLAYCTRL_FRAME	play one frame
NET_SDK_PLAYCTRL_NORMAL	normal play
NET_SDK_PLAYCTRL_STARTAUDIO	enable audio,choose channel by parameter *dwInValue* of function NET_SDK_PlayBackControl
NET_SDK_PLAYCTRL_STOPAUDIO	stop audio
NET_SDK_PLAYCTRL_AUDIOVOLUME	adjust audio volume,realize by parameter *dwInValue* of function NET_SDK_PlayBackControl
NET_SDK_PLAYCTRL_SETPOS	play progress,calculate seconds from January 1st

	in 1970

dwInValue
[in] parameter setting,when setting playback progress this parameter means progress value; when playing this parameter means file location of resuming from break point.

lpOutValue
[out] got parameter,if need to get total time of current play file,this parameter meets.

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Whether assign the third parameter of this interface depends on the control command,under NET_SDK_PLAYSETPOS command this parameter means the playback progress; when starting the control command it means offset of current file;when its value is 0 it means to play from the starting location,if not 0 ,it means the file location of resuming from break point.
The fourth parameter means the parameter got from current control command operation.

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_StopPlayBack

stop record file playback

```
BOOL NET_SDK_StopPlayBack(
LONG    lPlayHandle
);
```

Parameters

lPlayHandle
[in] handle for playback,return value of NET_SDK_PlayBackByTime

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

PLAY_DATA_CALLBACK

data callback when finding file and playback

```
void *PLAY_DATA_CALLBACK(
LONG                       lPlayHandle,
NET_SDK_FRAME_INFO         frameInfo,
BYTE                       *pBuffer,
void                       *pUser
);
```

Parameters

lPlayHandle
  [in] play handle
frameInfo
  [in] file data playback code stream frame information
**pBuffer*
  [in] buffer pointer to find files and playback
** pUser*
  [in] pointer to user information

Return Values

None. To get error code, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_PlayBackByTimeEx

Play back the record files by time.

```
LONG NET_SDK_PlayBackByTimeEx(
LONG            lUserID,
LONG            *pChannels,
LONG            channelNum,
  DD_TIME           *lpStartTime,
  DD_TIME           *lpStopTime,
HWND             *hWnds
BOOL            bFirstStream
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()

pChannels
[in]   the channel number group of playback

channelNum
[in] the channel quantities of pChannels

**lpStartTime*
[in] a pointer to start time of the file

**lpStopTime*
[in] a pointer to end time of the file

**hWnds*
[in] the handle of playback window, if it is null, SDK still can receive the stream data but not decoding.

bFirstStream
[in] Whether to play back the main stream. false is sub-stream.

Return Values

-1 means failure; other vallues is the parameters of the functions, like NET_SDK_StopPlayBack. To get error information, please call NET_SDK_GetLastError

Remarks

This interface specifies the record files needed to play. After successfully call up this interface, please register callback function by the interface of NET_SDK_SetPlayDataCallBack to capture the recording stream data and dispose the captured code stream data by itself.

See Also

NET_SDK_PlayBackControl NET_SDK_StopPlayBack
NET_SDK_SetLiveDataCallBack NET_SDK_Login

Client SDK Instructions

NET_SDK_SetPlayDataCallBack

register callback function,capture record data

```
BOOL NET_SDK_SetPlayDataCallBack(
LONG                  lPlayHandle,
PLAY_DATA_CALLBACK    fPlayDataCallBack,
void                  *pUser
);
```

Parameters

lPlayHandle
[in] play handle,return value of NET_SDK_PlayBackByTime

fPlayDataCallBack
[in] callback function of record data

pUser
[in] user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This function includes start and stop operation on user dispose captured data by SDK,when assigning callback function fLiveDataCallBack other value except NULL, start callback and disposing data,if assign NULL,stop callback and disposing data.

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_SetPlayYUVCallBack

(only in windows now)register callback function,capture record data.it can receive YUV data from the callback function,you should return quickly if you need to process the data.

```
BOOL NET_SDK_SetPlayYUVCallBack(
LONG                    lPlayHandle,
  PLAY_YUV_DATA_CALLBACK  fYuvDataCallBack,
void                  *pUser
);
```

Parameters

lPlayHandle
  [in] play handle,return value of NET_SDK_PlayBackByTime
fYuvDataCallBack
  [in] callback function of record data
pUser
  [in] user data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

This function includes start and stop operation on user dispose captured data by SDK,when assigning callback function *fYuvDataCallBack* other value except NULL, start callback and disposing data,if assign NULL,stop callback and disposing data.

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_PlayBackSaveData

capture record data of playback and store them into file

```
BOOL NET_SDK_PlayBackSaveData(
LONG    lPlayHandle,
LONG    lChannel,
char    *sFileName
);
```

Parameters

lPlayHandle
: [in] play handle,return value of NET_SDK_PlayBackByTime

lChannel
: [in] channel number,start from 0

**sFileName*
: [in] directory of storing data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_StopPlayBackSave

stop saving record data

```
BOOL NET_SDK_StopPlayBackSave(
LONG     lPlayHandle,
LONG     lChannel
);
```

Parameters

lPlayHandle
[in] handle for play,return value of NET_SDK_PlayBackByTime

lChannel
[in] channel number,start from 0

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_GetPlayBackOsdTime

get OSD time when playback

```
BOOL NET_SDK_GetPlayBackOsdTime(
LONG            lPlayHandle,
DD_TIME         lpOsdTime
);
```

Parameters

lPlayHandle
  [in] player handle,return value of NET_SDK_PlayBackByTime
**lpOsdTime*
  [out] pointer to OSD time

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_RefreshPlay

refresh window to display playback

```
BOOL NET_SDK_RefreshPlay(
LONG    lPlayHandle
);
```

Parameters

lPlayHandle
: [in] playback handle,return value of NET_SDK_PlayBackByTime

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

When pause or playback in one frame,if refresh the window,the image disappears ,at that time call this interface to display,this interface is valid just when pause and playback in one frame.

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_PlayBackCaptureFile

capture when playback and store them to file

```
BOOL NET_SDK_PlayBackCaptureFile(
LONG    lPlayHandle,
LONG    lChannel,
char    *sFileName
);
```

Parameters

lPlayHandle
: [in] play handle,return value of NET_SDK_PlayBackByTime

lChannel
: [in] channel number,start from 0

**sFileName*
: [in] directory of storing picture data

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackByTime

Client SDK Instructions

NET_SDK_GetFileByTime

download record file by time

```
LONG NET_SDK_GetFileByTime(
LONG                  lUserID,
LONG                  lChannel,
DD_TIME               *lpStartTime,
DD_TIME               *lpStopTime,
char                  *sSavedFileName
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login
lChannel
  [in] channel number,start from 0
**lpStartTime*
  [in] pointer to the start time
**lpStopTime*
  [in] pointer to the stop time
**sSavedFileName*
  [in] directory of storing downloaded file to PC

Return Values

-1 means failure and other value is parameter of NET_SDK_StopGetFile. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackControl   NET_SDK_StopGetFile
NET_SDK_Login

Client SDK Instructions

NET_SDK_GetDownloadPos

get current progress of downloading record file

```
int NET_SDK_GetDownloadPos(
LONG    lFileHandle
);
```

Parameters

lFileHandle
[in] handle for downloading,return value of NET_SDK_GetFileByTime()

Return Values

-1 means failure;0-100 means the progress of downloading;100 means finish;normal range is 0-100, if the return value is 200,network is unusual. To get error information,please call NET_SDK_GetLastError

Remarks

This interface is to get progress of downloading record file by file name.

See Also

NET_SDK_GetFileByTime

Client SDK Instructions

NET_SDK_StopGetFile

stop downloading record file

```
BOOL NET_SDK_StopGetFile(
LONG    lFileHandle
);
```

Parameters

lFileHandle
: [in] handle for download,return value of NET_SDK_GetFileByTime

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_GetFileByTime

Client SDK Instructions

NET_SDK_GetFileByTimeEx

Extended interface of download record file by time

```
LONG NET_SDK_GetFileByTime(
LONG                  lUserID,
LONG                  lChannel,
DD_TIME               *lpStartTime,
DD_TIME               *lpStopTime,
char                  *sSavedFileName
BOOL                  bCustomFormat
BOOL                  bUseCallBack
 BACKUP_DATA_CALLBACK     fBackupDataCallBack
void     *pUser
)
```

Parameters

lUserID
  [in] return value of NET_SDK_Login
lChannel
  [in] the channel number starts from 0
**lpStartTime*
  [in] a pointer to the start time
**lpStopTime*
  [in] a pointer to the end time
**sSavedFileName*
  [in] the directory of storing downloaded file to PC
bCustomFormat
  [in] whether to use private protocl format. TRUE means private protocol format
bUseCallBack
  [in] whether to use call back function，TRUE means use
fBackupDataCallBack

[in] call back function

pUser

[in] user's pointer，default is NULL

Return Values

-1 means failure and other value is parameter of NET_SDK_StopGetFile. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackControl   NET_SDK_StopGetFile
NET_SDK_Login

Client SDK Instructions

NET_SDK_GetFileByTimeExV2

Extended interface of download record file by time version 2

```
LONG NET_SDK_GetFileByTimeExV2(
LONG                 lUserID,
LONG                 lChannel,
DD_TIME              *lpStartTime,
DD_TIME              *lpStopTime,
char                 *sSavedFileName
char                 recFormat
BOOL                 bFirstStream
BOOL                 bUseCallBack
 BACKUP_DATA_CALLBACK    fBackupDataCallBack
void     *pUser
)
```

Parameters

lUserID
[in] eturn value of NET_SDK_Login
lChannel
[in] the channel number starts from 0
**lpStartTime*
[in] a pointer to the start time
**lpStopTime*
[in] a pointer to the end time
**sSavedFileName*
[in] the directory of storing downloaded file to PC
recFormat
[in] whether to use private protocl format. 1 means private protocol format
bFirstStream
[in] whether to download first stream, TRUE means yes

bUseCallBack
  [in] whether to use call back function，TRUE means use
fBackupDataCallBack
  [in] call back function
**pUser*
  [in] user's pointer，default is NULL

Return Values

-1 means failure and other value is parameter of NET_SDK_StopGetFile. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PlayBackControl NET_SDK_StopGetFile
NET_SDK_Login

BACKUP_DATA_CALLBACK

Call back the data when downloading the records.

```
void *PLAY_DATA_CALLBACK(
LONG                      lFileHandle,
UINT                      dataType,
BYTE                      *pBuffer,
UINT                      dataLen,
void                      *pUser
);
```

Parameters

lFileHandle
  [in] download file handle
dataType
  [in] data type
**pBuffer*
  [in] File download stream and frame information
dataLen
  [in] data length
** pUser*
  [in] a pointer to user information

Return Values

No return value. To get error information, please callNET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SaveFileToUsbByTime

Save the record to the USB of the device（support only N9000）。

```
BOOL NET_SDK_SaveFileToUsbByTime(
LONG                  lUserID,
NET_SDK_REC_FILE              *recordFile,
USB_BACKUP_FORMAT              recFormat
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login.
**recordFile*
[in] the pointer of the record file
recFormat
[in] 0 is avi，1 is private format

Return Values

FALSE means failed，TRUE means success。To get error information, please callNET_SDK_GetLastError

See Also

NET_SDK_GetSaveFileToUsbProcess

Client SDK Instructions

NET_SDK_GetSaveFileToUsbProcess

Get the process and status of the saving record to USB device（support only N9000）。

```
 BOOL NET_SDK_GetSaveFileToUsbProcess(
 LONG              lUserID,
 NET_SDK_USB_BACKUP_PROCESS_EX          *pUsbBackProcess,
unsigned int              lBuffSize,
unsigned int              *taskCount
 );
```

Parameters

lUserID
[in] the return value of NET_SDK_Login
**pUsbBackProcess*
[in] the pointer of the results
lBuffSize
[in] the expect number of the result
**taskCount*
[in] the actual number of the result

Return Values

FALSE means failed，TRUE means success。To get error information, please callNET_SDK_GetLastError

See Also

NET_SDK_SaveFileToUsbByTime

Client SDK Instructions

NET_SDK_LockFile

lock record file(only support IPC)

```
BOOL NET_SDK_LockFile(
LONG                lUserID,
NET_SDK_REC_FILE    *pFileTolock,
LONG                fileNum
);
```

Parameters

lUserID
    [in] user ID
pFileTolock
    [in] pointer to lock file
fileNum
    [in] quantity of file

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_UnlockFile

Client SDK Instructions

NET_SDK_UnlockFile

unlock record file(only support IPC)

```
BOOL NET_SDK_UnlockFile(
LONG                 lUserID,
NET_SDK_REC_FILE *pFileToUnlock,
LONG                 fileNum
);
```

Parameters

lUserID
: [in] user ID

pFileToUnlock
: [in] pointer to unlock file

fileNum
: [in] quantity of files

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_LockFile

Client SDK Instructions

NET_SDK_DeleteRecFile

delete recorded file(only support 3.0DVR)

```
BOOL NET_SDK_DeleteRecFile(
LONG                lUserID,
NET_SDK_REC_FILE    *pFileToUnlock,
LONG                fileNum
);
```

Parameters

lUserID
  [in] user ID
**pFileToUnlock*
  [in] pointer to unlock file
fileNum
  [in] number of files

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_StartDVRRecord

start device to record manually and remotely(only support N9000)

```
BOOL NET_SDK_StartDVRRecord(
LONG     lUserID,
LONG     lChannel,
LONG     lRecordType
);
```

Parameters

lUserID
: [in] return value of NET_SDK_Login

lChannel
: [in] channel number,start from 0,0x00ff means all analog channels,0xff00 means all digital channels,0xffff means all analog channels and digital channels

lRecordType
: [in] record type

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_StopDVRRecord

Client SDK Instructions

NET_SDK_StopDVRRecord

stop device record manually and remotely(only support N9000)

```
BOOL NET_SDK_StopDVRRecord(
LONG     lUserID,
LONG     lChannel
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] channel number,start from 0,0x00ff means all analog channels,0xff00 means all digital channels,0xffff means all analog channels and digital channels

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_StartDVRRecord

Client SDK Instructions

NET_SDK_SetDVRMessageCallBack

register callback function,receive alarm information from device etc.

```
BOOL NET_SDK_SetDVRMessCallBack(
NET_MESSAGE_CALLBACK    fMessCallBack
void                    *pUser
);
```

Parameters

fMessCallBack
[in] callback function

pUser
[in] user information

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_SetupAlarmChan

Client SDK Instructions

NET_MESSAGE_CALLBACK

data callback when alarm

```
BOOL NET_MESSAGE_CALLBACK(
LONG          lCommand,
LONG          lUserID,
char          *pBuf,
DWORD         dwBufLen,
void          * pUser
);
```

Parameters

lCommand
[in] command handle,refer to the list below:

Type	Description
NET_SDK_ALARM	device alarm information
NET_SDK_RECORD	device record information
NET_SDK_IVM_RULE	Intelligent behavior analysis information(reserved)
NET_SDK_TRADEINFO	ATM trade information(reserved)
NET_SDK_IPCCFG	IPC information change of mixed DVR(reserved)

lUserID
[in] user ID

**pBuf*

[in] pointer to buffer area,when **lCommand** value is NET_SDK_ALARM **pBuf** is array of struct NET_SDK_ALARMINFO,when **lCommand** value is NET_SDK_RECORD **pBuf** is array of struct NET_SDK_RECORD_STATUS

dwBufLen

[in] length of buffer area

** pUser*

[in] pointer to user information

Return Values

None. To get error code, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetDVRMessageCallBackEx

register callback function,receive alarm information from device etc.

```
BOOL NET_SDK_SetDVRMessageCallBackEx(
NET_MESSAGE_CALLBACK_EX    fMessCallBack
void                      *pUser
);
```

Parameters

fMessCallBack
  [in] callback function
pUser
  [in] user information

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_SetupAlarmChan

Client SDK Instructions

NET_MESSAGE_CALLBACK_EX

data callback when alarm

```
BOOL NET_MESSAGE_CALLBACK_EX(
LONG              lCommand,
LONG              lUserID,
char              *pBuf,
DWORD             dwBufLen,
void              * pUser
);
```

Parameters

lCommand
[in] command handle,refer to the list below:

Type	Description
NET_SDK_ALARM	device alarm information
NET_SDK_RECORD	device record information
NET_SDK_IVM_RULE	Intelligent behavior analysis information(reserved)
NET_SDK_TRADEINFO	ATM trade information(reserved)
NET_SDK_IPCCFG	IPC information change of mixed DVR(reserved)

lUserID
[in] user ID

**pBuf*

[in] pointer to buffer area,when **lCommand** value is NET_SDK_ALARM **pBuf** is array of struct NET_SDK_ALARMINFO_EX ,when **lCommand** value is NET_SDK_RECORD **pBuf** is array of struct NET_SDK_RECORD_STATUS_EX

dwBufLen

[in] length of buffer area

** pUser*

[in] pointer to user information

Return Values

None. To get error code, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetupAlarmChan

build uploading channel for alarm,get alarm information

```
LONG NET_SDK_SetupAlarmChan(
LONG   lUserID
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

Return Values

-1 means failure and other value is handle parameter of NET_SDK_CloseAlarmChan. To get error information, please call NET_SDK_GetLastError

Remarks

Start arming and then call the interface of register callback function(just call NET_SDK_SetDVRMessCallBack when need)to get uploading information.

See Also

NET_SDK_CloseAlarmChan NET_SDK_Login
NET_SDK_SetDVRMessCallBack

Client SDK Instructions

NET_SDK_CloseAlarmChan

cancel channel of uploading alarm.

```
BOOL NET_SDK_CloseAlarmChan(
LONG   lAlarmHandle
);
```

Parameters

lAlarmHandle
[in] return value of NET_SDK_SetupAlarmChan()

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_SetupAlarmChan

Client SDK Instructions

NET_SDK_PTZControl

PTZ control operation(need to start previewing image)

```
BOOL NET_SDK_PTZControl(
LONG      lLiveHandle,
DWORD     dwPTZCommand,
DWORD     dwSpeed
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
dwPTZCommand
[in] PTZ control command,list as follows:

macro definition	value of maro definition	meaning
PTZ_CMD_STOP	0	PTZ is stopped
PTZ_CMD_LEFT	1	PTZ turns left
PTZ_CMD_RIGHT	2	PTZ turns right
PTZ_CMD_UP	3	PTZ turns pitch up
PTZ_CMD_DOWN	4	PTZ turns pitch under
PTZ_CMD_LEFT_UP	5	PTZ turns upleft
PTZ_CMD_LEFT_DOWN	6	PTZ turns

		downleft
PTZ_CMD_RIGHT_UP	7	PTZ turns upright
PTZ_CMD_RIGHT_DOWN	8	PTZ turns downright
PTZ_CMD_NEAR	9	adjust focus fore
PTZ_CMD_FAR	10	adjust focus aft
PTZ_CMD_ZOOM_OUT	11	focus length decreases
PTZ_CMD_ZOOM_IN	12	focus length increases
PTZ_CMD_IRIS_OPEN	13	open aperture
PTZ_CMD_IRIS_CLOSE	14	close aperture
PTZ_CMD_RESET	0xF0	reset PTZ state

dwSpeed

[in] speed of PTZ ,list as follows:

macro definition	**value of macro definition**
PTZ_SPEED_1	1
PTZ_SPEED_2	2
PTZ_SPEED_3	3
PTZ_SPEED_4	4
PTZ_SPEED_5	5

PTZ_SPEED_6	6
PTZ_SPEED_7	7
PTZ_SPEED_8	8

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

Each command operated on PTZ needs to call this interface twice,start command and stop command,which command to call depends on the parameter *dwPTZCommand*. Before calling this interface preview should be open.Each operation on PTZ corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

PTZ default speed is the top speed.

See Also

NET_SDK_LivePlay

Android Interface

The corresponding Android interface

```
boolean   PTZControl(
long      handle,
int       command,
int       speed
);
```

Client SDK Instructions

NET_SDK_PTZControl_Other

PTZ control operation(no need to start image preview)

```
BOOL NET_SDK_PTZControl_Other(
LONG     lUserID,
LONG     lChannel,
DWORD    dwPTZCommand,
DWORD    dwSpeed
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] channel number,start from 0
dwPTZCommand
[in] PTZ control command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_STOP	0	PTZ is stopped
PTZ_CMD_LEFT	1	PTZ turns left
PTZ_CMD_RIGHT	2	PTZ turns right
PTZ_CMD_UP	3	PTZ turns pitch up
PTZ_CMD_DOWN	4	PTZ turns pitch under

PTZ_CMD_LEFT_UP	5	PTZ turns upleft
PTZ_CMD_LEFT_DOWN	6	PTZ turns downleft
PTZ_CMD_RIGHT_UP	7	PTZ turns upright
PTZ_CMD_RIGHT_DOWN	8	PTZ turns downright
PTZ_CMD_NEAR	9	adjust focus fore
PTZ_CMD_FAR	10	adjust focus aft
PTZ_CMD_ZOOM_OUT	11	focus length decreases
PTZ_CMD_ZOOM_IN	12	focus length increases
PTZ_CMD_IRIS_OPEN	13	open aperture
PTZ_CMD_IRIS_CLOSE	14	close aperture
PTZ_CMD_RESET	0xF0	reset PTZ state

dwSpeed
[in] speed of PTZ ,list as follows:

macro definition	**value of macro definition**
PTZ_SPEED_1	1
PTZ_SPEED_2	2

PTZ_SPEED_3	3
PTZ_SPEED_4	4
PTZ_SPEED_5	5
PTZ_SPEED_6	6
PTZ_SPEED_7	7
PTZ_SPEED_8	8

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

Each command operated on PTZ needs to call this interface twice,start command and stop command,which command to call depends on the parameter *dwPTZCommand*. Before calling this interface preview should be open.Each operation on PTZ corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

PTZ default speed is the top speed.

See Also

NET_SDK_LivePlay

Android Interface

The corresponding Android interface

```
boolean   PTZControl_Other(
long      userId,
long      channel,
int       command,
int       speed
);
```

Client SDK Instructions

NET_SDK_PTZPreset

PTZ preset point operation(need to open preview)

```
BOOL NET_SDK_PTZPreset(
LONG      lLiveHandle,
DWORD     dwPTZPresetCmd,
DWORD     dwPresetIndex
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
dwPTZPresetCmd
[in] operate PTZ preset point command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_PRESET_SET	16	PTZ sets preset points
PTZ_CMD_PRESET_GO	17	to appointed preset point
PTZ_CMD_PRESET_DEL	18	delete preset points

dwPresetIndex
[in] serial number of preset point,at most support 255 points

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_PTZPreset_Other

PTZ preset point operation

```
BOOL NET_SDK_PTZPreset_Other(
LONG      lUserID,
LONG      lChannel,
DWORD     dwPTZPresetCmd,
DWORD     dwPresetIndex
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] channel number,start from 0
dwPTZPresetCmd
[in] operate PTZ preset point command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_PRESET_SET	16	PTZ sets preset points
PTZ_CMD_PRESET_GO	17	to appointed preset point
PTZ_CMD_PRESET_DEL	18	delete preset points

dwPresetIndex

[in] serial number of preset point,at most support 255 points

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_PTZSetCruise

set PTZ cruise line

```
BOOL NET_SDK_PTZSetCruise(
LONG                    lLiveHandle,
BYTE                    byCruiseRoute,
DD_CRUISE_POINT_INFO    *pCruisePoint,
WORD                    pointNum
);
```

Parameters

lLiveHandle
: [in] play handle

byCruiseRoute
: [in] cruise line

**pCruisePoint*
: [in] cruise point

pointNum
: [in] count of cruise point

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZSetCruise_Other

Client SDK Instructions

NET_SDK_PTZSetCruise_Other

set PTZ cruise line operation

```
BOOL NET_SDK_PTZSetCruise_Other(
LONG                        lUserID,
LONG                        lChannel,
BYTE                        byCruiseRoute,
DD_CRUISE_POINT_INFO        *pCruisePoint,
WORD                        pointNum
);
```

Parameters

lUserID
    [in] user ID
lChannel
    [in] channel number,start from 0
byCruiseRoute
    [in] cruise line
**pCruisePoint*
    [in] cruise point
pointNum
    [in] count of cruise point

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZSetCruise

Client SDK Instructions

NET_SDK_PTZCruise

PTZ cruise operation,need to start preview

```
BOOL NET_SDK_PTZCruise(
LONG      lLiveHandle,
DWORD     dwPTZCruiseCmd,
BYTE      byCruiseRoute,
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
dwPTZCruiseCmd
[in] operate PTZ cruise command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_CRUISE_CFG	19	set cruise line,amount to execute Enter,Set and Leave commands
PTZ_CMD_ENTER_CURISE_MODE	20	enter cruise mode and then cruise preset point setting is permitted
PTZ_CMD_LEAVE_CURISE_MODE	22	quit cruise

		setting
PTZ_CMD_CRUISE_RUN	23	choose a cruise line to cruise
PTZ_CMD_CRUISE_STOP	24	PTZ stops cruise
PTZ_CMD_CRUISE_DEL	25	delete cruise line

byCruiseRoute
[in] cruise path,at most support 32 pathes

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_PTZCruise_Other

PTZ cruise operation

```
BOOL NET_SDK_PTZCruise_Other(
LONG     lUserID,
LONG     lChannel,
DWORD    dwPTZCruiseCmd,
BYTE     byCruiseRoute,
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

lChannel
[in] channel number,start from 0

dwPTZCruiseCmd
[in] operate PTZ cruise command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_CRUISE_CFG	19	set cruise line,amount to execute Enter,Set and Leave commands
PTZ_CMD_ENTER_CURISE_MODE	20	enter cruise mode and then cruise preset point

		setting is permitted
PTZ_CMD_LEAVE_CURISE_MODE	22	quit cruise setting
PTZ_CMD_CRUISE_RUN	23	choose a cruise line to cruise
PTZ_CMD_CRUISE_STOP	24	PTZ stops cruise
PTZ_CMD_CRUISE_DEL	25	delete cruise line

byCruiseRoute
[in] cruise path,at most support 32 pathes

Return Values

TRUE means success; FALSE means failure. to get error information,please call NET_SDK_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_PTZTrack

PTZ track operation,need to open preview

```
BOOL NET_SDK_PTZTrack(
LONG      lLiveHandle,
DWORD     dwPTZTrackCmd
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
dwPTZTrackCmd
[in] operate PTZ track command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_TRACK_START	26	start track
PTZ_CMD_TRACK_STOP	27	stop track
PTZ_CMD_TRACK_START_RECORD	28	start to store track
PTZ_CMD_TRACK_STOP_RECORD	29	stop storing track

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_DVR_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_LivePlay

Client SDK Instructions

NET_SDK_PTZTrack_Other

PTZ track operation

```
BOOL NET_SDK_PTZTrack_Other(
LONG     lUserID,
LONG     lChannel,
DWORD    dwPTZTrackCmd
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] channel number,starting with 0
dwPTZTrackCmd
[in] operate PTZ track command,list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_TRACK_START	26	start track
PTZ_CMD_TRACK_STOP	27	stop track
PTZ_CMD_TRACK_SET	28	start to store track
PTZ_CMD_TRACK_DEL	29	stop storing track

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_DVR_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_PTZAutoScan

PTZ automatic scan operation

```
BOOL NET_SDK_PTZAutoScan(
LONG        lLiveHandle,
DWORD       dwPTZAutoScanCmd,
DWORD       dwSpeed,
BOOL        bIsAutoScan
);
```

Parameters

lLiveHandle
[in] play handle
dwPTZAutoScanCmd
[in] PTZ automatic scan command, list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_AUTO_SCAN_START	29	start automatic scan
PTZ_CMD_AUTO_SCAN_STOP	30	stop automatic scan

dwSpeed
[in] speed of PTZ ,list as follows:

macro definition	value of macro definition

PTZ_SPEED_1	1
PTZ_SPEED_2	2
PTZ_SPEED_3	3
PTZ_SPEED_4	4
PTZ_SPEED_5	5
PTZ_SPEED_6	6
PTZ_SPEED_7	7
PTZ_SPEED_8	8

bIsAutoScan

[in] Scan mode, TURE means automatic scan, and FALSE represents random scan

Return Values

TURE means success,FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZAutoScan_Other

Client SDK Instructions

NET_SDK_PTZAutoScan_Other

PTZ automatic scan operation

```
BOOL NET_SDK_PTZAutoScan_Other(
LONG        lUserID,
LONG        lChannel,
DWORD       dwPTZAutoScanCmd
);
```

Parameters

lUserID
[in] user ID
lChannel
[in] channel number,starts from 0
dwPTZAutoScanCmd
[in] PTZ automatic scan command, list as follows:

macro definition	value of macro definition	meaning
PTZ_CMD_AUTO_SCAN_START	30	start automatic scan
PTZ_CMD_AUTO_SCAN_STOP	31	stop automatic scan

dwSpeed
[in] PTZ speed

Macro Definition	Meaning

PTZ_SPEED_1	1
PTZ_SPEED_2	2
PTZ_SPEED_3	3
PTZ_SPEED_4	4
PTZ_SPEED_5	5
PTZ_SPEED_6	6
PTZ_SPEED_7	7
PTZ_SPEED_8	8

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZAutoScan

Android Interface

The corresponding Android interface

```
boolean         PTZAutoScanOther(
long                    lUserID,
long                    lChannel,
int                     dwPTZAutoScanCmd
);
```

Client SDK Instructions

NET_SDK_PTZControl_3D

PTZ 3D control operation(need to start previewing image)

```
BOOL NET_SDK_PTZControl_3D(
LONG       lLiveHandle,
LONG       lChannel,
PTZ_3D_POINT_INFO      *pPtz3DInfo
);
```

Parameters

lLiveHandle
[in] return value of NET_SDK_LivePlay
lChannel
[in] channel number,starting with 0
**pPtz3DInfo*
[in] information about PTZ 3D control,as follows
PTZ_3D_POINT_INFO

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZControl

Client SDK Instructions

NET_SDK_PTZControl_3D_Ex

PTZ 3D control operation(no need to start previewing image)

```
BOOL NET_SDK_PTZControl_3D_Ex(
LONG     lUserID,
LONG       lChannel,
PTZ_3D_POINT_INFO      *pPtz3DInfo
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
lChannel
[in] channel number,starting with 0
**pPtz3DInfo*
[in] information about PTZ 3D control,as follows
PTZ_3D_POINT_INFO

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_PTZControl

Client SDK Instructions

NET_SDK_GetPTZCameraType

get camera type(only support IPC)

```
BOOL NET_SDK_PTZPreset_Other(
LONG      lUserID,
LONG      NET_SDK_CAMERA_TYPE *pCameraType
);
```

Parameters

lUserID
[in] return value of userID
*NET_SDK_CAMERA_TYPE *pCameraType*
[in] reture value of camera type,list as follows:

camera type	value of camera type	meaning
NET_SDK_NOT_SUPPORT_PTZ	0	bullet camera don't support ptz
NET_SDK_DOME_SUPPORT_PTZ	1	bullet camera support ptz
NET_SDK_SUPPORT_PTZ	2	dome camera support ptz
NET_SDK_PTZ_END		

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

Each operation on PTZ is corresponding to each control code between device and PTZ,device sends control code to PTZ according to current decoder type and decoder address. If current decoder doesnot match PTZ ,reconfig decoder is necessary.

See Also

NET_SDK_Login

Client SDK Instructions

```
typedef enum __Channel_type__
{
    E_NULL_CHL_TYPE,
    E_DIGITAL_CHL_TYPE,      //Digital Channel
    E_ANALOG_CHL_TYPE,       //Analog Channel
    E_ALARMOUT_CHL_TYPE,           //Alarm Output Channel
    E_SENSOR_CHL_TYPE,     //Sensor Channel
}CHANNEL_TYPE;
typedef struct _net_sdk_channel_ptz
{
    unsigned int         dwChannel;//Channel No.(starting with zero)
    CHANNEL_TYPE       eChanneltype;//Channel Type
    unsigned char        resv[8];
}NET_SDK_CHANNEL_PTZ;
```

NET_SDK_GetSupportPtzList

Get the channel of NVR in favor of PTZ list（only for N9000 device）

```
BOOL NET_SDK_GetSupportPtzList(
LONG     lUserID,
int     listNum,
NET_SDK_CHANNEL_PTZ     *pOutChannelPtz,
int     *returnListNum,
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
listNum
[in] the number of NET_SDK_CHANNEL_PTZ supported by the request memory of pOutChannelPtz
**pOutChannelPtz*
[out] Return PTZ information list
**returnListNum*
[out] the effective number of returning PTZ information list

Return Values

TRUE means success; FALSE means failure. To get error information, call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_GetPTZConfig

Get the related configuration of PTZ (only for N9000).

```
BOOL NET_SDK_GetPTZConfig(
LONG     lUserID,
LONG     lChannel,
DWORD    dwCommand,
LPVOID   lpInBuffer,
DWORD    dwInBufferSize,
LPVOID   lpOutBuffer,
DWORD    dwOutBufferSize,
LPDWORD  lpBytesReturned
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
lChannel
[in] channel number starts from 0. -1 means all channels.If the command don't need channel number, this parameter is invalid.
dwCommand
[in] device configuration command, see Configuration Commands
lpInBuffer
[in] buffer pointer to sending data. it is can not be NULL .
dwInBufferSize
[in] the buffer length of sending data (in byte). It is can not be 0.
lpOutBuffer
[out] the buffer pointer to receiving data.
dwOutBufferSize
[in] the buffer length of receiving data (in byte). It is can not be 0
lpBytesReturned
[out] the length pointer to the data actually received. it is can not be null

Remarks

Different functions has different structures and commands as shown below.

dwCommand **Macro Definition**	**dwCommand** **Meanings**	**Sending Structure**	**Receiving Structu**
DD_PTZ_CONFIG_PRESET	Get preset	NULL	DD_PTZ_PRESET_CON
DD_PTZ_CONFIG_CRUISE	Get cruise	NULL	DD_CH_CRUISE
DD_PTZ_CONFIG_CRUISE_POINT	Get presets f the curise	unsigned int(cruiseIndex)	DD_CRUISE_POINT_IN

See Also

NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_StartVoiceCom

start talkback

```
LONG NET_SDK_StartVoiceCom(
LONG                        lUserID,
BOOL                        bNeedCBNoEncData,
TALK_DATA_CALLBACK          fVoiceDataCallBack,
void                        * pUser
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

bNeedCBNoEncData
[in] whether need to encode voice when local voice data callback. If FALSE is selected, the stream from the device will be decoded first and then it is called back.

fVoiceDataCallBack
[in] function of audio data callback

PUser
[in] user data

Return Values

-1 means failure and other value is handle parameter of NET_SDK_StopVoiceCom. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_StopVoiceCom   NET_SDK_Login

Client SDK Instructions

NET_SDK_SetVoiceComClientVolume

set client volume of talkback

```
BOOL NET_SDK_SetVoiceComClientVolume(
LONG     lVoiceComHandle,
WORD     wVolume
);
```

Parameters

lVoiceComHandle
  [in] return value of NET_SDK_StartVoiceCom
wVolume
  [in] set volume,range[0,0xffff]

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_StartVoiceCom

Client SDK Instructions

TALK_DATA_CALLBACK

data callback in talkback

```
void *TALK_DATA_CALLBACK(
LONG           lVoiceComHandle,
char           *pRecvDataBuffer,
DWORD          dwBufSize,
BYTE           byAudioFlag,
void           * pUser
);
```

Parameters

lVoiceComHandle
[in] interface handle of talkback

**pRecvDataBuffer*
[in] buffer pointer to receive talkback data,received talkback data is PCM data without encoding

dwBufSize
[in] size of buffer area

byAudioFlag
[in] audio mark,the only value is 1,and it means the talkback data from device

pUser
[in] pointer to user information

Return Values

None. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_GetAudioInfo

Get audio information.

```
BOOL NET_SDK_GetAudioInfo(
LONG      lVoiceComHandle,
void      *pAudioInfo,
LONG      infoLen
);
```

Parameters

lVoiceComHandle
: [in] the return value of NET_SDK_StartVoiceCom()

pAudioInfo
: [in] a pointer to audio information

infoLen
: [in] the length of audio information

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_ReleaseAudioEncoder NET_SDK_EncodeAudioFrame

>

Client SDK Instructions

NET_SDK_StopVoiceCom

stop talkback or voice forward

```
BOOL NET_SDK_StopVoiceCom(
LONG     lVoiceComHandle
);
```

Parameters

lVoiceComHandle
[in] return value of NET_SDK_StartVoiceCom or NET_SDK_StartVoiceCom_MR

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_StartVoiceCom   NET_SDK_StartVoiceCom_MR

Client SDK Instructions

NET_SDK_StartVoiceCom_MR

start voice forward function

```
LONG NET_SDK_StartVoiceCom_MR(
LONG                     lUserID,
TALK_DATA_CALLBACK       fVoiceDataCallBack,
void                     * pUser
);
```

Parameters

lUserID
    [in] user ID
bNeedNoEncodeData
    [in] whether to encode the language data. If TRUE is selected, the stream from the device will be directly called back and the stream transfered to the device is encoded by SDK . If FALSE is selected, the stream from the deivce is decoded first and then called back; the stream transfered to the device is not encoded by SDK.
fVoiceDataCallBack
    [in] return value of voice data callback function
pUser
    [in] user information

Return Values

-1 means failure and other value is return value of voice forward. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_VoiceComSendData

Client SDK Instructions

NET_SDK_VoiceComSendData

send data when voice forward

```
BOOL NET_SDK_VoiceComSendData(
LONG          lVoiceComHandle,
char          *pSendBuf,
DWORD         dwBufSize
);
```

Parameters

lVoiceComHandle
: [in] handle of voice component

pSendBuf
: [in] buffer pointer to send data, the talkback data in buffer area is PCM data without encoding

dwBufSize
: [in] size of buffer area

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_StartVoiceCom_MR

Client SDK Instructions

NET_SDK_EncodeAudioFrame

encode audio

```
BOOL NET_SDK_EncodeAudioFrame(
LONG            lEncodeHandle,
unsigned char   *pInBuffer,
LONG            inLen,
unsigned char   * pOutBuffer ,
LONG            *pOutLen
);
```

Parameters

IEncodeHandle
[in] handle of encoding audio, the return value of NET_SDK_InitAudioEncoder()

**pInBuffer*
[in] buffer area for input,get PCM audio data according to sample standard(sample frequency is 16000,16 bytes, signal channel),the standard size of input data is 1280 bytes.

inLen
[out] buffer area for output,data length after encoding

pOutBuffer
[out] buffer area for output,the output data size after encoding is 80 bytes.

**pOutLen*
[out] buffer area for output,output data length after encoding

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

It's set to mate with talkback & forward function.When client sends original audio data to device, firstly compress and encode audio through function of audio encoding,and then send to device. Client gets the compressed code stream,and then call NET_SDK_DecodeAudioFrame to decode data. Before calling encode and decode function initialization is needed, also when call function is finished,free resource.

See Also

NET_SDK_InitAudioEncoder

Client SDK Instructions

NET_SDK_InitAudioEncoder

initialize audio encoder

```
void* NET_SDK_InitAudioEncoder(
void      *pAudioInfo,
LONG      infoLen
);
```

Parameters

pAudioInfo
[in] pointer to audio information
infoLen
[in] length of audio information

Return Values

-1 means failure and other value is handle of audio encoding. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_ReleaseAudioEncoder   NET_SDK_EncodeAudioFrame

Client SDK Instructions

NET_SDK_ReleaseAudioEncoder

free audio encoding resource

```
void NET_SDK_ReleaseAudioEncoder(
LONG   *IEncodeHandle
);
```

Parameters

IEncodeHandle
[in] handle of audio encoding,return value of NET_SDK_InitAudioEncoder

Return Values

None. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_InitAudioEncoder

Client SDK Instructions

NET_SDK_DecodeAudioFrame

audio decode

```
BOOL NET_SDK_DecodeAudioFrame(
LONG                 lDecodeHandle,
unsigned char        *pInBuffer,
LONG                 inLen,
unsigned char        * pOutBuffer,
LONG                 *pOutLen
);
```

Parameters

IDecodeHandle
: [in] audio encode handle,return value of NET_SDK_InitAudioDecoder()

**pInBuffer*
: [in] input buffer area,get PCM audio data according to sample standard(sampling frequency is in000,16 bits, signal channel),standard data size of input is 1280 bytes

inLen
: [out] output buffer are,data length after encoding

pOutBuffer
: [out] output buffer area,output data size after encoding is 80 bytes

**pOutLen*
: [out] output buffer area,output data length after encoding

Return Values

TRUE means success and FALSE means failure. To get error information,please call NET_SDK_GetLastError

Remarks

It's set to mate with talkback & forward function.When client send original audio data to device, firstly compress and encode audio through function of audio encoding,and then send to device. Client gets the compressed code stream,and then call this interface to decode data. Before calling encode and decode function initialization is needed, also when call function is finished,free resource.

See Also

NET_SDK_InitAudioDecoder

Client SDK Instructions

NET_SDK_InitAudioDecoder

initialize audio decoder

```
LONG NET_SDK_InitAudioDecoder(
void      *pAudioInfo,
LONG      infoLen
);
```

Parameters

pAudioInfo
[in] pointer to audio information

infoLen
[in] length of audio information

Return Values

-1 means failure and other value is handle of audio decoding. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_ReleaseAudioDecoder NET_SDK_DecodeAudioFrame

Client SDK Instructions

NET_SDK_ReleaseAudioDecoder

free audio decoding resource

```
void NET_SDK_ReleaseAudioDecoder(
LONG   IDecodeHandle
);
```

Parameters

IDecodeHandle
  [in] handle of audio decoding,return value of NET_SDK_InitAudioDecoder

Return Values

None. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_InitAudioDecoder

Client SDK Instructions

NET_SDK_FormatDisk

format harddisk remotely(only support 3.0DVR)

```
LONG NET_SDK_FormatDisk(
LONG    lUserID,
LONG    lDiskNumber
);
```

Parameters

lUserID
    [in] return value of NET_SDK_Login()

lDiskNumber
    [in] harddisk number,start from 0,0xff is valid to all harddisks(except read-only harddisk)

Return Values

-1 means failure and other value is parameter of function NET_SDK_CloseFormatHandle. To get error information,please call NET_SDK_GetLastError

Remarks

If network breaks in formation process,formation on device continues but client can't receive the state.

See Also

NET_SDK_CloseFormatHandle   NET_SDK_GetFormatProgress   NET_SDK_Login

Client SDK Instructions

NET_SDK_GetFormatProgress

get progress of formatting harddisk(only support 3.0DVR)

```
BOOL NET_SDK_GetFormatProgress(
LONG    lFormatHandle,
LONG    *pCurrentFormatDisk,
LONG    *pCurrentDiskPos,
LONG    *pFormatStatic
);
```

Parameters

lFormatHandle
: [in] handle of save formatting harddisk,return value of NET_SDK_FormatDisk

pCurrentFormatDisk
: [out] pointer to save current formatting harddisk number,harddisk number starts from 0,-1 is the initial value

pCurrentDiskPos
: [out] pointer to save current progress of formatting harddisk,progress range is 0-100

pFormatStatic
: [out] pointer to save the state of formatting harddisk,0-formatting;1-finish formation; 2-formatting current harddisk makes mistake,formation can't continue,this error shows both in local and network harddisks; 3-formation of current harddisk can't start due to lost of network harddisk caused by network exception

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_FormatDisk

Client SDK Instructions

NET_SDK_CloseFormatHandle

close handle of formatting harddisk,and free resource(only support 3.0DVR)

```
BOOL NET_SDK_CloseFormatHandle(
LONG    lFormatHandle
);
```

Parameters

lFormatHandle
[in] return value of NET_SDK_ FormatDisk()

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_FormatDisk

Client SDK Instructions

NET_SDK_FindDisk

Get the HDD information of the device.

```
POINTERHANDLE NET_SDK_FindDisk(
LONG    lUserID,
);
```

Parameters

lUserID
    [in] the return value of NET_SDK_Login()

Return Values

Return to the handle of getting HDD information. The value is greater than 0, which means success.   NET_SDK_GetLastError

See Also

NET_SDK_GetNextDiskInfo   NET_SDK_FindDiskClose
  NET_SDK_Login

Client SDK Instructions

NET_SDK_GetNextDiskInfo

Get the information of the HDD in order (call up serveral times until return failure).

```
BOOL NET_SDK_GetNextDiskInfo(

 POINTERHANDLE lDiskHandle ,
NET_SDK_DISK_INFO *pDiskInfo,
 );
```

Parameters

lDiskHandle
  [in] the return value of NET_SDK_FindDisk
pDiskInfo
  [out] return to the HDD information

Return Values

The return value is greater than 0 which means success. NET_SDK_GetLastError

See Also

NET_SDK_FindDisk NET_SDK_FindDiskClose
NET_SDK_Login

Client SDK Instructions

NET_SDK_FindDiskClose

Having finished getting the HDD information of the device, release resources.

```
BOOL NET_SDK_FindDiskClose(
POINTERHANDLE lDiskHandle,
);
```

Parameters

lDiskHandle
[in] the return value of NET_SDK_FindDisk

Return Values

The value is greater than 0, which means success. NET_SDK_GetLastError

See Also

NET_SDK_FindDisk   NET_SDK_GetNextDiskInfo

Client SDK Instructions

NET_SDK_ActiveDevice

active the IPC device

```
BOOL NET_SDK_ActiveDevice(
 char   *pIp,
 int     iPort,
 char   *password
);
```

Parameters

pIp
: [in] the IP address of the device to be activated(can be the second ip starting with 169.254)

iPort
: [in] the http port of the device is to be activated

password
: [in] the password to activate the device

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_GetUpgradeState NET_SDK_GetUpgradeProgres

Client SDK Instructions

NET_SDK_Upgrade

remote upgrade

```
LONG NET_SDK_Upgrade(
LONG    lUserID,
char   *sFileName
);
```

Parameters

lUserID
  [in] return value of NET_SDK_Login
wVolume
  [in] directory of file upgrade(include file name)

Return Values

-1 means failure and other values are parameter of NET_SDK_GetUpgradeState. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login   NET_SDK_CloseUpgradeHandle
NET_SDK_GetUpgradeState   NET_SDK_GetUpgradeProgres

Client SDK Instructions

NET_SDK_GetUpgradeState

get state of remote upgrading

```
int NET_SDK_GetUpgradeState(
LONG   lUpgradeHandle
);
```

Parameters

lUpgradeHandle
[in] return value of NET_SDK_Upgrade

Return Values

-1 means failure. Other values' meanings are as follows:
1-finish upgrading
2-upgrading
3-failed
4-network breaks,unknown state
5-language version of upgrading file doesnot match
Get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Upgrade

Client SDK Instructions

NET_SDK_GetUpgradeProgress

get progress of remote upgrading

```
int NET_SDK_GetUpgradeProgress(
LONG   lUpgradeHandle
);
```

Parameters

lUpgradeHandle
[in] return value of NET_SDK_Upgrade

Return Values

-1 means failure and 0-100 means progress of upgrading.
To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Upgrade

Client SDK Instructions

NET_SDK_CloseUpgradeHandle

close handle of remote upgrade,and free resource

```
BOOL NET_SDK_CloseUpgradeHandle(
LONG   lUpgradeHandle
);
```

Parameters

lUpgradeHandle
  [in] return value of NET_SDK_Upgrade()

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Upgrade

Client SDK Instructions

NET_SDK_UpgradeIPC

remote upgrade IPC

```
LONG NET_SDK_UpgradeIPC(
LONG    lUserID,
char   *sFileName,
unsigned int fileType
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login
sFileName
[in] directory of file upgrade(include file name)
fileType
[in] type of the upgrade file; 0:software;1:kernel;2:Uboot;3:AIlib

Return Values

-1 means failure and other values are parameter of NET_SDK_GetUpgradeState. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login NET_SDK_CloseUpgradeHandle
NET_SDK_GetUpgradeState NET_SDK_GetUpgradeProgres

Client SDK Instructions

NET_SDK_FindDVRLog

find log information of device(not support IPC)

```
LONG NET_SDK_FindDVRLog(
LONG                  lUserID,
DWORD                 dwType,
DD_TIME               *lpStartTime,
DD_TIME               *lpStopTime
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login()

dwType
[in] type of log(examples see N9000_LOG_MAJOR_TYPE)

**lpStartTime*
[in] the start time of file

**lpStopTime*
[in] the stop time of file

Return Values

-1 means failure and other values are part of parameters in function NET_DVR_FindNextLog. To get error information, please call NET_SDK_GetLastError

Remarks

This interface is used to search normal log information,and the maximum capacity is 2000.

See Also

NET_SDK_Login NET_SDK_FindNextLog NET_SDK_FindLogClose

Client SDK Instructions

NET_SDK_FindNextLog

get log information one by one(not support IPC)

```
LONG NET_SDK_FindNextLog(
LONG                    lLogHandle,
LPNET_SDK_LOG           lpLogData
);
```

Parameters

lLogHandle
[in] handle for finding log.return value of NET_SDK_FindDVRLog()

lpLogData
[out] pointer to store log information

Return Values

-1 means failure and other values are current state information. To get error information, please call NET_SDK_GetLastError

Remarks

Before calling this interface,call NET_SDK_FindDVRLog to get current handle for finding.

See Also

NET_SDK_FindDVRLog

Client SDK Instructions

NET_SDK_FindLogClose

free resource of finding log(not support IPC)

```
BOOL NET_SDK_FindLogClose(
LONG   lLogHandle
);
```

Parameters

lLogHandle
[in] handle of finding log,return value of NET_SDK_FindDVRLog()

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_FindDVRLog

Client SDK Instructions

NET_SDK_FindEventInfo

find event info(only support 3.0DVR)

```
LONG NET_SDK_FindEvent(
LONG                  lUserID,
DWORD                 dwType
ULONGLONG             channlMask,
DD_TIME               *lpStartTime,
DD_TIME               *lpStopTime
);
```

Parameters

lUserID
[in] returned value of NET_SDK_Login()
dwType
[in] event type,refer to DD_EVENT_TYPE:

Type	Value
DD_EVENT_TYPE_MOTION	0x0001
DD_EVENT_TYPE_SENSOR	0x0002
DD_EVENT_TYPE_V_LOSS	0x0004
DD_EVENT_TYPE_V_COVER	0x0008

channlMask
[in] event happend in which channel,
((ULONGLONG)0x1 << N) N is search channel.
**lpStartTime*
[in] event starting time
**lpStopTime*
[in] event ending time

Return Values

-1 means failure and other values are parameter of NET_SDK_FindNextEventInfo. To get error code, please call NET_SDK_GetLastError

See Also

NET_SDK_FindNextEventInfo   NET_SDK_FindEventInfoClos
e

Client SDK Instructions

NET_SDK_FindNextEventInfo

find event info one by one(only support 3.0DVR)

```
LONG NET_SDK_FindNextEventInfo(
LONG                        lEventHandle,
LPNET_SDK_EVENT           lpEventData
);
```

Parameters

lEventHandle
: [in] handle of searching event info, the return value of NET_SDK_FindEventInfo()

lpEventData
: [out] pointer of saving event info

Return Values

-1 means failure and other values are event info. To get error code, please call NET_SDK_GetLastError

Remarks

Before call this interface to search event info, please call NET_SDK_FindEventInfo() to get the search handle.

See Also

NET_SDK_FindEventInfo   NET_SDK_FindEventInfoClose

Client SDK Instructions

NET_SDK_FindEventInfoClose

Close searching event info,free resource(only support 3.0DVR)

```
BOOL NET_SDK_FindEventInfoClose(
LONG       lEventHandle
);
```

Parameters

lEventHandle
    [in] search handle

Return Values

TURE means success; FALSE means failure. To get error code,please call NET_SDK_GetLastError

See Also

NET_SDK_FindEventInfo   NET_SDK_FindNextEventInfo

Client SDK Instructions

NET_SDK_GetDefaultVideoEffect

get default video effect

```
BOOL NET_SDK_GetDefaultVideoEffect(
LONG           lUserID,
DWORD          *pBrightValue,
DWORD          *pContrastValue,
DWORD          *pSaturationValue,
DWORD          *pHueValue
);
```

Parameters

lUserID
[in] user ID
**pBrightValue*
[in] pointer to brightness value
**pContrastValue*
[in] pointer to color contrast
**pSaturationValue*
[in] pointer to color saturation
**pHueValue*
[in] pointer to gray scale

Return Values

TURE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_GetVideoEffect NET_SDK_SaveVideoEffect NET_SDK_SetVideoEffect

Client SDK Instructions

NET_SDK_SetConfigFile

import configuration file

```
BOOL NET_SDK_SetConfigFile(
LONG    lUserID,
char    *sFileName
);
```

Parameters

lUserID
: [in] user ID,return value of NET_SDK_Login

sFileName
: [in] directory of saving configuration file(binary file)

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_DVR_Login

Client SDK Instructions

NET_SDK_GetConfigFile

export configuration file

```
BOOL NET_SDK_GetConfigFile(
LONG    lUserID,
char    *sFileName
);
```

Parameters

lUserID
[in] user ID,return value of NET_SDK_Login()
sFileName
[in] directory of storing configuration file(binary file)

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_GetNvrRecordDays

Query the number of days of video exists on the NVR device（works for NVR only）

```
BOOL NET_SDK_GetNvrRecordDays(
  LONG    lUserID,
  NET_SDK_NVR_DISKREC_DATE_ITEM* pDiskRecDateInfo
  LONG lBuffSize,
  LONG* pDISKCount
 );
```

Parameters

lUserID
  [in] returned value of NET_SDK_Login()
pDiskRecDateInfo
  [out] video days information structure pointer for the NVR device
lBuffSize
  [in] size of NET_SDK_NVR_DISKREC_DATE_ITEM structure
pDISKCount
  [out] number of hard drives for the NVR

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

Each hard disk of the NVR device corresponds to a query result, with the video days format 2021-

11-27~2021-12-
07, indicating the existence of video on the hard di
sk during this time period

Client SDK Instructions

NET_SDK_ShutDownDVR

close device

```
BOOL NET_SDK_ShutDownDVR(
LONG        lUserID
);
```

Parameters

lUserID
[in] user ID,return value of NET_SDK_Login

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_RebootDVR

reboot device

```
BOOL NET_SDK_RebootDVR(
LONG   lUserID
);
```

Parameters

lUserID
[in] user ID,return value of NET_SDK_Login

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_ChangTime

modify system time of device

```
BOOL NET_SDK_ChangTime(
LONG            lUserID,
unsigned long    time
);
```

Parameters

lUserID
[in] user ID
time
[in] device system time

Return Values

TURE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

---

Client SDK Instructions

NET_SDK_FormatTime

Transform inter time to format time

```
void NET_SDK_FormatTime(
LONGLONG                    intTime,
DD_TIME                 *pFormatTime
);
```

Parameters

intTime
: [in] inter time form NET_SDK_FRAME_INFO, from 1970-01-01,00:00:00,unit is microsecond

**pFormatTime*
: [in] DD_TIME format time

Return Values

None.This interface transforms the *time* parameter of NET_SDK_FRAME_INFO *to DD_TIMEformat time. To get error code, please call NET_SDK_GetLastError*

Client SDK Instructions

NET_SDK_SaveVideoEffect

save setting of video effect

```
BOOL NET_SDK_SaveVideoEffect(
LONG          lUserID,
LONG          lChannel,
DWORD         dwBrightValue,
DWORD         dwContrastValue,
DWORD         dwSaturationValue,
DWORD         dwHueValue
);
```

Parameters

lUserID
[in] user ID

lChannel
[in] channel number,start from 0

dwBrightValue
[in] value of brightness

dwContrastValue
[in] value of constract

dwSaturationValue
[in] value of saturation

dwHueValue
[in] value of gray scale

Return Values

TURE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

See Also

NET_SDK_GetVideoEffect NET_SDK_GetDefaultVideoEffect NET_SDK_SetVideoEffect

Client SDK Instructions

NET_SDK_ModifyDeviceNetInfo

Modify the network configuration of the device accoding to the matching MAC.

```
BOOL NET_SDK_ModifyDeviceNetInfo(
 NET_SDK_DEVICE_IP_INFO *pDeviceIPInfo
);
```

Parameters

pDeviceIPInfo
[in] the network configuration of the device

Return Values

-1 means failure; other values means the returned information value. To get error information, please call NET_SDK_GetLastError

See Also

Client SDK Instructions

NET_SDK_TransparentConfig

Transparent API protocol Interface

```
BOOL NET_SDK_TransparentConfig(
LONG      lUserID,
char     *sendXML,
char     *strUrl,
LPVOID    lpOutBuffer,
DWORD    dwOutBufferSize,
LPDWORD    lpBytesReturned
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login

sendXML
[in] xml contents in API

strUrl
[in] URL of API。（IP and port are not included. eg：the origninal URL of API protocol ishttp://[:port]/PtzStopCruise[/channelId]. Here the URL is PtzStopCruise/channelId）.

lpOutBuffer
[out] the buffer pointer of receiving data

dwOutBufferSize
[in]the buffer size of receiving data in bytes can not be zero.

lpBytesReturned
[out]a pointer to the data length actually received can not be null.

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

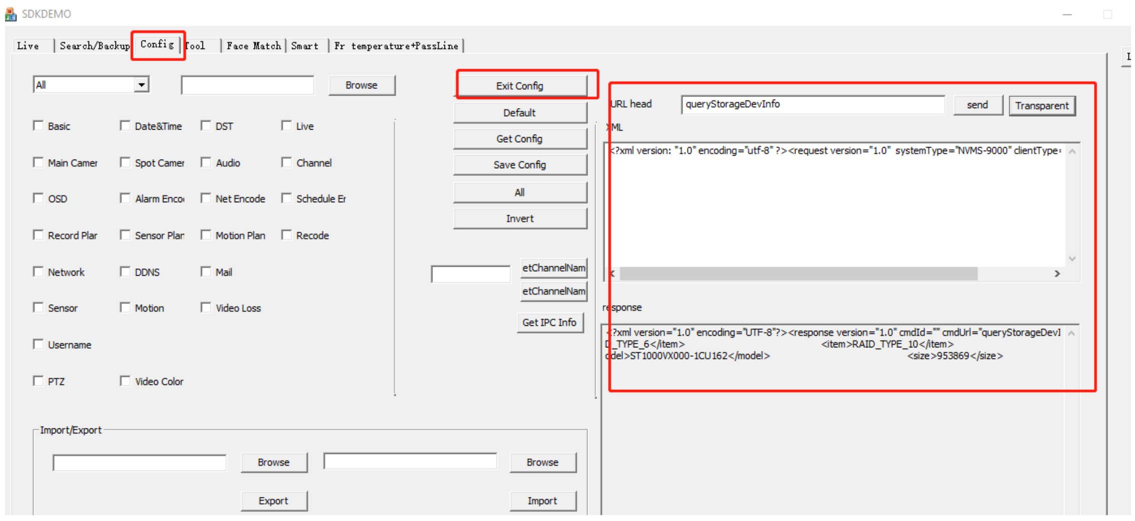
This function can be used to send http command to the device eg.

	strUrl	sendXML	lpOutBuffer

Query NVR's online channel list	queryOnlineChlList	`<?xml version="1.0" encoding="utf-8" ?>`
		`<request version="1.0" refresh = "true" systemType="NVMS-9000" clientType="WEB">`
		`</request>`
			`<?xml version="1.0" encoding="UTF-8"?>`
			`<response cmdId="" cmdUrl="queryOnlineChlList">`
			`<status>success</status>`
			`<content type="list">`
			`<item id="{00000007-0000-0000-0000-000000000000}"></item>`
			`<item id="{00000008-0000-0000-0000-000000000000}"></item>`
			`</content>`
			`</response>`

Set IPC's exposure mode to manual	SetImageConfig/1	`<config><image><autoExposureMode><mode>manual</mode><value>33333</value></autoExposureMode></image><cfgFile>normal</cfgFile></config>`	`<?xml version="1.0" encoding="UTF-8"?>`
			`<config version="1.0" xmlns="http://www.ipc.com/ver10" status="success" errorCode="200" IssameOldPwd="false"/>`

Test the function with the sdk demo: click the "Config" tab, input the strUrl and sendXML,click the "Transparent" button, the lpOutBuffer will display under



URL head queryStorageDevInfo

XML

```
<?xml version: "1.0" encoding="utf-8" ?><request version="1.0"  systemType="NVMS-9000" clientType=
```

response

```
<?xml version="1.0" encoding="UTF-8"?><response version="1.0" cmdId="" cmdUrl="queryStorageDevI
D_TYPE_6</item>                        <item>RAID_TYPE_10</item>
odel>ST1000VX000-1CU162</model>                        <size>953869</size>
```

Client SDK Instructions

NET_SDK_GetDeviceInfo

Get parameters of decoding device

```
BOOL NET_SDK_GetDeviceInfo(
LONG                lUserID,
LPNET_SDK_DEVICEINFO   pdecviceInfo
);
```

Parameters

lUserID
    [in] return value of NET_SDK_Login()
pdecviceInfo
    [out] information about device parameter

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_GetDeviceTypeName

get device type name

```
LONG NET_SDK_GetDeviceTypeName(
LONG            lUserID,
char    *pNameBuffer,
long     bufferLen
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
**pNameBuffer*
[out] type name buffer of device
bufferLen
[out] buffer length of device type name

Return Values

The returned value is the device type name. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_GetDeviceTime

get system time of device

```
 BOOL NET_SDK_GetDeviceTime(
 LONG           lUserID,
DD_TIME      *pTime
 );
```

Parameters

lUserID
[in] user ID
**pTime*
[in] pointer of device system time

Return Values

TURE means success,FALSE means failure.If live frame hasn't arrived client in 50 milliseconds, return value is False.Please try to call this interface more times until return value is True. To get error information,please call NET_SDK_GetLastError

NET_SDK_GetPTZCameraType

Get camera type

```
BOOL NET_SDK_GetPTZCameraType(
LONG     lUserID,
NET_SDK_CAMERA_TYPE *pCameraType ,
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
pCameraType
[in] Camera type as shown below:

Return Values

TRUE means sucess; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

NET_SDK_CAMERA_TYPE is an example. Please refer to the corresponding type of the libary function.

See Also

NET SDK GetDVRConfig

Camera type	Camera type value	Meanings
NET_SDK_NOT_SUPPORT_PTZ	0	The camera don't support PTZ
NET_SDK_DOME_SUPPORT_PTZ	1	The camera supports PTZ
NET_SDK_SUPPORT_PTZ	2	The camera supports PTZ
NET_SDK_PTZ_END		

NET_SDK_GetAlarmStatus

Get the alarm information of the device.

```
BOOL NET_SDK_GetAlarmStatus(
LONG lUserID,
LPVOID lpOutBuffer,
DWORD dwOutBufferSize,
LPDWORD lpBytesReturned);
```

Parameters

lUserID
[in]the return value of NET_SDK_Login()
lpOutBuffer
[out]buf of the alarm event output
dwOutBufferSize
[in]the space of lpOutBuffer applied for
lpBytesReturned
[out]the size of the valid data of the returned lpOutBuffer

```
typedef struct _alarm_status

{
   unsigned int iSize;                                   //The length of the structure
   int chanl;     //Alarm channel.The alarm unrelated to the channel is -1.
   unsigned int alarmType;     //Alarm event NET_SDK_N9000_ALARM_TYPE
}DD_ALARM_STATUS_INFO;

enum NET_SDK_N9000_ALARM_TYPE
{
    NET_SDK_N9000_ALARM_TYPE_RANGE_BEGIN,
    NET_SDK_N9000_ALARM_TYPE_MOTION=0x01,/////Motion detection alarm input
    NET_SDK_N9000_ALARM_TYPE_SENSOR,/////Sensor alarm input
    NET_SDK_N9000_ALARM_TYPE_VLOSS,////Video loss alarm input
    NET_SDK_N9000_ALARM_TYPE_FRONT_OFFLINE, //////Camera offline alarm
    NET_SDK_N9000_ALARM_TYPE_OSC,            ////Object removal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD,            ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_SCENE,      ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_CLARITY,    ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_COLOR,      ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_PEA_TRIPWIRE,   ////Line crossing detection alarm
    NET_SDK_N9000_ALARM_TYPE_PEA_PERIMETER,  ////Region Intrusion detection alarm
    NET_SDK_N9000_ALARM_TYPE_VFD,            ////Face detection(only for ipc)
    NET_SDK_N9000_ALARM_TYPE_CDD,            ////Crowdy density
    NET_SDK_N9000_ALARM_TYPE_IPD,            ////people intrusion
    NET_SDK_N9000_ALARM_TYPE_CPC,            ////people counting
    NET_SDK_N9000_ALARM_TYPE_FACE_MATCH,            ////face comparation alarm(for nvr)
    NET_SDK_N9000_ALARM_TYPE_FACE_MATCH_FOR_IPC,    ////face comparation alarm(for ipc)
    NET_SDK_N9000_ALARM_TYPE_PEA_FOR_IPC,           ////Line crossing and region intrus
    NET_SDK_N9000_ALARM_TYPE_TRAJECT,               ////target tracking trajectory
    NET_SDK_N9000_ALARM_TYPE_VEHICE,                ////license plate for ipc
    NET_SDK_N9000_ALARM_TYPE_AOIENTRY,              ////enter region  for ipc
    NET_SDK_N9000_ALARM_TYPE_AOILEAVE,           ////leave region  for ipc
    NET_SDK_N9000_ALARM_TYPE_PASSLINE,           ////passline counting for ipc

    NET_SDK_N9000_ALARM_TYPE_GPS_SPEED_OVER=0x21,//overspeed alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_CROSS_BOADER,//line crossing
    NET_SDK_N9000_ALARM_TYPE_GPS_TEMPERATURE_OVER,//temperature alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_X,//GSENSOR alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_Y,
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_Z,

    NET_SDK_N9000_ALARM_TYPE_EXCEPTION = 0x41,
    NET_SDK_N9000_ALARM_TYPE_IP_CONFLICT,   /////IP address conflict
    NET_SDK_N9000_ALARM_TYPE_DISK_IO_ERROR, /////Disk IO error
    NET_SDK_N9000_ALARM_TYPE_DISK_FULL,    /////Disk full
    NET_SDK_N9000_ALARM_TYPE_RAID_SUBHEALTH, //Raid subhealth
    NET_SDK_N9000_ALARM_TYPE_RAID_UNAVAILABLE, //Raid unavailabe
    NET_SDK_N9000_ALARM_TYPE_ILLEIGAL_ACCESS,  /////Illegal access
    NET_SDK_N9000_ALARM_TYPE_NET_DISCONNECT,  /////Network disconnection
    NET_SDK_N9000_ALARM_TYPE_NO_DISK,            ////No disk
    NET_SDK_N9000_ALARM_TYPE_SIGNAL_SHELTER, //Signal obstruction
    NET_SDK_N9000_ALARM_TYPE_HDD_PULL_OUT, //HDD pulled out


    NET_SDK_N9000_ALARM_TYPE_ALARM_OUT = 0x51,  /////Alarm output tpye.

    NET_SDK_N9000_ALARM_TYPE_RANGE_END = 0xFF,////It is unable to exceed this value, o
};
```

Return Values

TRUE means success, FALSE means failed. To get error information, please call NET_SDK_GetLastError

NET_SDK_GetAlarmStatusEx

Get the alarm information(including the raid alarm) of the device.

```
BOOL NET_SDK_GetAlarmStatusEx(
LONG lUserID,
LPVOID lpOutBuffer,
DWORD dwOutBufferSize,
LPDWORD lpBytesReturned,
 int  *exStructNum );
```

Parameters

lUserID
  [in]the return value of NET_SDK_Login()
lpOutBuffer
  [out]buf of the alarm event output
dwOutBufferSize
  [in]the space of lpOutBuffer applied for
lpBytesReturned
  [out]the size of the valid data of the returned lpOutBuffer
exStructNum
  [out]the number of the raid alarms

```
typedef struct _alarm_status_ex

{
   unsigned int iSize;                              //The length of the structure
   int chanl;     //Alarm channel.The alarm unrelated to the channel is -1.
   unsigned int alarmType;     //Alarm event NET_SDK_N9000_ALARM_TYPE,   NET_SDK_N9000
   char alarmNode[32];                              //The length of the structure
   char recv[32];             //reserved.
}DD_ALARM_STATUS_INFO_Ex;

enum NET_SDK_N9000_ALARM_TYPE
{
    NET_SDK_N9000_ALARM_TYPE_RANGE_BEGIN,
    NET_SDK_N9000_ALARM_TYPE_MOTION=0x01,/////Motion detection alarm input
    NET_SDK_N9000_ALARM_TYPE_SENSOR,/////Sensor alarm input
    NET_SDK_N9000_ALARM_TYPE_VLOSS,////Video loss alarm input
    NET_SDK_N9000_ALARM_TYPE_FRONT_OFFLINE, //////Camera offline alarm
    NET_SDK_N9000_ALARM_TYPE_OSC,            ////Object removal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD,            ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_SCENE,      ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_CLARITY,    ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_AVD_COLOR,      ////Abnormal video signal detection alarm
    NET_SDK_N9000_ALARM_TYPE_PEA_TRIPWIRE,   ////Line crossing detection alarm
    NET_SDK_N9000_ALARM_TYPE_PEA_PERIMETER, ////Region Intrusion detection alarm
    NET_SDK_N9000_ALARM_TYPE_VFD,            ////Face detection(only for ipc)
    NET_SDK_N9000_ALARM_TYPE_CDD,            ////Crowdy density
    NET_SDK_N9000_ALARM_TYPE_IPD,            ////people intrusion
    NET_SDK_N9000_ALARM_TYPE_CPC,            ////people counting
    NET_SDK_N9000_ALARM_TYPE_FACE_MATCH,            ////face comparation alarm(for nvr)
    NET_SDK_N9000_ALARM_TYPE_FACE_MATCH_FOR_IPC,    ////face comparation alarm(for ipc)
    NET_SDK_N9000_ALARM_TYPE_PEA_FOR_IPC,           ////Line crossing and region intrus
    NET_SDK_N9000_ALARM_TYPE_TRAJECT,               ////target tracking trajectory
    NET_SDK_N9000_ALARM_TYPE_VEHICE,                ////license plate for ipc
    NET_SDK_N9000_ALARM_TYPE_AOIENTRY,              ////enter region  for ipc
    NET_SDK_N9000_ALARM_TYPE_AOILEAVE,           ////leave region  for ipc
    NET_SDK_N9000_ALARM_TYPE_PASSLINE,           ////passline counting for ipc

    NET_SDK_N9000_ALARM_TYPE_GPS_SPEED_OVER=0x21,//overspeed alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_CROSS_BOADER,//line crossing
    NET_SDK_N9000_ALARM_TYPE_GPS_TEMPERATURE_OVER,//temperature alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_X,//GSENSOR alarm
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_Y,
    NET_SDK_N9000_ALARM_TYPE_GPS_GSENSOR_Z,

    NET_SDK_N9000_ALARM_TYPE_EXCEPTION = 0x41,
    NET_SDK_N9000_ALARM_TYPE_IP_CONFLICT,   /////IP address conflict
    NET_SDK_N9000_ALARM_TYPE_DISK_IO_ERROR, /////Disk IO error
    NET_SDK_N9000_ALARM_TYPE_DISK_FULL,    /////Disk full
    NET_SDK_N9000_ALARM_TYPE_RAID_SUBHEALTH, //Raid subhealth
    NET_SDK_N9000_ALARM_TYPE_RAID_UNAVAILABLE, //Raid unavailabe
    NET_SDK_N9000_ALARM_TYPE_ILLEIGAL_ACCESS,  /////Illegal access
    NET_SDK_N9000_ALARM_TYPE_NET_DISCONNECT,  /////Network disconnection
    NET_SDK_N9000_ALARM_TYPE_NO_DISK,           ////No disk
    NET_SDK_N9000_ALARM_TYPE_SIGNAL_SHELTER, //Signal obstruction
    NET_SDK_N9000_ALARM_TYPE_HDD_PULL_OUT, //HDD pulled out


    NET_SDK_N9000_ALARM_TYPE_ALARM_OUT = 0x51,  /////Alarm output tpye.
    NET_SDK_N9000_ALARM_TYPE_RANGE_END = 0xFF,////It is unable to exceed this value, o
};
```

Return Values

TRUE means success, FALSE means failed. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_GetDeviceSupportFunction

get the functions of the IPC（only support IPC）。

```
BOOL NET_SDK_GetDeviceSupportFunction(
LONG               lUserID,
NET_SDK_DEV_SUPPORT*   pDevSupport;
);
```

Parameters

lUserID
  [in] the return value of NET_SDK_Login()
pDevSupport
  [out] the functions of the IPC

Return Values

TRUE means success, FALSE means failed。To get error information, please callNET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_GetDeviceIPCInfo

get device management information

```
LONG NET_SDK_GetDeviceIPCInfo(
LONG                lUserID,
NET_SDK_IPC_DEVICE_INFO    *pDeviceIPCInfo,
LONG      lBuffSize,
LONG      *pIPCCount
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login()
**pDeviceIPCInfo*
[in] added IPC structural buffer
lBuffSize
[in] size of pDeviceIPCInfo(sizeof(NET_SDK_IPC_DEVICE_INFO)* the number of digital channel)
**pIPCCount*
[in] the number of IPC has been added

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_GetDeviceCHStatus

Get the channel information of the NVR, like channel type, online or offline status, etc.

```
BOOL NET_SDK_GetDeviceCHStatus(
LONG            lUserID,
NET_SDK_CH_DEVICE_STATUS* pDeviceCHStatus,
LONG    lBuffSize,
LONG    *pCHCount
);
```

Parameters

lUserID
  [in] the return value of NET_SDK_Login()
**pDeviceCHStatus*
  [in] the connection status of the current channel configured
lBuffSize
  [in] size of pDeviceCHStatus(sizeof(NET_SDK_CH_DEVICE_STATUS) * support how many channels)
**pCHCount*
  [in] the actual numbers of the current channels

Return Values

TRUE means success; FALSE means failure. To get error information,please call NET_SDK_GetLastError

See Also

NET_SDK_Login

Client SDK Instructions

NET_SDK_SetDeviceManualAlarm

set device manual alarm

```
BOOL NET_SDK_SetDeviceManualAlarm(
LONG      lUserID,
LONG      *pAramChannel,
LONG      *pValue,
LONG      lAramChannelCount
BOOL      bAlarmOpen
);
```

Parameters

lUserID
[in] return value of NET_SDK_Login()

**pAramChannel*
[in] List of alarm output channels,an array witch is needed to asign values,its size is **lAramChannelCount**

**pValue*
[in] Alarm channel status(1 means enable alarm channel,0 means disable,if all 0 means all alarm channel is disable)

lAramChannelCount
[in] Number of alarm channels(return value of **NET_SDK_DEVICEINFO::sensorOutputNum**)

BOOL bAlarmOpen
[in] TRUE means open alarm,FALSE means close alarm

Return Values

TRUE means success; FALSE means failure. To get error information, please callNET_SDK_GetLastError

Client SDK Instructions

NET_SDK_GetIVMRuleConfig

Get IVM configuration information of the device(only support IPC).

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

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] the configuration command of the device. Refer to configuration command.
lChannel
[in] the channel number starts from 0
lpOutBuffer
[out] a pointer to the buffer of receiving data.
dwOutBufferSize
[in] the buffer length receiving data (in bytes) can not be zero.
lpBytesReturned
[out] a pointer to the data length actually received can not be NULL

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

The structures and commands are as follows:

dwCommand macro definition	dwCommand value	dwCommand Meanings	structu
IVM_RULE_VFD_CONFIG	0x0	face recognition configuration	NET_SDK_VFD_CON
IVM_RULE_VFD_TRIGGER_CONFIG	0x1	alarm trigger configuration of face recognition	NET_SDK_VFD_TRI
IVM_RULE_VFD_SCHEDULE_CONFIG	0x2	schedule of face recognition	NET_DVR_SCHEDUL
IVM_RULE_AVD_CONFIG	0x3	abnormal video signal detection	NET_SDK_AVD_CON

		configuration	

See Also

NET_SDK_SetDVRConfig NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetIVMRuleConfig

Set the configuration information of the device.(only support IPC)

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

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] the configuration command of the device. Refer to configuration command.
lChannel
[in] the channel number starts from 0
lpInBuffer
[in] a pointer to the buffer of input data
dwOutBufferSize
[in] the buffer length of the input data (in bytes)

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

The structures and commands are as follows:

dwCommand macro definition	dwCommand value	dwCommand Meanings	structu
IVM_RULE_VFD_CONFIG	0x0	face recognition configuration	NET_SDK_VFD_CON
IVM_RULE_VFD_TRIGGER_CONFIG	0x1	alarm trigger configuration of face recognition	NET_SDK_VFD_TRI
IVM_RULE_VFD_SCHEDULE_CONFIG	0x2	schedule of face recognition	NET_DVR_SCHEDUL
IVM_RULE_AVD_CONFIG	0x3	abnormal video signal detection configuration	NET_SDK_AVD_CON

See Also

NET_SDK_GetIVMRuleConfig NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SmartSubscrib

Subscribe the smart alarm events.(only support IPC)

```
BOOL NET_SDK_SmartSubscrib(
LONG     lUserID,
DWORD    dwCommand,
LONG     lChannel,
NET_DVR_SUBSCRIBE_REPLY *pOutBuffer
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] the configuration command of the device. Refer to configuration command.
lChannel
[in] the channel number starts from 0
pOutBuffer
[out] a pointer to the buffer of input data

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

The dwCommand is as follows:

enum definition	value	meanings

NET_DVR_SMART_AVD	0x0	Abnormal video signal diagnosis
NET_DVR_SMART_VFD	0x1	Face detection
NET_DVR_SMART_VFD_MATCH	0x2	Face comparison
NET_DVR_SMART_PEA	0x3	Region intrusion
NET_DVR_SMART_OSC	0x4	Object removal
NET_DVR_SMART_CPC(obsolete)	0x5	People counting
NET_DVR_SMART_CDD	0x6	Crowd density detection
NET_DVR_SMART_IPD	0x7	People intrusion
NET_DVR_SMART_VIHICLE	0x8	Vehicle detection
NET_IPC_SMART_AOIENTRY	0x9	Enter region
NET_IPC_SMART_AOILEAVE	0xA	Leave region
NET_DVR_SMART_VFD_MATCH_FAILED	0xB	Face match failed, for stranger
NET_IPC_SMART_PASSLINE	0xC	pass line

See Also

NET_SDK_UnSmartSubscrib NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_UnSmartSubscrib

Cancel the subscription of smart alarm events(only support IPC).

```
BOOL NET_SDK_UnSmartSubscrib(
LONG     lUserID,
DWORD    dwCommand,
LONG     lChannel,
char  *pInServerAddress,
int  *dwResult
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] the configuration command of the device. Refer to configuration command.
lChannel
[in]the channel number starts from 0
pInServerAddress
[in] a pointer to the buffer of input data
dwResult
[out] the length of output data

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

The structures and commands are as follows:

enum definition	value	meanings
NET_DVR_SMART_AVD	0x0	Abnormal video signal diagnosis
NET_DVR_SMART_VFD	0x1	Face detection
NET_DVR_SMART_VFD_MATCH	0x2	Face comparison
NET_DVR_SMART_PEA	0x3	Region intrusion
NET_DVR_SMART_OSC	0x4	Object removal
NET_DVR_SMART_CPC(obsolete)	0x5	People counting
NET_DVR_SMART_CDD	0x6	Crowd density detection
NET_DVR_SMART_IPD	0x7	People intrusion
NET_DVR_SMART_VIHICLE	0x8	Vehicle detection
NET_IPC_SMART_AOIENTRY	0x9	Enter region
NET_IPC_SMART_AOILEAVE	0xA	Leave region
NET_DVR_SMART_VFD_MATCH_FAILED	0xB	Face match failed, for stranger
NET_IPC_SMART_PASSLINE	0xC	pass line

See Also

NET_SDK_SmartSubscrib NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_SetSubscribCallBack

Set the report and callback of the smart alarm events.

```
BOOL NET_SDK_SetSubscribCallBack(
SUBSCRIBE_CALLBACK     fSubscribCallBack,
void    *pUser,
);
```

Parameters

fSubscribCallBack
  [in] callback function
pUser
  [in] client data

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

SUBSCRIBE_CALLBACK

When the subscribed smart alarm event happens, the uploading analytic data is called back

```
 void *SUBSCRIBE_CALLBACK(
LONG           lUserID,
DWORD          dwCommand,
char           *pBuf,
DWORD          dwBufLen,
void           *pUser
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] the configuration command of the device. Refer to configuration command.
pBuf
[in] the return data （different data types have different structures）
dwBufLen
[out] the reture data length （different data types have different data length）
pUser
[in] user data

Remarks

The dwCommand is as follows:

dwCommand enum	value	mean
NET_SDK_SMART_EVENT_TYPE_AVD	6	Abnoral video sig
NET_SDK_SMART_EVENT_TYPE_VFD	12	Face detection
NET_SDK_SMART_EVENT_TYPE_FACE_MATCH	16	Face comparison
NET_SDK_SMART_EVENT_TYPE_FACE_MATCH_FOR_IPC	17	Face comparison
NET_SDK_SMART_EVENT_TYPE_PEA_FOR_IPC	18	Line crossing and
NET_SDK_SMART_EVENT_TYPE_VEHICE	20	Vehicle number d
NET_SDK_SMART_EVENT_TYPE_PASSLINE	23	pass line

See Also

NET_SDK_UnSmartSubscrib   NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FaceMatchOperate

(only support N9000, not support IPC, IPC refer to NET_SDK_TransparentConfig)The relevant operation of face comparison: whether to support face comparison, face picture database management, face match alarm, getting the data of target.

```
BOOL NET_SDK_FaceMatchOperate(
LONG     lUserID,
DWORD    dwCommand,
LPVOID   lpInBuffer,
DWORD    dwInBufferSize,
LPVOID   lpOutBuffer,
DWORD    dwOutBufferSize,
LPDWORD  lpBytesReturned,
);
```

Parameters

lUserID
[in] the return value of NET_SDK_Login()
dwCommand
[in] Command types refer to configuration command
lpInBuffer
[in] a buffer pointer to send data
dwInBufferSize
[in] the buffer size of sending data (in bytes)
lpOutBuffer
[out] a buffer pointer to receive data
dwOutBufferSize
[in] the buffer size of receiving data (in bytes)
lpBytesReturned
[out] the data length pointer that actually receives can not larger than dwOutBufferSize

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Remarks

Different functions have different structure and commands as shown below.

dwCommand Macro Definition	**dwCommand Value**	**dwCommand Definition**	
NET_SDK_GET_FACE_MATCH_SUPPORT	0x01	Whether to support face comparison or not	NULL

NET_SDK_GET_FACE_INFO_GROUP_LIST	0x02	Get group list	NULL
NET_SDK_ADD_FACE_INFO_GROUP	0x03	Add group	NET_SDK
NET_SDK_SET_FACE_INFO_GROUP	0x04	Edit group	NET_SDK
NET_SDK_DEL_FACE_INFO_GROUP	0x05	Delete group	NET_SDK
NET_SDK_GET_FACE_INFO_LIST	0x06	Get target face list	NET_SDK
NET_SDK_ADD_FACE_INFO	0x07	Add target face	NET_SDK
NET_SDK_SET_FACE_INFO	0x08	Edit face information	NET_SDK
NET_SDK_DEL_FACE_INFO	0x09	Delete face picture	NET_SDK
NET_SDK_GET_FACE_MATCH_ALARM	0x0A	Get face match alarm linkage	NULL
NET_SDK_SET_FACE_MATCH_ALARM	0x0B	Set face match alarm linkage	NET_SDK
NET_SDK_GET_FACE_INFO_IMG	0x0C	Get target face data	NET_SDK
NET_SDK_SEARCH_IMAGE_BY_IMG	0x0D	Search image by image	NET_SDK
NET_SDK_SEARCH_CH_SNAP_FACE_IMG_LIST	0x0E	Search faces of the camera	NET_SDK H
NET_SDK_SEARCH_CH_SNAP_FACE_IMG	0x0F	Search the face information of the camera	NET_SDK

Client SDK Instructions

NET_SDK_SetFishEyeAdjust

(windows only)Set the fisheye correction mode, This interface is called to enter and exit fisheye correction mode both, only in single window mode.

```
BOOL NET_SDK_SetFishEyeAdjust(
POINTERHANDLE lPlayHandle,
FISHEYE_MODE fishEyeMode
);
```

Parameters

lPlayHandle
[in] the handel of play video
fishEyeMode
[in]
fisheye mode:Installation mode + correction mode, The specific defined values are as follows:

```
typedef enum
        {
            FISHEYE_ORIGNAL = 0,  //Original mode
,That is, the fisheye map in the top / wall / bottom mount mode, equivalent to quitting the fisheye
 correction mod

FISHEYE_ROOF = 0x0100,  //Top (suction top)
            FISHEYE_ROOF_360,   //Top-
mounted 360 rectangular expansion panorama +
independent sub-screen;Sub-
frames and rectangular expansion panorama sup
```

port doubling and moving operations，
Rectangular expansion panorama also supports le
ft and right start point movement

FISHEYE_ROOF_2x180,   //Two associated 180 re
ctangular expansion screens of top-mounted，
At any moment, the two sub-
Windows constitute 360 panoramic views, also kn
own as the "double panorama",Both rectangular e
xpansion pictures support the left and right move
ment start point operation, and linkage with each
other;
    FISHEYE_ROOF_FISH_3PTZ, //Top-
mounted original image + 3 independent sub-
images，Both sub-
frames and frames in the original image support d
oubling and moving，
The original image also supports rotation change
start point operations;
    FISHEYE_ROOF_FISH_4PTZ, //Top-
mounted original image + 4 independent subim
ages，Both sub-
frames and frames in the original image support d
oubling and moving，
The original image also supports rotation change
start point operations;
    FISHEYE_ROOF_360_6PTZ, //Top-
mounted 360 rectangular expansion panorama +6
 independent sub-screens，Sub-
frames and rectangular expansion panorama suppo
rt doubling and moving operations，

Rectangular expansion panorama also supports le
ft and right start point movement
            FISHEYE_ROOF_FISH_8PTZ, //Top-
mounted original image + 8 independent sub-
images，Both sub-
frames and frames in the original image support d
oubling and moving，
The original image also supports rotation change
start point operations;
    FISHEYE_WALL = 0x0200, //Wall-mounted
     FISHEYE_WALL_180,   //The 180 wall-
  mounted panorama, from left to right 180 rect
  angular expansion panorama, 180 rectangular
  expansion panorama support up and down mo
  vement operation, change the vertical viewing
  angle;
     FISHEYE_WALL_180_3PTZ, //180 Rectangle
   panoramic panorama   of wall-
  mounted+ 3 independent sub-frames，sub-
  frames and rectangular panoramic panorama s
  upport doubling and moving operations，
  Rectangular expansion panorama supports up
  and down movement, to change the vertical
  perspective
     FISHEYE_WALL_180_4PTZ, //180 Rectangle
   panoramic panorama   of wall-
  mounted+ 4 independent sub-frames，sub-
  frames and rectangular panoramic panorama s
  upport doubling and moving operations，
  Rectangular expansion panorama supports up
  and down movement, to change the vertical

perspective
    FISHEYE_WALL_180_8PTZ, //180 Rectangl
e panoramic panorama   of wall-
mounted+ 8 independent sub-frames, sub-
frames and rectangular panoramic panorama s
upport doubling and moving operations,
Rectangular expansion panorama supports up
and down movement,   to change the vertical
perspective
  FISHEYE_DESKTOP = 0x0300, //Bottom-
mounted(desktop)
    FISHEYE_DESKTOP_360, // 360 rectangula
r expansion panorama of bottom-
mounted+independent sub-frames;sub-
frames and rectangular expansion panorama su
pport doubling and moving operations,
Rectangular expansion panorama also support
s left and right start point movement
    FISHEYE_DESKTOP_2x180, //Two associate
d 180 rectangular expansion screens   of bott
om-mounted, At any moment, the two sub-
Windows constitute 360 panoramic views, also
 known as the "double panorama",Both rectan
gular expansion pictures support the left and ri
ght movement start point operation, and linka
ge with each other;
  FISHEYE_DESKTOP_FISH_3PTZ, //Bottom-
mounted original image + 3 independent sub-
images, Both sub-
frames and frames in the original image suppo
rt doubling and moving,

```
The original image also supports rotation change start point operations;
    FISHEYE_DESKTOP_FISH_4PTZ, //Bottom-mounted original image + 4 independent sub-images，Both sub-frames and frames in the original image support doubling and moving,
The original image also supports rotation change start point operations;
    FISHEYE_DESKTOP_360_6PTZ, //Bottom-mounted 360 rectangular expansion panorama +6 independent sub-screens，Sub-frames and rectangular expansion panorama support doubling and moving operations,
Rectangular expansion panorama also supports left and right start point movement
    FISHEYE_DESKTOP_FISH_8PTZ, //Bottom-mounted original image + 8 independent sub-images，Both sub-frames and frames in the original image support doubling and moving,
The original image also supports rotation change start point operations;
 }FISHEYE_MODE;
```

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FishEyeAdjustFocus

(windows only)set focus, to identify which segmentation region of the current action acts on fisheye correction

```
BOOL NET_SDK_FishEyeAdjustFocus(
POINTERHANDLE lPlayHandle,
int nX,
int nY
);
```

Parameters

lPlayHandle
[in] the handel of play video
nX
[in] the X coordinate value of current focus,relative to the coordinate system of current play window
nY
[in] the Y coordinate value of current focus,relative to the coordinate system of current play window

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FishEyeAdjustFocusEx

(windows only)set focus, to identify which segmentation region of the current action acts on fisheye correction

```
BOOL NET_SDK_FishEyeAdjustFocusEx(
 POINTERHANDLE lPlayHandle,
 int nX,
 int nY,
 int &nIndex
);
```

Parameters

lPlayHandle
[in] the handel of play video
nX
[in] the X coordinate value of current focus,relative to the coordinate system of current play window
nY
[in] the Y coordinate value of current focus,relative to the coordinate system of current play window
nIndex
[in] fccus index

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FishEyeAdjustMove

(windows only)E-cloud platform movement,the segmentation belong to the e-cloud platform can be moved only when it's under the fisheye correction mode

```
 BOOL NET_SDK_FishEyeAdjustMove(
  POINTERHANDLE lPlayHandle,
  int nMoveX,
  int nMoveY
);
```

Parameters

lPlayHandle
[in] the handel of play video
nMoveX
[in] the left mouse button drags horizontally against the X axis of the starting point,positive to the right and negative to the left,with the starting point as the origin
nMoveY
[in] the left mouse button drags vertically against the Y axis of the starting point,positive up and negative down,with the starting point as the origin

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FishEyeAdjustGetArea

(windows only)obtain the correction area location of the current focus

```
BOOL NET_SDK_FishEyeAdjustGetArea(
 POINTERHANDLE lPlayHandle,
 RECT &AreaRect
);
```

Parameters

lPlayHandle
[in] the handel of play video

AreaRect
[in] the correction area location of the current focus,relative to the current play window coordinate system

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

NET_SDK_FishEyeAdjustZoom

(windows only)E-cloud platform amplification,the division belongs to the e-cloud platform can be amplified only when it's into the fisheye correction mode

```
BOOL NET_SDK_FishEyeAdjustZoom(
  POINTERHANDLE lPlayHandle,
  const RECT &ZoomRect
);
```

Parameters

lPlayHandle
  [in] the handel of play video
ZoomRect
  [in] Specifies the area location information to zoom in,relative to the current play window coordinate system

Return Values

TRUE means success; FALSE means failure. To get error information, please call NET_SDK_GetLastError

Client SDK Instructions

DD_ACCOUNT_CONFIG

struct of account configuration

```
struct _dd_account_config{
unsigned long           iSize;
unsigned long           enable;
unsigned long           bindMAC;
unsigned long           group;
char                    MAC [8];
char                    name [DD_MAX_USER_NAME_BUF_LEN];
char                    password [DD_MAX_PASSWORD_BUF_LEN];
unsigned char           logSearch;
unsigned char           systemSetup;
unsigned char           fileManagement;
unsigned char           diskManagement;
unsigned char           remoteLogin;
unsigned char           twoWayAudio;
unsigned char           systemMaintain;
unsigned char           OnlineUserManagement;
unsigned char           shutdown;
unsigned char           alarmOutCtrl;
unsigned char           netAlarm;
unsigned char           netSerialCtrl;
unsigned char           authLive;
unsigned char           authRecord;
unsigned char           authPlayback;
unsigned char           authBackup;
unsigned char           authPTZ;
unsigned char           netAuthView;
unsigned char           netauthRecord;
unsigned char           netauthPlayback;
unsigned char           netauthBackup;
unsigned char           netauthPTZ;
unsigned char           recv[2];
unsigned char           authLiveCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           authRecordCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           authPlaybackCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           authBackupCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           authPTZCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           netAuthViewCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           netAuthRecordCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           netAuthPlaybackCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           netAuthBackupCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char           netAuthPTZCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
}DD_ACCOUNT_CONFIG;
```

Members

iSize
size of the struct
enable
whether to use the account
bindMAC
whether to bind MAC
group
the group belonged to,refer to DD_USER_GROUP:

Group Name	Value	Description
DD_USER_GROUP_NONE	0x00	

DD_USER_GROUP_ADMIN	0x01	administrator,have all rights
DD_USER_GROUP_ADVANCE	0x02	advanced user,default rights:basic,record,config,playback,backup,data management,disk management,PTZ control,remote login and all channels rights
DD_USER_GROUP_NORMAL	0x03	normal user,default rights:basic,record,playback,backup,PTZ control,remote login and all channels rights

MAC [8]
  binded MAC
name [DD_MAX_USER_NAME_BUF_LEN]
  user name
password [DD_MAX_PASSWORD_BUF_LEN]
  password
logSearch
  limit of log search
systemSetup
  system configuration
fileManagement
  file management
diskManagement
  disc management
remoteLogin
  remote login
twoWayAudio
  audio talkback
systemMaintain
  system maintain
OnlineUserManagement
  online user management
shutdown
  shutdown or reboot
alarmOutCtrl
  alarm output control
netAlarm
  network alarm
netSerialCtrl
  network serial port control
authLive
  live preview
authRecord
  local record
authPlayback
  local search playback
authBackup
  local backup
authPTZ
  local PTZ
netAuthView
  local control

netauthRecord
  remote record
netauthPlayback
  remote live playback
netauthBackup
  remote backup
netauthPTZ
  remote PTZ
recv[2]
  reserved bytes
authLiveCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  live preview channel
authRecordCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  local record manually
authPlaybackCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  local search and playback
authBackupCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  local backup
authPTZCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  local PTZ control
netAuthViewCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  remote live preview
netAuthRecordCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  remote record manually
netAuthPlaybackCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  remote playback
netAuthBackupCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  remote backup
netAuthPTZCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  remote PTZ control

Client SDK Instructions

DD_AREA

area struct

```
struct  _dd_area_{
unsigned short x;
unsigned short y;
unsigned short cx;
unsigned short cy;
}DD_AREA;
```

Members

x
: abscissa,range 0-99

y
: ordinate,range 0-99

cx
: width,range 1-100

cy
: height,range 1-100

Notice: x+cx <= 100, y+cy <=100

Client SDK Instructions

DD_AUTO_REPORT

Passively receive the struct of DVR register.

```
struct  _dd_auto_report_{
unsigned long   bUse;
char            host[256];
unsigned long   dwPort;
unsigned long   ID;
}DD_AUTO_REPORT;
```

Members

bUse
  Whether to enable auto report register function
host[256]
  server address of register platform
dwPort
  server port of register platform
ID
  assigned register ID

Client SDK Instructions

DD_BASIC_CONFIG

struct of basic configuration information

```
struct _dd_basic_config_{
unsigned long iSize;
unsigned long videoFormat;
unsigned long videoOut;
unsigned long videoOutResolution;
unsigned long VGARefresh;
unsigned long screensaver;
unsigned long deviceLanguage;
unsigned long passwordCheck;
unsigned long RecycleRecord;
unsigned long videoFormatMask;
unsigned long videoOutMask;
unsigned long videoOutResolutionMask;
unsigned long languageMask;
}DD_BASIC_CONFIG;
```

Members

iSize
size of the struct

videoFormat
video format, refer to DD_VIDEO_FORMAT:

Type	Value
DD_VIDEO_FORMAT_NTSC	0x01
DD_VIDEO_FORMAT_PAL	0x02

videoOut
video output device(reserved)

videoOutResolution
video output resolution,refer to DD_VGA_RESOLUTION:

Type	Value
DD_VGA_640X480	0x0001
DD_VGA_720X480	0x0002
DD_VGA_720X576	0x0004
DD_VGA_800X600	0x0008
DD_VGA_1024X768	0x0010
DD_VGA_1280X960	0x0020
DD_VGA_1280X1024	0x0040
DD_VGA_1920X1080	0x0080

VGARefresh
VGA refresh rate(reserved)
screensaver
screensaver time(0 means close)
deviceLanguage
device language
passwordCheck
whether to open password check
RecycleRecord
whether permit overlap record
videoFormatMask
supportive video format mask(read only)
videoOutMask
supportive video output device mask(read only)
videoOutResolutionMask
supportive video output device resolution mask(read only)
languageMask
language mask group supported by device(read only),refer to the list below:

类型	对应值
LANGUAGE_ENGLISH	0x0000001

LANGUAGE_CHINESE_S	0x0000002
LANGUAGE_CHINESE_B	0x0000004
LANGUAGE_PORTUGUESE	0x0000008
LANGUAGE_GREECE	0x0000010
LANGUAGE_SPAISH	0x0000020
LANGUAGE_RUSSIAN	0x0000040
LANGUAGE_NORWAY	0x0000080
LANGUAGE_TURKEY	0x0000100
LANGUAGE_ITALY	0x0000200
LANGUAGE_CZECH	0x0000400
LANGUAGE_GERMAN	0x0000800
LANGUAGE_HEBREW	0x0001000
LANGUAGE_JAPANESE	0x0002000
LANGUAGE_FRENCH	0x0004000
LANGUAGE_POLISH	0x0008000
LANGUAGE_BULGARIAN	0x0010000
LANGUAGE_INDONESIA	0x0020000
LANGUAGE_RUSSIAN_D	0x0040000
LANGUAGE_THAI	0x0080000
LANGUAGE_HUNGARY	0x0100000
LANGUAGE_LITHUANIA	0x0200000

Client SDK Instructions

DD_BUZZER_CONFIG

struct of buzzer configuration

```
struct _dd_buzzer_config_{
unsigned char                enable;
unsigned char                recv;
unsigned short               holdTime;
}DD_BUZZER_CONFIG;
```

Members

enable
: buzzer enable switch

recv
: reserved bytes

holdTime
: delay time

Client SDK Instructions

DD_CHANNEL_CONFIG

struct of channel configuration information

```
struct _dd_channel_config_{
unsigned long    iSize;
unsigned long    hide;
char              name [DD_MAX_CAMERA_NAME_BUF_LEN];
}DD_CHANNEL_CONFIG;
```

Members

iSize
size of the struct

hide
whether hide channel

name [DD_MAX_CAMERA_NAME_BUF_LEN]
channel name

Client SDK Instructions

DD_CRUISE_POINT_INFO

struct of setting cruise position information.

```
struct _dd_cruise_point_info{
unsigned long                  presetIndex;
unsigned long                  dwellSpeed;
unsigned long                  dwellTime;
}DD_CRUISE_POINT_INFO;
```

Members

presetIndex
  index of cruise position(1-128).
dwellSpeed
  speed of cruise(1-8).
dwellTime
  seconds of cruise .

Client SDK Instructions

DD_DATE

struct of date configuration infomation of device.

```
struct _dd_date_{
unsigned char     mday;
unsigned char     month;
unsigned short    year;
}DD_DATE,  *LP_DD_DATE;
```

Members

mday
: day of month, range(1-31).

month
: month, range(1-12).

year
: current solar year.

Client SDK Instructions

DD_DATE_SCHEDULE

struct of data schedule

```
struct _dd_date_schedule_{
unsigned long long       hour [24];
}DD_DATE_SCHEDULE;
```

Members

hour [24]

data schedule formation,24 stands for 24 hours format,each position of unsigned long long stands for each minute's state

Client SDK Instructions

DD_DATE_TIME_CONFIG

struct of data and time schedule configuration

```
struct _dd_date_time_config{
unsigned long           iSize;
unsigned char           dateFormat;
unsigned char           timeFormat;
unsigned char           timeZone;
unsigned char           enableNTP;
unsigned short     ntpPort;
unsigned short     recv;
char                     ntpServerAddr[DD_MAX_URL_BUF_LEN];
}DD_DATE_TIME_CONFIG;
```

Members

iSize
size of the struct

dateFormat
data format, refer to DD_DATE_MODE:

Type	Value	Description
DD_DATE_MODE_YMD	0x01	YMD format
DD_DATE_MODE_MDY	0x02	MDY format
DD_DATE_MODE_DMY	0x03	DMY format

timeFormat
time format,refer to the list below:

Type	Value	Description
TIME_MODE_12	0x01	12 hours
TIME_MODE_24	0x02	24 hours

timeZone
time zone, refer to DD_TIME_ZOME_NAME:

Type	Value
DD_TIME_ZONE_GMT_D12	0
DD_TIME_ZONE_GMT_D11	1
DD_TIME_ZONE_GMT_D10	2
DD_TIME_ZONE_GMT_D9	3
DD_TIME_ZONE_GMT_D8	4
DD_TIME_ZONE_GMT_D7	5
DD_TIME_ZONE_GMT_D6	6
DD_TIME_ZONE_GMT_D5	7
DD_TIME_ZONE_GMT_D4_30	8
DD_TIME_ZONE_GMT_D4	9
DD_TIME_ZONE_GMT_D3_30	10
DD_TIME_ZONE_GMT_D3	11
DD_TIME_ZONE_GMT_D2	12
DD_TIME_ZONE_GMT_D1	13
DD_TIME_ZONE_GMT	14
DD_TIME_ZONE_GMT_A1	15
DD_TIME_ZONE_GMT_A2	16
DD_TIME_ZONE_GMT_A3	17
DD_TIME_ZONE_GMT_A3_30	18
DD_TIME_ZONE_GMT_A4	19
DD_TIME_ZONE_GMT_A4_30	20
DD_TIME_ZONE_GMT_A5	21
DD_TIME_ZONE_GMT_A5_30	22
DD_TIME_ZONE_GMT_A5_45	23
DD_TIME_ZONE_GMT_A6	24
DD_TIME_ZONE_GMT_A6_30	25
DD_TIME_ZONE_GMT_A7	26
DD_TIME_ZONE_GMT_A8	27
DD_TIME_ZONE_GMT_A9	28

DD_TIME_ZONE_GMT_A9_30	29
DD_TIME_ZONE_GMT_A10	30
DD_TIME_ZONE_GMT_A11	31
DD_TIME_ZONE_GMT_A12	32
DD_TIME_ZONE_GMT_A13	33

enableNTP
whether open NTP synchronization service

ntpPort
NTP port

ntpServerAddr[DD_MAX_URL_BUF_LEN]
NTP service address

Client SDK Instructions

DD_DAYLIGHT_INFO

struct of daylight saving time information

```
struct _dd_daylight_info_{
unsigned char          InMonth;
unsigned char          InMday;
unsigned char          OutMonth;
unsigned char          OutMday;
unsigned char          InWeekIndex;
unsigned char          InWday;
unsigned char          OutWeekIndex;
unsigned char          OutWday;
unsigned short         InYear;
unsigned short         OutYear;
unsigned short         enable;
unsigned short         type;
unsigned long          InSecond;
unsigned long          OutSecond;
unsigned long          offSet;
}DD_DAYLIGHT_INFO;
```

Members

InMonth
which month to enter DST

InMday
which day to enter DST(data mode is valid)

OutMonth
which month to exit DST

OutMday
which day to exit DST(data mode is valid)

InWeekIndex
which week to enter DST(week mode is valid)

InWday
which weekday to enter DST(week mode is valid)

OutWeekIndex
which week to exit DST(week mode is valid)

OutWday

which weekday to exit DST(week mode is valid)

InYear
which year to enter DST,reserved due to align the struct

OutYear
which year to exit DST,reserved due to align the struct

enable
whether enable DST function

type
DST setting modern: week or data mode

InSecond
second offset in one day of DST(0-86399),it can switch to be hour and minute and second

OutSecond
second offset out of one day of DST(0-86399),it can switch to be hour and minute and second

offSet
offset second in DST(0-86399)

Client SDK Instructions

DD_DDNS_CONFIG

struct of DDNS configuration

```
struct _dd_ddns_config_{
unsigned long            iSize;
unsigned short           enable;
unsigned short           interval;
unsigned long            useDDNSServer;
unsigned long            userHostDomain;
char                     userName [DD_MAX_DDNS_ACCOUNT_BUF_LEN];
char                     password [DD_MAX_PASSWORD_BUF_LEN];
char                     hostDomain [DD_MAX_URL_BUF_LEN];
}DD_DDNS_CONFIG;
```

Members

iSize
  size of the struct
enable
  whether enable DDNS
interval
  report upgrade interval
useDDNSServer
  type or address of DDNS server in use
userHostDomain
  whether enable host domain name
userName [DD_MAX_DDNS_ACCOUNT_BUF_LEN]
  DDNS account
password [DD_MAX_PASSWORD_BUF_LEN]
  DDNS password
hostDomain [DD_MAX_URL_BUF_LEN]
  host domain name(correspond to a certain protocol,specifying server is permitted)

Client SDK Instructions

DD_DDNS_SERVER_INFO

struct of DDNS server information

```
struct _dd_ddns_server_info{
unsigned char     DDNSID;
unsigned char     supportproperty;
unsigned char     noused[2];
char              DDNSServerName[64];
}DD_DDNS_SERVER_INFO;
```

Members

DDNSID
: DDNS ID,server name is valid only if ID value is greater than 0

supportproperty
: NCFG_ENUM_DDNS_SUPPORT_DOMAIN1 =0x01(support domain 1,maybe need to support two domains)

noused[2]
: unenable DDNS server

DDNSServerName[64]
: address of DDNS server

Client SDK Instructions

DD_DEVICEINFO

struct of device basic information.

```
struct _dd_device_info_{
unsigned long          iSize;
unsigned long           deviceID;
char                   deviceNo[DD_MAX_SERIAL_NUMBER_LEN];
char                   deviceName [DD_MAX_NAME_LEN];
char                   firmwareVersion [DD_MAX_VERSION_BUF_LEN];
char                   firmwareBuildDate [DD_MAX_VERSION_BUF_LEN];
char                   hardwareVersion [DD_MAX_VERSION_BUF_LEN];
char                   kernelVersion [DD_MAX_VERSION_BUF_LEN];
char                   mcuVersion [DD_MAX_VERSION_BUF_LEN];
unsigned char          audioNum;
unsigned char          localVideoInNum;
unsigned char          netVideoInNum;
unsigned char          sensorInNum;
unsigned char          relayOutNum;
unsigned char          rs232Num;
unsigned char          rs485Num;
unsigned char          networkPortNum;
unsigned char          diskCtrlNum;
unsigned char          DiskNum;
unsigned char          vgaNum;
unsigned char          usbNum;
}DD_DEVICE_INFO;
```

Members

iSize
size of the struct.
deviceID
device ID(0~255).
deviceNo[DD_MAX_SERIAL_NUMBER_LEN]
serial number of device,letter is usable.
deviceName [DD_MAX_NAME_LEN]
device name(attention to double byte character).
firmwareVersion [DD_MAX_VERSION_BUF_LEN]
software version number.
firmwareBuildDate [DD_MAX_VERSION_BUF_LEN]
software building date.
hardwareVersion [DD_MAX_VERSION_BUF_LEN]
hardware version.
kernelVersion [DD_MAX_VERSION_BUF_LEN]
system core version.

mcuVersion [DD_MAX_VERSION_BUF_LEN]
MCU version.
audioNum
audio number.
localVideoInNum
channel number of local video input.
netVideoInNum
channel number of network video input.
sensorInNum
number of sensor for input.
relayOutNum
number of relay for output.
rs232Num
channel number of 232 remote sensing.
rs485Num
channel number of 485 remote sensing.
networkPortNum
number of network port.
diskCtrlNum
number of harddisk for control.
DiskNum
number of harddisk.
vgaNum
number of displayer.
usbNum
number of USB soket.

Client SDK Instructions

DD_ENCODE_CONFIG

struct of encoding configuration

```
struct _dd_encode_config_{
unsigned long       iSize;
unsigned short      resolution;
unsigned short      rate;
unsigned short      encodeType;
unsigned short      quality;
unsigned short      minBitrate;
unsigned short      maxBitrate;
}DD_ENCODE_CONFIG;
```

Members

iSize

size of the struct

resolution

resolution, refer to DD_VIDEO_SIZE:

Type	Value	Video Format
DD_VIDEO_SIZE_QCIF	0x0001	QCIF
DD_VIDEO_SIZE_CIF	0x0002	CIF
DD_VIDEO_SIZE_HD1	0x0004	HD1
DD_VIDEO_SIZE_D1	0x0008	D1
DD_VIDEO_SIZE_QVGA	0x0010	QVGA
DD_VIDEO_SIZE_VGA	0x0020	VGA
DD_VIDEO_SIZE_XVGA	0x0040	XVGA
DD_VIDEO_SIZE_QQVGA	0x0080	QQVGA
DD_VIDEO_SIZE_480P	0x0100	480P
DD_VIDEO_SIZE_720P	0x0200	720P
DD_VIDEO_SIZE_1080P	0x0400	1080P

rate

frame rate

encodeType

encoding type,refer to the following list:

rate

frame rate

encodeType

encoding type,refer to the following list:

type	Value	Description
DD_VIDEO_ENCODE_MODE_VBR	0x01	mutable code stream
DD_VIDEO_ENCODE_MODE_CBR	0x02	fixed code stream

quality

image quality, refer to DD_IMAGE_QUALITY:

Type	Value	Description
DD_IMAGE_QUALITY_LOWEST	0x01	lowest image quality
DD_IMAGE_QUALITY_LOWER	0x02	lower image quality
DD_IMAGE_QUALITY_LOW	0x03	low image quality
DD_IMAGE_QUALITY_MEDIUM	0x04	medium image quality
DD_IMAGE_QUALITY_HEIGHTER	0x05	heighter image quality
DD_IMAGE_QUALITY_HEIGHTEST	0x06	heightest

		image quality

minBitrate
  code stream lower limit,in kbps
maxBitrate
  code stream upper limit,in kbps

Client SDK Instructions

DD_ENCODE_CONFIG_SUPPORT

struct of encode config supported by device

```
 struct   _dd_encode_config_support_
{
       DD_ENCODE_CONFIG encodeConfig[DD_MAX_SUPPORT_RESOLUTION];
        unsigned long      num;
}DD_ENCODE_CONFIG_SUPPORT;
```

Members

encodeConfig[DD_MAX_SUPPORT_RESOLUTION]
: DD_MAX_SUPPORT_RESOLUTION equals 7, which is the max supported resolution type num, encodeConfig includes main stream(sub stream) resolution, fps, max and min bitrate.

num
: the real supported resolution num by device

Client SDK Instructions

DD_FRAME_INFO

struct of data frame information

```
struct _dd_frame_info_{
unsigned long          frameType;
unsigned long          length;
unsigned long          keyFrame;
unsigned long             width;
unsigned long             height;
unsigned long             *pData;
unsigned short            deviceIndex;
unsigned short            channel;
unsigned long             bufIndex;
unsigned long          frameIndex;
unsigned long          frameAttrib;
unsigned long          streamID;
LONGLONG                       time;
LONGLONG                       relativeTime;
DD_TIME                   localTime;
}DD_FRAME_INFO, *LP_DD_FRAME_INFO;
```

Members

frameType

data frame type, refer to DD_FRAME_TYPE:

Type	Value
DD_FRAME_TYPE_NONE	0x00
DD_FRAME_TYPE_VIDEO	0x01
DD_FRAME_TYPE_AUDIO	0x02
DD_FRAME_TYPE_TALK_AUDIO	0x03
DD_FRAME_TYPE_JPEG	0x04
DD_FRAME_TYPE_VIDEO_FORMAT	0x05
DD_FRAME_TYPE_AUDIO_FORMAT	0x06
DD_FRAME_TYPE_TALK_AUDIO_FORMAT	0x07
DD_FRAME_TYPE_END	

length

data length

keyFrame

keyframe,0:non-key-frame,1:keyframe

width

width of data frame

height

height of data frame

**pData*

pointer to data

deviceIndex

device index number

channel

data channel

bufIndex

buffer area index

frameIndex

data frame index

frameAttrib

data frame attribute, refer to DD_FRAME_ATTRIB:

Type	Value	Description
DD_PLAY_FRAME_NO_SHOW	0x01	no show this frame
DD_PLAY_FRAME_SHOW	0x02	the frame can be showed
DD_PLAY_FRAME_ALL_END	0x04	reading data is finished,no more data
DD_PLAY_FRAME_SEC_END	0x08	the event section is ended
DD_PLAY_FRAME_NO_TIME_STAMP	0x10	the frame includes timestamp,shield time function when capture
DD_PLAY_FRAME_FF	0x20	the frame applied to fastforward
DD_LIVE_FRAME_FIRST_STREAM	0x40	the frame is live main code

		stream
DD_LIVE_FRAME_SECOND_STREAM	0x80	the frame is live sub code stream
DD_LIVE_FRAME_JPEG	0x100	the frame is JPEG image
DD_LIVE_FRAME_TALK	0x200	the frame is talkback audio data

streamID
data stream ID

time
absolute time,calculate from 00:00:00 on Jan.1st in 1970,in microsecond,it changes when change device time

relativeTime
relative time,in microsecond,it won't change when change device time,because it is continuous

localTime
device local time, later fill in

Client SDK Instructions

DD_LIVE_AUDIO_GROUP

struct of audio group

```
struct _dd_live_audio_group_{
unsigned short      holdTime;
unsigned char       volume;
unsigned char       channel;
}DD_LIVE_AUDIO_GROUP;
```

Members

holdTime
: hold time(in second), 0 means invalid

volume
: volume(0-100)

channel
: channel number,start from 0

Client SDK Instructions

DD_LIVE_DISPLAY

struct of real time display

```
struct _dd_live_display_{
unsigned long        iSize;
unsigned long        showTime;
unsigned long        showNetwork;
unsigned long        showHDD;
unsigned long        showUSB;
unsigned short       alarmInNum;
unsigned short       alarmOutNum;
unsigned long          showAlarmIn;
unsigned long          showAlarmOut;
unsigned long        cameraNum;
unsigned char        showCameraName [DD_MAX_CAMERA_NUM];
unsigned char        showRecordStatus [DD_MAX_CAMERA_NUM];
}DD_LIVE_DISPLAY;
```

Members

iSize
  size of the struct
showTime
  whether show system time
showNetwork
  whether show network status
showHDD
  whether show harddisc information
showUSB
  whether show movable storage infomation
alarmInNum
  alarm input number(read only)
alarmOutNum
  alarm output number(read only)
showAlarmIn
  whether show alarm input information
showAlarmOut
  whether show alarm output information

cameraNum
valid channel number(read only)
showCameraName [DD_MAX_CAMERA_NUM]
whether show channel name
showRecordStatus [DD_MAX_CAMERA_NUM]
whether show record status

Client SDK Instructions

DD_LIVE_VIDEO_GROUP

struct of preview video group

```
struct _dd_live_video_group_{
unsigned short holdTime;
unsigned short channelNum;
unsigned long         splitMode;
unsigned char         channel [DD_MAX_CAMERA_NUM];
}DD_LIVE_VIDEO_GROUP;
```

Members

holdTime
hold time,(in second), 0 means invalid
channelNum
valid channel number(read only)
splitMode
split mode, refer to DD_VIEW_SPLIT_MODE:

type
DD_VIEW_SPLIT_1X1
DD_VIEW_SPLIT_2X2
DD_VIEW_SPLIT_1A2/ DD_VIEW_SPLIT_2X3
DD_VIEW_SPLIT_1A5/DD_VIEW_SPLIT_3X3
DD_VIEW_SPLIT_1A7/DD_VIEW_SPLIT_1A12/DD_VIEW_SPLIT_4X4
DD_VIEW_SPLIT_2A6/DD_VIEW_SPLIT_4X6
DD_VIEW_SPLIT_1A9/DD_VIEW_SPLIT_4A9/DD_VIEW_SPLIT_1A16/DD_VIEW_SPL
DD_VIEW_SPLIT_1A11/DD_VIEW_SPLIT_1A20/DD_VIEW_SPLIT_4A20/DD_VIEW_S

channel [DD_MAX_CAMERA_NUM]
channel number corresponding to each area,array index stands for channel number,element value stands for window area number,0xff means invalid channel

Client SDK Instructions

DD_LOG_INFO

struct of log information

```
struct _dd_log_info_{
unsigned long      majorType;
unsigned long      minorType;
unsigned long      time;
unsigned long      IP;
char              name [36];
DD_TIME          localTime;
unsigned long      infoLen;
char              info[1024];
}DD_LOG_INFO, *LP_DD_LOG_INFO;
```

Members

majorType
major type, refer to DD_LOG_CONTENT:

Type	Value
DD_LOG_CONTENT_SYSTEM_CTRL	0x00000001
DD_LOG_CONTENT_CONFIG	0x00000002
DD_LOG_CONTENT_PLAYBACK	0x00000004
DD_LOG_CONTENT_BACKUP	0x00000008
DD_LOG_CONTENT_SEARCH	0x00000010
DD_LOG_CONTENT_VIEW_INFO	0x00000020
DD_LOG_CONTENT_EVENT_INFO	0x00000040
DD_LOG_CONTENT_ERROR_INFO	0x00000080

minorType
minor type, refer to DD_LOG_TYPE:

DD_LOG_TYPE_SYSTEM_CTRL	**0x01000000**
Type	**description**
DD_LOG_TYPE_BOOT	boot system
DD_LOG_TYPE_SHUTDOWN	shutdown

	system
DD_LOG_TYPE_REBOOT	reboot system
DD_LOG_TYPE_FORMAT_SUCC	format disc successfully
DD_LOG_TYPE_FORMAT_FAIL	formatting disc fail
DD_LOG_TYPE_UPGRADE_SUCC	upgrade successfully
DD_LOG_TYPE_UPGRADE_FAIL	upgrade fail
DD_LOG_TYPE_CLEAR_ALARM	clear alarm
DD_LOG_TYPE_OPEN_ALARM	open alarm
DD_LOG_TYPE_MANUAL_START	open manual record
DD_LOG_TYPE_MANUAL_STOP	stop manual record
DD_LOG_TYPE_PTZ_ENTER	start PTZ control
DD_LOG_TYPE_PTZ_CTRL	PTZ operation
DD_LOG_TYPE_PTZ_EXIT	exit PTZ control
DD_LOG_TYPE_AUDIO_CH_CHANGE	chang audio channel
DD_LOG_TYPE_VOLUME_ADJUST	adjust volume
DD_LOG_TYPE_MUTE_ENABLE	enable mute
DD_LOG_TYPE_MUTE_DISENABLE	disenable mute
DD_LOG_TYPE_DWELL_ENABLE	enable dwell
DD_LOG_TYPE_DWELL_DISENABLE	disenable dwell
DD_LOG_TYPE_LOG_IN	login
DD_LOG_TYPE_LOG_OFF	logout
DD_LOG_TYPE_CHANGE_TIME	change system time
DD_LOG_TYPE_MANUAL_SNAP_SUCC	manual

	capture succeed
DD_LOG_TYPE_MANUAL_SNAP_FAIL	manual capture fail
DD_LOG_TYPE_CONFIG	**0x02000000**
DD_LOG_TYPE_CHGE_VIDEO_FORMAT	change video format
DD_LOG_TYPE_CHGE_VGA_RESOLUTION	change VGA resolution
DD_LOG_TYPE_CHGE_LANGUAGE	change language
DD_LOG_TYPE_CHGE_NET_USER_NUM	change network user number
DD_LOG_TYPE_CHGE_TIME_ZONE	change time zone
DD_LOG_TYPE_NTP_MANUAL	manual network time check
DD_LOG_TYPE_NTP_ON	enable automatic network time check
DD_LOG_TYPE_NTP_OFF	disenable automatic network time check
DD_LOG_TYPE_CHGE_NTP_SERVER	change network time server address
DD_LOG_TYPE_CHGE_DST	change daylight saving time setting
DD_LOG_TYPE_PASSWD_ON	enable

	operation password
DD_LOG_TYPE_PASSWD_OFF	disappear operation password
DD_LOG_TYPE_CHGE_CAM_NAME	change channel name
DD_LOG_TYPE_MODIFY_COLOR	modify color
DD_LOG_TYPE_CHGE_HOST_MONITOR	change host monitor image setting
DD_LOG_TYPE_CHGE_SPOT	change auxiliary output image setting
DD_LOG_TYPE_CHGE_OSD	change character overlap setting
DD_LOG_TYPE_CHGE_LOCAL_ENCODE	change encoding parameter of record stream
DD_LOG_TYPE_CHGE_REC_VIDEO_SWITCH	change record video switch setting
DD_LOG_TYPE_CHGE_REC_AUDIO_SWITCH	change record audio switch setting
DD_LOG_TYPE_CHGE_REC_REDU_SWITCH	change redundant record switch setting
DD_LOG_TYPE_CHGE_REC_PRE_TIME	change the time before record
DD_LOG_TYPE_CHGE_REC_POST_TIME	change the

	time after record
DD_LOG_TYPE_CHGE_REC_HOLD_TIME	change record data expiry time
DD_LOG_TYPE_CHGE_SCH_SCHEDULE	change the plan of regular record
DD_LOG_TYPE_CHGE_SCH_MOTION	change motion detection record schedule
DD_LOG_TYPE_CHGE_SCH_ALARM	change alarm record schedule
DD_LOG_TYPE_CHGE_SENSOR_SWITCH	change alarm input switch setting
DD_LOG_TYPE_CHGE_SENSOR_TYPE	change alarm input sensor type
DD_LOG_TYPE_CHGE_SENSOR_TRIGGER	change alarm input semsor trigger setting
DD_LOG_TYPE_CHGE_SENSOR_SCH	change alarm input detection schedule
DD_LOG_TYPE_CHGE_MOTION_SWITCH	change motion detection switch setting
DD_LOG_TYPE_CHGE_MOTION_SENS	change motion

	detection sensitivity
DD_LOG_TYPE_CHGE_MOTION_AREA	change motion detection area setting
DD_LOG_TYPE_CHGE_MOTION_TRIGGER	change motion detection process mode
DD_LOG_TYPE_CHGE_MOTION_SCH	change motion detection schedule
DD_LOG_TYPE_CHGE_VL_TRIGGER	change video lost process mode setting
DD_LOG_TYPE_CHGE_RELAY_SWITCH	change alarm output relay setting
DD_LOG_TYPE_CHGE_RELAY_SCH	change alarm output schedule
DD_LOG_TYPE_BUZZER_ON	enable buzzer alarm
DD_LOG_TYPE_BUZZER_OFF	disenable buzzer alarm
DD_LOG_TYPE_CHGE_BUZZER_SCH	change buzzer alarm schedule
DD_LOG_TYPE_CHGE_HTTP_PORT	modify HTTP server port
DD_LOG_TYPE_CHGE_SER_PORT	modify network server port
DD_LOG_TYPE_CHGE_IP	change network IP

DD_LOG_TYPE_DHCP_SUCC	obtain DHCP automatically succeed
DD_LOG_TYPE_DHCP_FAIL	obtain DHCP automatically fail
DD_LOG_TYPE_CHGE_PPPOE	set PPPoE
DD_LOG_TYPE_CHGE_DDNS	set DDNS
DD_LOG_TYPE_NET_STREAM_CFG	change network stream edcoding setting
DD_LOG_TYPE_CHGE_SERIAL	change PTZ serial port setting
DD_LOG_TYPE_PRESET_MODIFY	modify preset point
DD_LOG_TYPE_CRUISE_MODIFY	modify cruise line
DD_LOG_TYPE_TRACK_MODIFY	modify track
DD_LOG_TYPE_USER_ADD	add users
DD_LOG_TYPE_USER_MODIFY	modify user authority
DD_LOG_TYPE_USER_DELETE	delete user
DD_LOG_TYPE_CHANGE_PASSWD	modify user password
DD_LOG_TYPE_LOAD_DEFAULT	recover default configuration
DD_LOG_TYPE_IMPORT_CONFIG	import configuration
DD_LOG_TYPE_EXPORT_CONFIG	export configuration
DD_LOG_TYPE_CHGE_IMAGE_MASK	image shield
DD_LOG_TYPE_RECYCLE_REC_ON	enable loop

	record
DD_LOG_TYPE_RECYCLE_REC_OFF	close loop record
DD_LOG_TYPE_CHGE_DISK_ALARM	change disc alarm space
DD_LOG_TYPE_CHGE_SEND_EMAIL	set Email sender information
DD_LOG_TYPE_CHGE_RECV_EMAIL	set Email receiver information
DD_LOG_TYPE_CHGE_SNAP_SETTING	change capture configuration
DD_LOG_TYPE_PLAYBACK	**0x03000000**
DD_LOG_TYPE_PLAYBACK_PLAY	play
DD_LOG_TYPE_PLAYBACK_PAUSE	pause
DD_LOG_TYPE_PLAYBACK_RESUME	resume play
DD_LOG_TYPE_PLAYBACK_FF	fast forward
DD_LOG_TYPE_PLAYBACK_REW	rewind
DD_LOG_TYPE_PLAYBACK_STOP	stop
DD_LOG_TYPE_PLAYBACK_NEXT_SECTION	play next section
DD_LOG_TYPE_PLAYBACK_PREV_SECTION	play previous section
DD_LOG_TYPE_BACKUP	**0x04000000**
DD_LOG_TYPE_BACKUP_START	start to backup
DD_LOG_TYPE_BACKUP_COMPLETE	backup is completed
DD_LOG_TYPE_BACKUP_CANCEL	cancel backup
DD_LOG_TYPE_BACKUP_FAIL	backup fails
DD_LOG_TYPE_SEARCH	**0x05000000**
DD_LOG_TYPE_SEARCH_TIME	search by

	time
DD_LOG_TYPE_SEARCH_EVENT	search by event
DD_LOG_TYPE_SEARCH_FILE_MAN	search file management
DD_LOG_TYPE_SEARCH_PICTURE	search picture
DD_LOG_TYPE_DELETE_FILE	delete file
DD_LOG_TYPE_LOCK_FILE	lock file
DD_LOG_TYPE_UNLOCK_FILE	unlock file
DD_LOG_TYPE_DELETE_PICTURE	delete picture
DD_LOG_TYPE_LOCK_PICTURE	lock picture
DD_LOG_TYPE_UNLOCK_PICTURE	unlock picture
DD_LOG_TYPE_EVENT_INFO	**0x07000000**
DD_LOG_TYPE_SENSOR_START	start sensor alarm
DD_LOG_TYPE_SENSOR_END	sensor alarm ends
DD_LOG_TYPE_MOTION_START	motion detection starts
DD_LOG_TYPE_MOTION_END	motion detection ends
DD_LOG_TYPE_VLOSS_START	video loss start
DD_LOG_TYPE_VLOSS_END	video loss ends
DD_LOG_TYPE_SHELTER_START	video shelter starts
DD_LOG_TYPE_SHELTER_END	video shelter ends
DD_LOG_TYPE_BEHAVIOR_INFO	**0x08000000**
DD_LOG_TYPE_ENTER_AREA	enter area
DD_LOG_TYPE_EXIT_AREA	exit area

DD_LOG_TYPE_INTRUSION	intrusion
DD_LOG_TYPE_LOITER	loiter
DD_LOG_TYPE_LEFT_TAKE	across left cordon
DD_LOG_TYPE_PARKING	parking
DD_LOG_TYPE_RUN	run
DD_LOG_TYPE_HIGH_DENSITY	high density behaviour
DD_LOG_TYPE_ERROR_INFO	**0x09000000**
DD_LOG_TYPE_IP_CONFLICT	network IP conflict
DD_LOG_TYPE_NETWORK_ERR	network exception
DD_LOG_TYPE_DDNS_ERR	DDNS error
DD_LOG_TYPE_DISK_IO_ERR	disc read-write error
DD_LOG_TYPE_UNKNOWN_OFF	electricity outage exception
DD_LOG_TYPE_UNKNOWN_ERR	unknown error

time
  log occurrence time
IP
  user IP
name [36]
  user name
localTime
  local time, later fill in
infoLen
  length of log information
info[1024]
  length of the log information

Client SDK Instructions

DD_MOTION_AREA

struct of motion area

```
struct _dd_motion_area_{
unsigned long           sensitivity;
unsigned short          widthNum;
unsigned short          hightNum;
unsigned char           area [DD_MAX_MOTION_AREA_HIGHT_NUM][DD_MAX_MOTION_AREA_WIDTH_NUM]
}DD_MOTION_AREA;
```

Members

sensitivity
  sensitivity(0-7),high number means high sensitivity
widthNum
  width grid number of area
hightNum
  height grid number of area
area [DD_MAX_MOTION_AREA_HIGHT_NUM] [DD_MAX_MOTION_AREA_WIDTH_NUM]
  grid mask data of area,compatible 1920x1080,each size is 8X8

Client SDK Instructions

DD_MOTION_CONFIG

struct of motion configuration

```
struct _dd_motion_config_{
unsigned long                iSize;
unsigned char                  enable;
unsigned char                  recv;
unsigned short          holdTime;
DD_MOTION_AREA          area;
}DD_MOTION_CONFIG;
```

Members

iSize
: size of the struct

enable
: whether enable motion detection

recv
: reserved byte

holdTime
: delay time

area
: area setting

Client SDK Instructions

DD_NETWORK_ADVANCE_CONFIG

struct of network advanced configuration

```
struct _dd_network_advance_config_{
unsigned long           iSize;
unsigned char           bMessagePort;
unsigned char           bAlarmPort;
unsigned char           bMultiCastIP;
unsigned char           bMTUByteNum;
unsigned short          httpPort;
unsigned short          datePort;
unsigned short          messagePort;
unsigned short          alarmPort;
unsigned short          maxOnlineUserNum;
unsigned short          OnlineUserNum;
unsigned long           multiCastIP;
unsigned long           mtuByteNum;
}DD_NETWORK_ADVANCE_CONFIG;
```

Members

iSize
size of the struct

bMessagePort
whether support message port(read only)

bAlarmPort
whether support alarm port(read only)

bMultiCastIP
whether support multicast address(read only)

bMTUByteNum
whether support MTU byte number(read only)

httpPort
HTTP port

datePort
data port

messagePort
message command port

alarmPort
  alarm port
maxOnlineUserNum
  supportable maximum online user number(read only)
  OnlineUserNum
  number of online users
multiCastIP
  multicast address
mtuByteNum
  MTU byte number

Client SDK Instructions

DD_NETWORK_IP_CONFIG

struct of network IP configuration

```
struct _dd_network_ip_config_{
unsigned long     iSize;
unsigned long       useDHCP;
unsigned long       IP;
unsigned long       subnetMask;
unsigned long       gateway;
unsigned long       preferredDNS;
unsigned long       alternateDNS;
unsigned long       usePPPoE;
char              account[DD_MAX_PPPOE_ACCOUNT_BUF_LEN];
char              password[DD_MAX_PASSWORD_BUF_LEN];
}DD_NETWORK_IP_CONFIG;
```

Members

iSize
  size of the struct

useDHCP
  whether enable DHCP

IP
  network IP

subnetMask
  subnet mask

gateway
  gateway

preferredDNS
  host DNS

alternateDNS
  alternate DNS

usePPPoE
  whether enable PPPoE

account[DD_MAX_PPPOE_ACCOUNT_BUF_LEN]/i>
  PPPoE account

password[DD_MAX_PASSWORD_BUF_LEN]
  PPPoE password

Client SDK Instructions

DD_POSITION

struct of position

```
struct  _dd_position_{
unsigned short x;
unsigned short y;
}DD_POSITION;
```

Members

x
: abscissa

y
: ordinate

Client SDK Instructions

DD_PTZ_CONFIG

struct of PTZ configuration

```
struct _dd_ptz_config_{
unsigned long              iSize;
unsigned char              enable;
unsigned char              address;
unsigned char              recv1;
unsigned char              recv2;
unsigned long              protocol;
DD_SERIAL_CONFIG           serial;
}DD_PTZ_CONFIG;
```

Members

iSize
: size of the struct

enable
: whether enable PTZ function

address
: address

recv1
: reserved byte

recv2
: reserved byte

protocol
: protocol,refer to PROTOCOL_TYPE:

Type	Value
PROTOCOL_NULL	0
PROTOCOL_PELCOP	1
PROTOCOL_PELCOD	2
PROTOCOL_LILIN	3
PROTOCOL_MINKING	4

PROTOCOL_NEON	5
PROTOCOL_STAR	6
PROTOCOL_VIDO	7
PROTOCOL_DSCP	8
PROTOCOL_VISCA	9
PROTOCOL_SAMSUNG	10
PROTOCOL_RM110	11
PROTOCOL_HY	12

serial
: serial port

Client SDK Instructions

DD_PTZ_PRESET_CONFIG

struct of PTZ preset configuration

```
struct _dd_ptz_preset_config_{
unsigned long              iSize;
unsigned char              enablePreset [DD_MAX_PRESET_NUM];
}DD_PTZ_PRESET_CONFIG;
```

Members

iSize
  size of the struct
enablePreset [DD_MAX_PRESET_NUM]
  whether enable preset point

Client SDK Instructions

DD_PTZ_PROTOCOL_INFO

struct of PTZ protocol information

```
struct _dd_ptz_protocol_info {
unsigned long                  protocolID;
unsigned long                  pportproperty;
char                           ProtocolName[64];
}DD_PTZ_PROTOCOL_INFO;
```

Members

protocolID
: protocol type ID

pportproperty
: other attribute's MASK except ID,baud rate, such as whether support some special attribute track etc.

ProtocolName[64]
: protocal name

Client SDK Instructions

DD_RECORD_CONFIG

struct of record configuration

```
struct _dd_record_config_{
unsigned long        iSize;
unsigned char        bOnlyVideo;
unsigned char        bWithAudio;
unsigned char        bindAudioChannel;
unsigned char        bRedundancy;
unsigned short       preAlarmTime;
unsigned short       postAlarmTime;
unsigned short       expired;
unsigned short       recv;
}DD_RECORD_CONFIG;
```

Members

iSize
: size of the struct

bOnlyVideo
: transcript video(only video)

bWithAudio
: transcript audio(based on transcript video)

bindAudioChannel
: corresponding audio channel(may different from video channel number)

bRedundancy
: whether redundant record

preAlarmTime
: time of record before alarm

postAlarmTime
: time of record after alarm

expired
: record expiry time

recv
: reserved byte

Client SDK Instructions

DD_RECORD_CONFIG_MASK

struct of record configuration mask

```
struct  _dd_record_config_mask_{
unsigned long               iSize;
unsigned char               bindAudioChannel;
unsigned char               bRedundancy;
unsigned char               recv1;
unsigned char               recv2;
unsigned short              minPreAlarmTime;
unsigned short              maxPreAlarmTime;
unsigned short              minPostAlarmTime;
unsigned short              maxPostAlarmTime;
unsigned short              minExpired;
unsigned short              maxExpired;
}DD_RECORD_CONFIG_MASK;
```

Members

iSize
size of the struct

bindAudioChannel
whether support to bind audio and video channel

bRedundancy
whether support redundant record

recv1
reserved byte

recv2
reserved byte

minPreAlarmTime
the minimum time of record before alarm

maxPreAlarmTime
the maximum time of record before alarm

minPostAlarmTime
the minimum time of record after alarm

maxPostAlarmTime
the maximum time of record after alarm

minExpired
  the minimum record data expiry time
maxExpired
  the maximum record data expiry time

Client SDK Instructions

DD_RECORD_LOG

struct of record log information

```
struct_dd_record_log_{
unsigned char  bLocked;
unsigned char  bUnofficial;
unsigned char  enableCard;
unsigned char  recv1;
unsigned short diskIndex;
unsigned short fileIndex;
unsigned short logIndex;
unsigned short recv2;
unsigned short deviceID;
unsigned short cameraID;
unsigned long  channel;
unsigned long  type;
unsigned long  size;
DD_TIME                 startTime;
DD_TIME                endTime;
char           cardNo[32];
}DD_RECORD_LOG, *LP_DD_RECORD_LOG;
```

Members

bLocked
    0 means unlocked,1 means locked
bUnofficial
    0 means official record,1 means unofficial record(overlap record after modifying time)
enableCard
    whether enable card
recv1
    reserved byte
diskIndex
    disk number
fileIndex
    file index
logIndex

log index

recv2
: reserved byte

deviceID
: device ID

cameraID
: camera ID

channel
: virtual channel number

type
: record type

size
: size of the record data

startTime
: start time

endTime
: end time

cardNo[32]
: card number

Client SDK Instructions

DD_RELAY_CONFIG

struct of delay configuration

```
struct _dd_relay_config_{
unsigned char           enable;
unsigned char           recv;
unsigned short          holdTime;
char                    name [DD_MAX_NAME_BUF_LEN];
}DD_RELAY_CONFIG;
```

Members

enable
: alarm output device enable switch

recv
: reserved byte

holdTime
: delay time

name [DD_MAX_NAME_BUF_LEN]
: device name

Client SDK Instructions

DD_SENSOR_CONFIG

struct of sensor configuration

```
struct _dd_sensor_config_{
unsigned long                iSize;
unsigned char                  enable;
unsigned char                  bNO;
unsigned short         holdTime;
char                         name [DD_MAX_NAME_BUF_LEN];
}DD_SENSOR_CONFIG;
```

Members

iSize
  size of the struct
enable
  whether enable detection
bNO
  device type:normal open or normal close
holdTime
  delay time
name [DD_MAX_NAME_BUF_LEN]
  device name

Client SDK Instructions

DD_SERIAL_CONFIG

struct of serial configuration

```
struct _dd_serial_config_{
unsigned long             baudRate;
unsigned long             dataBit;
unsigned long             stopBit;
unsigned long             parity;
unsigned long             dataFlowControl;
}DD_SERIAL_CONFIG;
```

Members

baudRate

baud rate,refer to the list below:

Type	Value
SBR_110	0
SBR_300	1
SBR_600	2
SBR_1200	3
SBR_2400	4
SBR_4800	5
SBR_9600	6
SBR_19200	7
SBR_38400	8
SBR_57600	9
SBR_115200	10
SBR_230400	11
SBR_460800	12
SBR_921600	13

dataBit
data bit,refer to the list below:

Type	Value
DATABITS7	7
DATABITS8	8

stopBit
stop bit,refer to the list below:

Type	Value
STOPBITS1	2
STOPBITSONEHALF	3
STOPBITS2	4

parity
parity check bit,refer to the list below:

Type	Value	Description
PARITYEVEN	'E'	even parity check
PARITYODD	'O'	odd parity check
PARITYMARK	'M'	mark parity check
PARITYSPACE	'S'	space parity check
PARITYNONE	'N'	no parity check

dataFlowControl
data stream control

Client SDK Instructions

DD_SMTP_CONFIG

struct of SMTP configuration

```
struct _dd_smtp_config_{
unsigned long              iSize;
unsigned short             port;
unsigned short             enableSSL;
char                       server [DD_MAX_URL_BUF_LEN];
char                       sendAddress [DD_MAX_URL_BUF_LEN];
char                       password [DD_MAX_PASSWORD_BUF_LEN];
unsigned long              enableRecvAddrNum;           £©
char                       receiveAddress [DD_MAX_EMAIL_RECEIVE_ADDR_NUM][DD_MAX_URL_BUF_LEN]
}DD_SMTP_CONFIG;
```

Members

iSize
  size of the struct
port
  SMTP server port
enableSSL
  whether enable SSL check
server [DD_MAX_URL_BUF_LEN]
  send server address
sendAddress [DD_MAX_URL_BUF_LEN]
  send SMTP address
password [DD_MAX_PASSWORD_BUF_LEN]
  password
enableRecvAddrNum
  available address number for receiving(read only)
receiveAddress [DD_MAX_EMAIL_RECEIVE_ADDR_NUM][DD_MAX_URL_BUF_LEN]
  list of acception address

Client SDK Instructions

DD_TIME

struct of system time setting information.

```
struct _dd_time_{
unsigned char          second;
unsigned char          minute;
unsigned char          hour;
unsigned char          wday;
unsigned char          mday;
unsigned char          month;
unsigned short year;
}DD_TIME, *LP_DD_TIME;
```

Members

second
: seconds after minute,range (0-59).

minute
: minutes after hour,range (0-59).

hour
: hours since midnight,range (0-23).

wday
: day of week ,range(0-6; Sunday=0).

mday
: day of month,range (1-31).

month
: month range(0-11; January=0).

year
: year (current year minus 1900).

Client SDK Instructions

DD_TRIGGER_ALARM_OUT

struct of triggering alarm

```
struct _dd_trigger_alarm_out_{
unsigned char              toBuzzer;
unsigned char              ShowFullScreen;
unsigned char              sendEmail;
unsigned char              toUploadToAlarmCentre;
unsigned long              toAlarmOut;
}DD_TRIGGER_ALARM_OUT;
```

Members

toBuzzer
trigger buzzer alarm

ShowFullScreen
trigger full screen alarm(no trigger when channel number is 0xff)

sendEmail
send email

toUploadToAlarmCentre
upload to alarm center

toAlarmOut
alarm output(bit matches output device)

Client SDK Instructions

DD_TRIGGER_PTZ

struct of triggering PTZ

```
struct _dd_trigger_ptz_{
unsigned char                toPTZType;
unsigned char                toIndex;
unsigned char                backIndex;
unsigned char                recv;
}DD_TRIGGER_PTZ;
```

Members

toPTZType
linkage type£¬refer to the list below:

Type	Value
DD_PTZ_TYPE_PRESET	1
DD_PTZ_TYPE_CRUISE	2
DD_PTZ_TYPE_TRACE	3

toIndex
linkage number(preset point, cruise line, track)

backIndex
linkage returned number(preset point, cruise line, track)

recv
reserved byte

Client SDK Instructions

DD_TRIGGER_RECORD

struct of triggering video record

```
struct _dd_trigger_record_{
unsigned char                snapCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
unsigned char                recordCH [DD_MAX_CAMERA_NUM_BYTE_LEN];
}DD_TRIGGER_RECORD;
```

Members

snapCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  trigger capture
recordCH [DD_MAX_CAMERA_NUM_BYTE_LEN]
  trigger record

Client SDK Instructions

DD_VIDEO_COLOR

struct of video color

```
struct _dd_video_color_{
unsigned long      startTime;
unsigned char      brightness;
unsigned char      hue;
unsigned char      saturation;
unsigned char      contrast;
}DD_VIDEO_COLOR;
```

Members

startTime
: start time of the color(relative time in one day)

brightness
: brightness ,range 0-255

hue
: hue ,range 0-255

saturation
: saturation ,range 0-255

contrast
: contrast ,range 0-255

Client SDK Instructions

DD_VIDEO_COLOR_CONFIG

struct of video color configuration

```
struct  _dd_video_color_config_{
unsigned long          iSize;
unsigned long          usedNum;
DD_VIDEO_COLOR  videoColor[DD_MAX_COLOR_CFG_NUM];
}DD_VIDEO_COLOR_CONFIG;
```

Members

iSize
    size of the struct
usedNum
    scheme number in use, default value is 1
videoColor[DD_MAX_COLOR_CFG_NUM]
    color scheme

Client SDK Instructions

DD_VIDEO_OSD_CONFIG

struct of video OSD configuration

```
 struct _dd_video_osd_config_{
 unsigned long iSize;
 unsigned char enableCameraName;
 unsigned char enableTimeStamp;
 unsigned char enableTimeStampWithWeek;
 unsigned char enableDefineText;
 DD_POSITION cameraName;
 DD_POSITION timeStamp;
 DD_POSITION defineText;
 char text [DD_MAX_TEXT_BUF_LEN];
 struct
        {
          unsigned long enable;
          DD_AREA                area;
        }cover[DD_MAX_VIDEO_COVER_NUM];
}DD_VIDEO_OSD_CONFIG;
```

Members

iSize
size of the struct
enableCameraName
overlap channel name
enableTimeStamp
overlap timestamp
enableTimeStampWithWeek
timestamp with week
enableDefineText
overlap self-defined text
cameraName
channel name position,range from (0,0) to (100,100)
timeStamp
timestamp position,range from (0,0) to (100,100)
defineText
self-defined text position,range from (0,0) to (100,100)
text [DD_MAX_TEXT_BUF_LEN]

self-defined text

enable

area overlap is valid or not

area

area parameter

cover[DD_MAX_VIDEO_COVER_NUM]

covered area parameter

Client SDK Instructions

DD_WEEK_SCHEDULE

struct of week schedule

```
struct _dd_week_schedule_{
DD_DATE_SCHEDULE   week[7];
}DD_WEEK_SCHEDULE;
```

Members

week[7]
week schedule structure,7 stands for each day's schedule in a week of 7 days

Client SDK Instructions

DEC_ADVANCE_NETWORK

struct of decoder advanced network configuration

```
struct _dec_advance_network{
unsigned long                iSize;
unsigned char               TimeZone;
unsigned char               hour;
unsigned char               min;
unsigned char               sec;
unsigned char               mday;
unsigned char               month;
unsigned short              year;
bool                         enableFlag;
char                         name [132];
int                          NTP_Port;
int                          syncInterval;
}DEC_ADVANCE_NETWORK;
```

Members

iSize
size of the struct
TimeZone
time zone, refer to DD_TIME_ZONE_NAME:

Type	Value
DD_TIME_ZONE_GMT_D12	0
DD_TIME_ZONE_GMT_D11	1
DD_TIME_ZONE_GMT_D10	2
DD_TIME_ZONE_GMT_D9	3
DD_TIME_ZONE_GMT_D8	4
DD_TIME_ZONE_GMT_D7	5
DD_TIME_ZONE_GMT_D6	6
DD_TIME_ZONE_GMT_D5	7

DD_TIME_ZONE_GMT_D4_30	8
DD_TIME_ZONE_GMT_D4	9
DD_TIME_ZONE_GMT_D3_30	10
DD_TIME_ZONE_GMT_D3	11
DD_TIME_ZONE_GMT_D2	12
DD_TIME_ZONE_GMT_D1	13
DD_TIME_ZONE_GMT	14
DD_TIME_ZONE_GMT_A1	15
DD_TIME_ZONE_GMT_A2	16
DD_TIME_ZONE_GMT_A3	17
DD_TIME_ZONE_GMT_A3_30	18
DD_TIME_ZONE_GMT_A4	19
DD_TIME_ZONE_GMT_A4_30	20
DD_TIME_ZONE_GMT_A5	21
DD_TIME_ZONE_GMT_A5_30	22
DD_TIME_ZONE_GMT_A5_45	23
DD_TIME_ZONE_GMT_A6	24
DD_TIME_ZONE_GMT_A6_30	25
DD_TIME_ZONE_GMT_A7	26
DD_TIME_ZONE_GMT_A8	27
DD_TIME_ZONE_GMT_A9	28
DD_TIME_ZONE_GMT_A9_30	29
DD_TIME_ZONE_GMT_A10	30
DD_TIME_ZONE_GMT_A11	31
DD_TIME_ZONE_GMT_A12	32
DD_TIME_ZONE_GMT_A13	33

hour
    hour

min
    minute

sec

second

mday

which day in a week

month

which month in a year

year

a particular year,2008-2025

enableFlag

NTP enable flag

name [132]

NTP server address

NTP_Port

NTP server port

syncInterval

synchronous time interval,in hour

Client SDK Instructions

DEC_DATE_SCHEDULE

struct of decoder date schedule

```
struct _dec_date_schedule_{
unsigned long long            hour [24];
}DEC_DATE_SCHEDULE;
```

Members

hour [24]
: time format,24 means in 24-hours time system, each bit of unsigned long long stands for each minute's status

Client SDK Instructions

DEC_DEVICE_CONFIG

struct of decoder configuration

```
struct  _dec_device_config{
unsigned long                   iSize;
char                             deviceName [DEC_MAX_NAME_LEN];
unsigned long                   channelNum;
unsigned long                   productID;
unsigned long                   productSubID;
unsigned long                   softVersion;
char                             mcuVersion [DEC_MAX_VERSION_BUF_LEN];
char                             kernelVersion [DEC_MAX_VERSION_BUF_LEN];
char                             hardwareVersion [DEC_MAX_VERSION_BUF_LEN];
}DEC_DEVICE_CONFIG;
```

Members

iSize
  size of the struct
deviceName [DEC_MAX_NAME_LEN]
  device name(notice double byte character)
channelNum
  sum of decoder channels
productID
  product ID
productSubID
  product sub ID
softVersion
  soft version
mcuVersion [DEC_MAX_VERSION_BUF_LEN]
  MCU version
kernelVersion [DEC_MAX_VERSION_BUF_LEN]
  kernel version
hardwareVersion [DEC_MAX_VERSION_BUF_LEN]
  hardware version

Client SDK Instructions

DEC_NETWORK_CONFIG

struct of decoder network configuration

```
struct _dec_network_config{
unsigned long                  iSize;
unsigned long                  IP;
unsigned long                  subnetMask;
unsigned long                  gateway;
unsigned short                httpPort;
unsigned short                decoderPort;
char                          MAC [8];
unsigned long                 multiCastIP;
unsigned long                 bDHCP;
unsigned long                 dns1;
unsigned long                 dns2;
}DEC_NETWORK_CONFIG;
```

Members

iSize
  size of the struct
IP
  network address
subnetMask
  sub net mask
gateway
  gateway
httpPort
  HTTP port
decoderPort
  decoder port
MAC [8]
  binded MAC
multiCastIP
  multicast address
dns1
  DNS1
dns2
  DNS2

Client SDK Instructions

DEC_OTHER_ALARM

other alarm struct of decoder

```
struct _dec_other_alarm_{
unsigned long        iSize;
unsigned long        toBuzzerForIPConflict;
unsigned long        toAlarmOutForDisconnect;
unsigned long        toBuzzerForDisconnect;
unsigned long        toAlarmOutForIPConflict;
}DEC_OTHER_ALARM;
```

Members

iSize
size of the struct

toBuzzerForIPConflict
IP conflict triggers buzzer

toAlarmOutForDisconnect
IP conflict alarm output(bit matches output device)

toBuzzerForDisconnect
network disconnection triggers buzzer

toAlarmOutForIPConflict
network disconnection alarm output(bit matches output device)

Client SDK Instructions

DEC_SENSOR_SETUP

struct of decoder sensor setting

```
struct _dec_sensor_setup_{
unsigned long                iSize;
unsigned char                  enable;
unsigned char                  bNO;
unsigned short        holdTime;
char                         name [DEC_MAX_BUF_LEN];
unsigned long                toBuzzer;
unsigned long                toAlarmOut;
}DEC_SENSOR_SETUP;
```

Members

iSize
  size of the struct
enable
  whether enable detection
bNO
  device type:normal open or normal close
holdTime
  delay time
name [DEC_MAX_BUF_LEN]
  device name
toBuzzer
  trigger buzzer alarm
toAlarmOut
  alarm output(bit matches output device)

Client SDK Instructions

DEC_WEEK_SCHEDULE

struct of decoder week schedule

```
struct _dec_week_schedule_{
DEC_DATE_SCHEDULE        week[7];
}DEC_WEEK_SCHEDULE;
```

Members

week[7]
: week schedule structure,7 stands for 7 days' schedule in a week

Client SDK Instructions

NET_SDK_ALARMINFO

struct of alarm information

```
struct _net_sdk_alarminfo{
DWORD          dwAlarmType;
DWORD          dwSensorIn;
DWORD          dwChannel;
DWORD          dwDisk;
}NET_SDK_ALARMINFO;
```

Members

dwAlarmType

alarm type,refer to the following list:

Type	Description
NET_SDK_N9000_ALARM_TYPE_MOTION	Motion detection alarm
NET_SDK_N9000_ALARM_TYPE_SENSOR	Sensor alarm input
NET_SDK_N9000_ALARM_TYPE_VLOSS	Video loss alarm
NET_SDK_N9000_ALARM_TYPE_FRONT_OFFLINE	Front-end device offline ala
NET_SDK_N9000_ALARM_TYPE_OSC	Object Abandoned/Missing ala
NET_SDK_N9000_ALARM_TYPE_AVD	Exception alarm
NET_SDK_N9000_ALARM_TYPE_AVD_SECENE	Exception detection-Scene change, for IPC only
NET_SDK_N9000_ALARM_TYPE_AVD_CLARITY	Exception detection-video blurred, for IPC onl
NET_SDK_N9000_ALARM_TYPE_AVD_COLOR	Exception detection-video color cast, for IPC c
NET_SDK_N9000_ALARM_TYPE_PEA_TRIPWIRE	Tripwire alarm
NET_SDK_N9000_ALARM_TYPE_PEA_PERIMETER	Intrusion alarm
NET_SDK_N9000_ALARM_TYPE_VFD	Face Detection (currently I
NET_SDK_N9000_ALARM_TYPE_CDD	Crowd density detection
NET_SDK_N9000_ALARM_TYPE_IPD	Intrusion person detection
NET_SDK_N9000_ALARM_TYPE_CPC	People counting
NET_SDK_N9000_ALARM_TYPE_FACE_MATCH	Face match alarm(NVR)
NET_SDK_N9000_ALARM_TYPE_FACE_MATCH_FOR_IPC	Face match alarm(IPC)
NET_SDK_N9000_ALARM_TYPE_TRAJECT	Target tracking trajectory
NET_SDK_N9000_ALARM_TYPE_VEHICE	license plate(IPC)
NET_SDK_N9000_ALARM_TYPE_AOIENTRY	Enter the area(IPC)
NET_SDK_N9000_ALARM_TYPE_AOILEAVE	Leave the area(IPC)
NET_SDK_N9000_ALARM_TYPE_PASSLINE	Tripwire counting
NET_SDK_N9000_ALARM_TYPE_IP_CONFLICT	IP address conflict
NET_SDK_N9000_ALARM_TYPE_DISK_IO_ERROR	Disk IO error
NET_SDK_N9000_ALARM_TYPE_DISK_FULL	Full disk
NET_SDK_N9000_ALARM_TYPE_RAID_SUBHEALTH	Array sub-health
NET_SDK_N9000_ALARM_TYPE_RAID_UNAVAILABLE	Array unavailable
NET_SDK_N9000_ALARM_TYPE_ILLEIGAL_ACCESS	Illegal access
NET_SDK_N9000_ALARM_TYPE_NET_DISCONNECT	Network disconnect
NET_SDK_N9000_ALARM_TYPE_NO_DISK	No disk in disk group
NET_SDK_N9000_ALARM_TYPE_SIGNAL_SHELTER	Signal shelter
NET_SDK_N9000_ALARM_TYPE_HDD_PULL_OUT	Front panel HDD pull out

the alarmtype NET_SDK_ALARM_TYPE(refer to the following list) is not be used now,it is invalid

Type	Description
NET_SDK_ALARM_TYPE_MOTION	motion detection
NET_SDK_ALARM_TYPE_SENSOR	sensor alarm
NET_SDK_ALARM_TYPE_VLOSS	single loss
NET_SDK_ALARM_TYPE_SHELTER	shelter alarm
NET_SDK_ALARM_TYPE_DISK_FULL	full harddisk
NET_SDK_ALARM_TYPE_DISK_UNFORMATTED	disc unformatted
NET_SDK_ALARM_TYPE_DISK_WRITE_FAIL	harddisk read-write error
NET_SDK_ALARM_TYPE_EXCEPTION	exception alarm

dwSensorIn
sensor alarm input port number
dwChannel
when alarm is relative to channel,dwChannel means alarm channel
dwDisk
in disc alarming, it means disc number which alarms

Client SDK Instructions

NET_SDK_ALARMINFO_EX

struct of alarm information

```
struct _net_sdk_alarminfo_ex{
DWORD          dwAlarmType;
DWORD          dwSensorIn;
DWORD          dwChannel;
DWORD          dwDisk;
char                        sensorName[36];
char                                 alarmTime[20];
char           resv[128];
}NET_SDK_ALARMINFO_EX;
```

Members

dwAlarmType

alarm type,refer to the following list:

Type	Description
NET_SDK_N9000_ALARM_TYPE_MOTION	Motion detection alarm
NET_SDK_N9000_ALARM_TYPE_SENSOR	Sensor alarm input
NET_SDK_N9000_ALARM_TYPE_VLOSS	Video loss alarm
NET_SDK_N9000_ALARM_TYPE_FRONT_OFFLINE	Front-end device offline ala
NET_SDK_N9000_ALARM_TYPE_OSC	Object Abandoned/Missing ala
NET_SDK_N9000_ALARM_TYPE_AVD	Exception alarm
NET_SDK_N9000_ALARM_TYPE_AVD_SECENE	Exception detection- Scene change, for IPC only
NET_SDK_N9000_ALARM_TYPE_AVD_CLARITY	Exception detection- video blurred, for IPC onl
NET_SDK_N9000_ALARM_TYPE_AVD_COLOR	Exception detection- video color cast, for IPC
NET_SDK_N9000_ALARM_TYPE_PEA_TRIPWIRE	Tripwire alarm
NET_SDK_N9000_ALARM_TYPE_PEA_PERIMETER	Intrusion alarm
NET_SDK_N9000_ALARM_TYPE_VFD	Face Detection (currently I
NET_SDK_N9000_ALARM_TYPE_CDD	Crowd density detection
NET_SDK_N9000_ALARM_TYPE_IPD	Intrusion person detection
NET_SDK_N9000_ALARM_TYPE_CPC	People counting
NET_SDK_N9000_ALARM_TYPE_FACE_MATCH	Face match alarm(NVR)
NET_SDK_N9000_ALARM_TYPE_FACE_MATCH_FOR_IPC	Face match alarm(IPC)
NET_SDK_N9000_ALARM_TYPE_TRAJECT	Target tracking trajectory
NET_SDK_N9000_ALARM_TYPE_VEHICE	license plate(IPC)
NET_SDK_N9000_ALARM_TYPE_AOIENTRY	Enter the area(IPC)
NET_SDK_N9000_ALARM_TYPE_AOILEAVE	Leave the area(IPC)
NET_SDK_N9000_ALARM_TYPE_PASSLINE	Tripwire counting
NET_SDK_N9000_ALARM_TYPE_IP_CONFLICT	IP address conflict
NET_SDK_N9000_ALARM_TYPE_DISK_IO_ERROR	Disk IO error
NET_SDK_N9000_ALARM_TYPE_DISK_FULL	Full disk
NET_SDK_N9000_ALARM_TYPE_RAID_SUBHEALTH	Array sub-health
NET_SDK_N9000_ALARM_TYPE_RAID_UNAVAILABLE	Array unavailable
NET_SDK_N9000_ALARM_TYPE_ILLEIGAL_ACCESS	Illegal access
NET_SDK_N9000_ALARM_TYPE_NET_DISCONNECT	Network disconnect
NET_SDK_N9000_ALARM_TYPE_NO_DISK	No disk in disk group

NET_SDK_N9000_ALARM_TYPE_SIGNAL_SHELTER	Signal shelter
NET_SDK_N9000_ALARM_TYPE_HDD_PULL_OUT	Front panel HDD pull out

the alarmtype NET_SDK_ALARM_TYPE(refer to the following list) is not be used now,it is invalid

Type	Description
NET_SDK_ALARM_TYPE_MOTION	motion detection
NET_SDK_ALARM_TYPE_SENSOR	sensor alarm
NET_SDK_ALARM_TYPE_VLOSS	single loss
NET_SDK_ALARM_TYPE_SHELTER	shelter alarm
NET_SDK_ALARM_TYPE_DISK_FULL	full harddisk
NET_SDK_ALARM_TYPE_DISK_UNFORMATTED	disc unformatted
NET_SDK_ALARM_TYPE_DISK_WRITE_FAIL	harddisk read-write error
NET_SDK_ALARM_TYPE_EXCEPTION	exception alarm

dwSensorIn
sensor alarm input port number
dwChannel
when alarm is relative to channel,dwChannel means alarm channel
dwDisk
in disc alarming, it means disc number which alarms
sensorName
in sensor alarming, it means the alarming sensor's name
alarmTime
alarm time
resv
preserve

Client SDK Instructions

NET_SDK_CLIENTINFO

struct of log information

```
struct _net_sdk_clientinfo{
LONG lChannel;
LONG streamType;
HWND hPlayWnd;
int bNoDecode;
}NET_SDK_CLIENTINFO, *LPNET_SDK_CLIENTINFO;
```

Members

lChannel
: channel number,start from 0

streamType
: data stream type,two types:NET_SDK_MAIN_STREAM and NET_SDK_SUB_STREAM

hPlayWnd
: play window handle

bNoDecode
: 0:decode,1:not decode.only for windows os,default value is 0

Client SDK Instructions

NET_SDK_DEVICE_DISCOVERY_INFO

discover device automatically on LAN

```
struct  _net_sdk_device_discovery_info{
unsigned long          deviceType;
char                           productType[16];
char                           strIP[16];
char                           strNetMask[16];
char                            strGateWay[16];
unsigned char  byMac[8];
unsigned short         netPort;
unsigned short         httpPort;
unsigned long          softVer;
unsigned long          softBuildDate;
}NET_SDK_DEVICE_DISCOVERY_INFO;
```

Members

deviceType
device type,refer to the follow:

Type	Description
NET_SDK_DVR	Digital video record
NET_SDK_DVS	Network Video Server
NET_SDK_IPCAMERA	IP camera
NET_SDK_SUPERDVR	board card
NET_SDK_DECODER	decoder

productType[16]
product type
strIP[16]
IP
strNetMask[16]
subnet mask
strGateWay[16]

gateway

byMac[8]
: MAC address

netPort
: network port

httpPort
: http port

softVer
: software version

softBuildDate
: software build date

Client SDK Instructions

NET_SDK_DEVICEINFO

structure of device login

```
struct _net_sdk_deviceinfo{
 unsigned char  localVideoInputNum;
 unsigned char  audioInputNum;
 unsigned char  sensorInputNum;
 unsigned char  sensorOutputNum;
 unsigned long    displayResolutionMask;
 unsigned char  videoOuputNum;
 unsigned char  netVideoOutputNum;
 unsigned char  netVideoInputNum;
 unsigned char  IVSNum;
 unsigned char  presetNumOneCH;
 unsigned char  cruiseNumOneCH;
 unsigned char  presetNumOneCruise;
 unsigned char  trackNumOneCH;
 unsigned char  userNum;
 unsigned char  netClientNum;
 unsigned char  netFirstStreamNum;
 unsigned char  deviceType;
 unsigned char  doblueStream;
 unsigned char  audioStream;
 unsigned char  talkAudio;
 unsigned char  bPasswordCheck;
 unsigned char  defBrightness;
 unsigned char  defContrast;
 unsigned char  defSaturation;
 unsigned char  defHue;
 unsigned short videoInputNum;
 unsigned short   deviceID;
 unsigned long    videoFormat;
 unsigned long  function[8];
 unsigned long  deviceIP;
 unsigned char  deviceMAC[8];
 unsigned long  buildDate;
 unsigned long  buildTime;
 char           deviceName[36];
 char            firmwareVersion[36];
 char            kernelVersion[64];
 char            hardwareVersion[36];
 char            MCUVersion[36];
 char   firmwareVersionEx[100];                 //firmware version extension for new product
 char   deviceProduct[28];                      //Device model
 }NET_SDK_DEVICEINFO, *LPNET_SDK_DEVICEINFO;
```

Members

localVideoInputNum
  number of local video input channel
audioInputNum
  number of audio input channel
sensorInputNum
  number of sensor input channel
sensorOutputNum
  number of relay output
displayResolutionMask
  monitor optional resolution
videoOuputNum
  number of video output(supportable maximum playback channel number)
netVideoOutputNum
  maximum channel number of network playback

netVideoInputNum
 channel number of digital single input
IVSNum
 number of smart analytics channel
presetNumOneCH
 number of preset point in each channel
cruiseNumOneCH
 number of cruise line in each channel
presetNumOneCruise
 number of preset point in each cruise line
trackNumOneCH
 track number in each channel
userNum
 user number
netClientNum
 maximum client number
netFirstStreamNum
 channel maximum number of main code stream transmission,namely how many clients check main code stream meanwhile
deviceType
 device type
doblueStream
 whether provide dual stream
audioStream
 whether provide audio stream
talkAudio
 whether enable talkback function:1 means yes;0 means no
bPasswordCheck
 whether need to input password
defBrightness
 default brightness
defContrast
 default contrast
defSaturation
 default saturation
defHue
 default hue
videoInputNum
 channel number of video input(add network channel number if local video input)
deviceID
 device ID
videoFormat
 video format,refer to the following list:

Type	Value
DD_VIDEO_FORMAT_NTSC	0x01
DD_VIDEO_FORMAT_PAL	0x02

function[8]
 function description
deviceIP

device network address
deviceMAC[8]
device MAC
buildDate
building date:year<<16 + month<<8 + mday
buildTime
building time:hour<<16 + min<<8 + sec
deviceName[36]
device name
firmwareVersion[36]
firmware version
kernelVersion[64]
kernel version
hardwareVersion[36]
hardware version
MCUVersion[36]
MCU version
firmwareVersionEx[100]
Reserved characters
deviceProduct[28]
Device model

Client SDK Instructions

NET_SDK_EVENT

struct of event log

```
struct  _net_sdk_event{
unsigned short   chnn;
unsigned short   type;
DD_TIME        startTime;
DD_TIME        endTime;
}NET_SDK_EVENT,*LPNET_SDK_EVENT;
```

Members

chnn
: event happend in which channel

type
: event type

startTime
: event starting time

endTime
: event ending time

Client SDK Instructions

NET_SDK_FRAME_INFO

struct of data frame information

```
struct   _net_sdk_frame_info{
unsigned long           deviceID;
unsigned long         channel;
unsigned long           frameType;
unsigned long           length;
unsigned long           keyFrame;
unsigned long        width;
unsigned long        height;
unsigned long           frameIndex;
unsigned long           frameAttrib;
unsigned long           streamID;
LONGLONG                time;
LONGLONG                relativeTime;
}NET_SDK_FRAME_INFO;
```

Members

deviceID
device ID
channel
data channel,channel number starts from 0
frameType
data frame type,refer to DD_FRAME_TYPE:

Type	Value
DD_FRAME_TYPE_NONE	0x00
DD_FRAME_TYPE_VIDEO	0x01
DD_FRAME_TYPE_AUDIO	0x02
DD_FRAME_TYPE_TALK_AUDIO	0x03
DD_FRAME_TYPE_JPEG	0x04
DD_FRAME_TYPE_VIDEO_FORMAT	0x05
DD_FRAME_TYPE_AUDIO_FORMAT	0x06
DD_FRAME_TYPE_TALK_AUDIO_FORMAT	0x07

DD_FRAME_TYPE_EVENT	0x08
DD_FRAME_TYPE_TEXT	0x09
DD_FRAME_TYPE_END	

length
data length
keyFrame
keyframe,0:minor frame;1:key frame
width
width of data frame
height
height of data frame
frameIndex
data frame index
frameAttrib
data frame attribute,refer to DD_FRAME_ATTRIB:

Type	Value	Description
DD_PLAY_FRAME_NO_SHOW	0x01	no show the frame
DD_PLAY_FRAME_SHOW	0x02	show the frame
DD_PLAY_FRAME_ALL_END	0x04	data read is finished,no more data
DD_PLAY_FRAME_SEC_END	0x08	the section ends
DD_PLAY_FRAME_NO_TIME_STAMP	0x10	the frame with time stamp, so shield time function when capture
DD_PLAY_FRAME_FF	0x20	fast forward frame

DD_LIVE_FRAME_FIRST_STREAM	0x40	live main code stream frame
DD_LIVE_FRAME_SECOND_STREAM	0x80	live sub code stream frame
DD_LIVE_FRAME_JPEG	0x100	JPEG image frame
DD_LIVE_FRAME_TALK	0x200	talkback audio data frame

streamID
data stream ID

time
absolute time,calculate from 00:00:00 on Jan.1st in 1970,in microsecond,it changes when change device time

relativeTime
relative time,in microsecond,it won't change when change device time,because it is continuous

Client SDK Instructions

NET_SDK_JPEGPARA

struct of JPEG image

```
 struct{
  WORD     wPicSize;
  WORD     wPicQuality;
}NET_DVR_JPEGPARA,*LPNET_DVR_JPEGPARA;
```

Members

wPicSize
: Picture size：0-CIF，1-QCIF，2-D1，3-UXGA，4-SVGA，5-HD720p，6-VGA，7-XVGA，8-HD900p

wPicQuality
: Picture quality level：0-best，1-better，2-ordinary

Client SDK Instructions

NET_SDK_LOG

struct of device log information.

```
struct{
DD_TIME                     strLogTime;
DWORD                        dwMajorType;
DWORD                        dwMinorType;
char                         sNetUser[MAX_NAMELEN];
DWORD                        dwRemoteHostAddr;
char                         sContent[MAX_CONTENTLEN];
}NET_SDK_LOG,*LPNET_SDK_LOG;
```

Members

strLogTime
time of log.

dwMajorType
main type.

dwMinorType
minor type.

sNetUser[MAX_NAMELEN]
network user.

dwRemoteHostAddr
remote host address.

sContent[MAX_CONTENTLEN]
Details.

Client SDK Instructions

NET_SDK_REC_EVENT

struct of record file information according to event.

```
struct _net_sdk_rec_event{
DWORD                      dwChannel;
DD_TIME                   startTime;
DD_TIME                    stopTime;
DWORD                      dwRecType;
}NET_SDK_REC_EVENT;
```

Members

dwChannel
channel number.
startTime
start time.
stopTime
stop time.
dwRecType
record event type,refer to DD_RECORD_TYPE:

Type	Value	Description
DD_RECORD_TYPE_NONE	0x0000	no record type
DD_RECORD_TYPE_MANUAL	0x0001	record manually
DD_RECORD_TYPE_SCHEDULE	0x0002	record at regular time
DD_RECORD_TYPE_MOTION	0x0004	motion detection record
DD_RECORD_TYPE_SENSOR	0x0008	sensor alarm record
DD_RECORD_TYPE_BEHAVIOR	0x0010	behaviour

		analysis alarm record

Client SDK Instructions

NET_SDK_REC_FILE

struct of record file information.

```
struct _net_sdk_rec_file{
DWORD                   dwChannel;
DWORD                   bFileLocked;
DD_TIME                 startTime;
DD_TIME                 stopTime;
DWORD                   dwRecType;
DWORD                   dwPartition;
DWORD                   dwFileIndex;
}NET_SDK_REC_FILE;
```

Members

dwChannel
channel number.

bFileLocked
whether record file is locked.

startTime
start time.

stopTime
stop times.

dwRecType
record event type,refer to DD_RECORD_TYPE:

Type	Value	Description
DD_RECORD_TYPE_NONE	0x0000	no record type
DD_RECORD_TYPE_MANUAL	0x0001	record manually
DD_RECORD_TYPE_SCHEDULE	0x0002	record at regular time
DD_RECORD_TYPE_MOTION	0x0004	motion detection

		record
DD_RECORD_TYPE_SENSOR	0x0008	sensor alarm record
DD_RECORD_TYPE_BEHAVIOR	0x0010	behaviour analysis alarm record

dwPartition
record file partition.

dwFileIndex
index of record filename.

Client SDK Instructions

NET_SDK_REC_TIME

struct of record file information by time.

```
struct _net_sdk_rec_time{
DWORD                         dwChannel;
DD_TIME                        startTime;
DD_TIME                        stopTime;
}NET_SDK_REC_TIME;
```

Members

dwChannel
: channel number.

startTime
: the start time of videotape.

stopTime
: the stop time of videotape.

Client SDK Instructions

NET_SDK_RECORD_STATUS

struct of record status

```
struct _net_sdk_record_status{
DWORD           dwRecordType;
DWORD           dwChannel;
}NET_SDK_RECORD_STATUS;
```

Members

dwRecordType

record event type,refer to DD_RECORD_TYPE:

Type	Value	Description
DD_RECORD_TYPE_NONE	0x0000	no record type
DD_RECORD_TYPE_MANUAL	0x0001	manual record
DD_RECORD_TYPE_SCHEDULE	0x0002	regular record
DD_RECORD_TYPE_MOTION	0x0004	motion detection record
DD_RECORD_TYPE_SENSOR	0x0008	sensor alarm record
DD_RECORD_TYPE_BEHAVIOR	0x0010	behaviour analysis alarm record

dwChannel

record channel

Client SDK Instructions

NET_SDK_RECORD_STATUS_EX

struct of record status

```
struct _net_sdk_record_status_ex{
DWORD          dwRecordType;
DWORD          dwChannel;
DWORD          dwRecordStatus;
}NET_SDK_RECORD_STATUS_EX;
```

Members

dwRecordType
record event type,refer to DD_RECORD_TYPE:

Type	Value	Description
DD_RECORD_TYPE_NONE	0x0000	no record type
DD_RECORD_TYPE_MANUAL	0x0001	manual record
DD_RECORD_TYPE_SCHEDULE	0x0002	regular record
DD_RECORD_TYPE_MOTION	0x0004	motion detection record
DD_RECORD_TYPE_SENSOR	0x0008	sensor alarm record
DD_RECORD_TYPE_BEHAVIOR	0x0010	behaviour analysis alarm record

dwChannel
record channel
dwRecordStatus

0 stoped,1 recording,2 abnormal

Client SDK Instructions

PTZ_3D_POINT_INFO

information about PTZ 3D control

```
struct PTZ_3D_POINT_INFO{
int       selBeginX;
int       selBeginY;
int       selEndX;
int       selEndY;
int       displayWidth;
int       displayHeight;
int       reserve[2];
}PTZ_3D_POINT_INFO;
```

Members

selBeginX
X coordinates of the starting point

selBeginY
Y coordinates of the starting point

selEndX
End point X coordinates

selEndY
End point Y coordinates

displayWidth
Image width

displayHeight
Image height

reserve[2]
Retention value,not enable

Remarks

4 coordinate variable is the mouse position relative to the current window of the upper left corner of the screen. enlarge:selBeginX < selEndX, narrow:selBeginX > selEndX

Client SDK Instructions

NET_SDK_IPC_DEVICE_INFO

struct of IPC in device management

```
_net_sdk_ipc_device_info_{
unsigned long   deviceID;
unsigned short  channel;
unsigned short  status;
char            szEtherName[16];
char            szServer[64];
unsigned short  nPort;
unsigned short  nHttpPort;
unsigned short  nCtrlPort;
char            szID[64];
char            username[36];
unsigned long   manufacturerId;
char            manufacturerName[36];
char            productModel[36];
unsigned char   bUseDefaultCfg;
unsigned char   bPOEDevice;
unsigned char   resv[2];
 }NET_SDK_IPC_DEVICE_INFO;
```

Members

deviceID
  device ID(reserved)

channel
  Channel number of IPC(Start from 0)

status
  Connection status(1 means online,0 means offline)

szEtherName[16]
  If it is null,default is eth0

szServer[64]
  IP address of IPC

nPort
  Port of IPC

nHttpPort
  http port

nCtrlPort
  Control ports, generally the same as the nPort
szID[64]
  Device identification (or MAC address)
username[36]
  username
manufacturerId
  (reserved)
manufacturerName[36]
  (reserved)
productModel[36]
  (reserved)
bUseDefaultCfg
  (reserved)
bPOEDevice
  (reserved)
resv[2]
  (reserved)

Client SDK Instructions

NET_SDK_SEARCH_IMAGE_ITEM

searched face picture information

```
typedef struct _net_sdk_search_image_item_
{
    DD_TIME_EX     recStartTime;
    DD_TIME_EX     recEndTime;
    unsigned int    similarity;              //similarity

    unsigned int    faceFeatureId; //the matched feature when searching by face features

    NET_SDK_FACE_IMG_INFO_CH sfaceImg; //the matched picture when searching by fa
    unsigned char      resv[4];//reserved
}NET_SDK_SEARCH_IMAGE_ITEM;
```

Members

recStartTime
start time of face based recording
recEndTime
end time of face based recording
similarity
similarity
faceFeatureId
target face ID
sfaceImg
matched image information
resv
reserved

Client SDK Instructions

NET_SDK_SEARCH_IMAGE_BY_IMAGE_LIST

the return information of search image by image

```
typedef struct _net_sdk_search_image_by_image_list_
{
    unsigned int    bEnd; //1 indicates finishing searching images; 0 means there are still i

    unsigned int    listNum;//return NET_SDK_SEARCH_IMAGE_ITEM num
    NET_SDK_SEARCH_IMAGE_ITEM *pSearchImageItem;
}NET_SDK_SEARCH_IMAGE_BY_IMAGE_LIST;
```

Members

bEnd
Whether all return or not?

listNum
the number of return data

pSearchImageItem
return the searched face information

Client SDK Instructions

NET_SDK_CH_SNAP_FACE_IMG_LIST

captured face picture data of a camera

```
typedef struct _net_sdk_ch_snap_face_img_list_
{
    unsigned int    bEnd; //1 indicates finishing searching images; 0 means there are still i

    unsigned int    listNum;//return NET_SDK_FACE_IMG_INFO_CH num
    NET_SDK_FACE_IMG_INFO_CH *pCHFaceImgItem;
}NET_SDK_CH_SNAP_FACE_IMG_LIST;
```

Members

bEnd
  Whether all return or not?
listNum
  the number of return data
pSearchImageItem
  return the searched face information

Client SDK Instructions

DD_TIME_EX

Struct of time configuration information of the device. Compared to DD_TIME, the month and year value of DD_TIME is different .

```
 typedef struct _dd_time_ex_
{
    unsigned char       second;         //Seconds after minute (0-59)
    unsigned char       minute;         //Minutes after hour (0-59)
    unsigned char       hour;           //Hours since midnight (0-23)
    unsigned char       wday;           //Day of week (0-6; Sunday = 0)
    unsigned char       mday;           //Day of month (1-31)
    unsigned char       month;          //Month (1-12; January = 1)
    unsigned short      year;           //Year (current year )
    int                         nTotalseconds;          //total seconds

    int                         nMicrosecond;   //microsecond
}DD_TIME_EX;//Compared to DD_TIME, the month and year value of DD_TIME is different
```

Members

second
second; it ranges from 0 to 59.
minute
minute; it ranges from 0 to 59.
hour
Count from 0 0'clock. It ranges from 0 to 23.
wday
Day (it ranges from 0 to 6), A week starts from sunday (the corresponding value is 0).
mday
date of a month. It ranges from 1 to 31.
month
month. It ranges from 1 to 12. It starts from January. The coresponding value of January is 1.
year
Year, the current year

Client SDK Instructions

DD_NETWORK_PLATFORM

Reigster the upper-level platform

```
typedef struct _network_platform
{
        //N9000 supports two platforms: national standard platform and platform software

        unsigned int CurrentPlat;    //the current platform. default: 1 (it indicates platform s

        //platform software

        unsigned int Switcher;       //1 indicates "Enable"，0 indicates "Disable"
        unsigned int Port;                   //port
        unsigned int ReportId;       //device ID
        char szAddress[16];                  //ip address
        //national standard platform，ipc unavailable，N9000 available

        unsigned int SwitchGB;       //11 indicates "Enable"，0 indicates "Disable"
        unsigned int PortGB;         //port

        unsigned int uLocalPort;     //local port
        char szRelm[16];                             //sip server domain

        char szAddressGB[16];        // address

        char szUserName[16];                 //username

        char szPassword[16];         //password

        char szDeviceIdGB[32];       //device ID
        char szServerIdGB[32];       //sip server ID
}DD_NETWORK_PLATFORM;
```

Members

CurrentPlat
the current platform. default: 1 (it indicates platform software)，2 (it indicates national standard platform)

Switcher
platform software; 1 indicates "Enable"，0 indicates "Disable"
Port
platform software; port
ReportId
platform software; device ID
szAddress
platform software; ip address
SwitchGB
national standard platform; 1 indicates "Enable"，0 indicates "Disable
PortGB
national standard platform; port
uLocalPort
national standard platform; local port

szRelm
  national standard platform; sip server domain
szAddressGB
  national standard platform; address
szUserName
  national standard platform; username
  szPassword
  national standard platform; password
szDeviceIdGB
  national standard platform; device ID
szServerIdGB
  national standard platform; sip server ID

Client SDK Instructions

DD_SMART_VFD_CONFIG

Face comparison configuration

```
 typedef struct _dd_smart_vfd_config_
{
      unsigned int iSize;                                   //struct size

      unsigned char  enableFaceDetect;        //enable/disable face detection
      unsigned char  enableSaveFacePicture;   //enable/disable "Save Face Picture"
      unsigned short enableSaveSourcePicture; //enable/disable "Save Source Picture"
      unsigned int   holdTime;               //hold time

      DD_POSITION    startPoint;             //coordinate information of the upper left poin
      DD_POSITION    endPoint;               //coordinate information of the bottom right p
      unsigned int   pushModeType;           //snapshot mode: 0：auto; it will not capture

      unsigned int   intervalTime;           //inteval period of snapshot (seconds)，only wh
}DD_SMART_VFD_CONFIG;
```

Members

iSize
  struct size
enableFaceDetect
  enable/disable face detection
enableSaveFacePicture
  enable/disable "Save Face Picture"
  enableSaveSourcePicture
  enable/disable "Save Source Picture"
  holdTime
  hold time
startPoint
  coordinate information of the upper left point of the rectangle
endPoint
  coordinate information of the bottom right point of the rectangle
  pushModeType
  snapshot mode: 0：auto; it will not capture repeatedly. 1：capture pictures according to the fixed time interval.
intervalTime
  inteval period of snapshot (seconds)，only when the snapshot mode is 1, will it take effect.

Client SDK Instructions

NET_SDK_FACE_INFO_GROUP_ITEM

struct of the group of face comarison

```
 typedef struct _net_sdk_face_info_group_item_
{
    unsigned char        guid[48];                            //GROUP GUID
    char                 name[DD_MAX_NAME_LEN];//GROUP NAME
    unsigned int   property;           //NET_SDK_FACE_INFO_GROUP_PROPERTY_TYPE
    unsigned int    groupId;                //
    unsigned int    enableAlarmSwitch;

}NET_SDK_FACE_INFO_GROUP_ITEM;
```

Members

guid
  group GUID
name
  group name
property
  type of the group; refer to NET_SDK_FACE_INFO_GROUP_PROPERTY_TYPE.
groupId
  group id
enableAlarmSwitch
  Support face match alarm

Client SDK Instructions

NET_SDK_FACE_INFO_GROUP_ADD

Add the group of face comparison

```
 typedef struct _net_sdk_face_info_group_add_
{
    char              name[DD_MAX_NAME_LEN];//GROUP NAME
    unsigned int   property;          //NET_SDK_FACE_INFO_GROUP_PROPERTY_TYPE
}NET_SDK_FACE_INFO_GROUP_ADD;
```

Members

name
  group name
property
  type of the group; refer to NET_SDK_FACE_INFO_GROUP_PROPERTY_TYPE

Client SDK Instructions

NET_SDK_FACE_INFO_GROUP_DEL

Delete the group of face comparison

```
typedef struct _net_sdk_face_info_group_del_
{
    unsigned char        guid[48];                          //GROUP GUID
}NET_SDK_FACE_INFO_GROUP_DEL;
```

Members

guid
   Delet the GUID of the group

Client SDK Instructions

NET_SDK_FACE_INFO_LIST_GET

struct of searching face comparison target

```
 typedef struct _net_sdk_face_info_list_get_
{
 unsigned int     pageIndex;               // 1、2、3... (compulsory)
    unsigned int    pageSize;   //compulsory

    unsigned int    groupId;//1、2、3....(compulsory)

    char          name[DD_MAX_NAME_LEN];//name of NET_SDK_FACE_INFO_LIST_ITEN

    unsigned int    itemId;    // itemID of NET_SDK_FACE_INFO_LIST_ITEM

}NET_SDK_FACE_INFO_LIST_GET;
```

Members

pageIndex
  page number
pageSize
  the number of items in the page
groupId
  group Id。
name
  target name (non-compulsory)
itemId
  target ID (non-compulsory)

Client SDK Instructions

NET_SDK_FACE_INFO_LIST_ITEM_GROUPS

The group the face picture belongs to and the terms of validity the face picture lasts in this group

```
  typedef struct _net_sdk_face_info_list_item_groups_
{
    unsigned int    groupId;        //There is no guid when GROUP id gets target list.
    unsigned char        guid[48];                          //Add GROUP GUID. when target inforn
    DD_TIME_EX    validStartTime;//when the property state is "limited", the validStartTin
    DD_TIME_EX    validEndTime;//when the property state is "limited", the validStartTime
}NET_SDK_FACE_INFO_LIST_ITEM_GROUPS;
```

Members

groupId
group Id
guid
GUID of the group。
validStartTime
valid start time of the group; when the type of the group is "limited", it takes effect.

validEndTime

valid end time of the group; when the type of the group is "limited", it takes effect.

Client SDK Instructions

NET_SDK_FACE_INFO_LIST_ITEM

target face information of face comparsion

```
typedef struct _net_sdk_face_info_list_item_
{
    unsigned int    itemId;                         //id
    char            name[DD_MAX_NAME_LEN];              //compulsory

    unsigned int    sex; //0:male 1:female
    unsigned int    birthday;//eg:19900707
    char            nativePlace[DD_MAX_NAME_LEN];             //
    unsigned int    certificateType; //0:idCard
    char            certificateNum[DD_MAX_CERTIFICATE_NUM];             //
    char            mobile[20];              //
    char            number[20];             //
    unsigned int    faceImgCount;
    NET_SDK_FACE_INFO_LIST_ITEM_GROUPS    groups[DD_MAX_FACE_INFO_GROUPS]

}NET_SDK_FACE_INFO_LIST_ITEM;
```

Members

itemId :target face Id
name: name
sex : gender
birthday :date of birth
nativePlace: native place
certificateType : certifcate type; 0:idCard
certificateNum : certificate number
mobile: phone number
number: nuber
faceImgCount: number of face pictures
groups :group information; one face picture can be added to a maximum of 16 groups.

Client SDK Instructions

NET_SDK_FACE_INFO_LIST

search the returned face comparison target lsit.

```
typedef struct _net_sdk_face_info_list_
{
    unsigned int    totalNum;                              //
    unsigned int    listNum;//return NET_SDK_FACE_INFO_LIST_ITEM num
       NET_SDK_FACE_INFO_LIST_ITEM  *pFaceInfoListItem;
}NET_SDK_FACE_INFO_LIST;
```

Members

totalNum
  total number of target
listNum<
  number of target
pFaceInfoListItem
  target list

Client SDK Instructions

NET_SDK_FACE_IMG_INFO_CH

The face picture captured by the camera can be used as target picture of comparison.

```
 typedef struct _net_sdk_face_img_info_ch_
{
    DD_TIME_EX          frameTime;
   unsigned int    imgId;
   unsigned int    chl; //return value of 255 means the deleted channel.
   unsigned char   resv[8];//reserved

}NET_SDK_FACE_IMG_INFO_CH;
```

Members

frameTime<
  snapshot time

imgId
  image Id

chl
  snapshot channel

Client SDK Instructions

NET_SDK_FACE_INFO_ADD

Add the face picture you want to compare.

```
 typedef struct _net_sdk_face_info_add_
{
     NET_SDK_FACE_INFO_LIST_ITEM sFaceInfoItem;
   unsigned int              imgNum;
     NET_SDK_FACE_IMG_INFO_CH      sFaceImgInfo[DD_MAX_FACE_INFO_IMG];//最大
   unsigned int              haveImgData;//0、1
   unsigned int              imgWidth;//haveImgData ==1  Valid

   unsigned int              imgHeight;//haveImgData ==1 Valid
   unsigned int              imgLen;//haveImgData ==1 Valid
   unsigned char              *imgData;//haveImgData ==1 Valid
}NET_SDK_FACE_INFO_ADD;
```

Members

sFaceInfoItem
Face feature information
imgNum
the number of images in sFaceImgInfo
sFaceImgInfo
image information
haveImgData
Picture data of the external importing picture
imgWidth
picture width of the external importing picture. haveImgData==1 valid
imgHeight
picture height of the external importing picture. haveImgData==1valid
>imgLen<
picture size of the external importing picture.haveImgData==1valid
imgData
picture data. haveImgData==1valid

Client SDK Instructions

NET_SDK_FACE_INFO_EDIT

Edit the face picture you want to compare.

```
 typedef struct _net_sdk_face_info_edit_
{
    unsigned int
delFaceImgs[DD_MAX_FACE_INFO_IMG];
      NET_SDK_FACE_INFO_ADD   sFaceInfoItem;

}NET_SDK_FACE_INFO_EDIT;
```

Members

delFaceImgs
  Delete face image
sFaceInfoItem
  modified target face information

Client SDK Instructions

NET_SDK_FACE_INFO_DEL

Delete the face picture you want to compare.

```
 typedef struct _net_sdk_face_info_del_
{
   unsigned int    faceInfoListItemId;//NET_SDK_FACE_INFO_LIST_ITEM中的itemId
   unsigned int    groupsId[DD_MAX_FACE_INFO_GROUPS];//NET_SDK_FACE_INFO_LIST

}NET_SDK_FACE_INFO_DEL;
```

Members

faceInfoListItemId
  target face Id。
groupsId
  group Id。

Client SDK Instructions

NET_SDK_FACE_MATCH_ALARM_TRIGGER

Alarm linkage information of face match alarm for target groups

```
   typedef struct _net_sdk_face_match_alarm_trigger_
{
    unsigned char       guid[48];   //GROUP GUID
    unsigned int    groupId;    //group ID
    unsigned char    groupSwitch;//enable
    unsigned char    alarmOutSwitch;//trigger alarm output
    unsigned char    alarmOut[16];//trigger a maximum of 16 alarm outputs. The index st
    unsigned char    recSwitch;//recording
    unsigned int    recCH[128];//trigger recording channels/cameras. The index starts fron
    unsigned char    snapSwitch;//snapshot

    unsigned int    snapCH[128];//trigger snapshot channels/camera. The index starts fror
    unsigned int    popVideo;//pop up window

    unsigned char    msgPushSwitch;
    unsigned char    buzzerSwitch;
    unsigned char    popMsgSwitch;
    unsigned char    emailSwitch;

}NET_SDK_FACE_MATCH_ALARM_TRIGGER;
```

Members

guid
Group GUID
groupId
Group ID
groupSwitch
Whether to enable alarm for the group.
alarmOutSwitch
trigger alarmOut
alarmOut
Trigger the channels of alarm out. The index of channels starts from 1.
recSwitch
Trigger recording
recCH
Trigger the recording channels. The index of channels starts from 1.
snapSwitch
Trigger snapshot
snapCH
Trigger snapshot channels. The index of channels starts from 1.
popVideo
Trigger pop-up video. 0 menas no video pops up. Other number means the video pops up.
msgPushSwitch
Trigger message push
buzzerSwitch
Trigger buzzer
popMsgSwitch

Trigger pop-up message

emailSwitch

Trigger email

Client SDK Instructions

NET_SDK_FACE_MATCH_ALARM

Face match alarm

```
typedef struct _net_sdk_face_match_alarm_
{
    unsigned int    similarity;// similarity
    unsigned int    enableCH[128];//【Enable CH】 starts from 1.
    unsigned int    faceFeatureGroupsNum;//number of face group
       NET_SDK_FACE_MATCH_ALARM_TRIGGER  *pFaceMatchAlarmTrigger

}NET_SDK_FACE_MATCH_ALARM;
```

Members

similarity
  face picture similarity
enableCH
  enable channel. It starts from1.
faceFeatureGroupsNum
  Number of linkage alarms of face target groups
sFaceMatchAlarmTrigger
  linkage alarm information of face target groups

Client SDK Instructions

NET_SDK_FACE_INFO_IMG_DATA

Image data of face picture

```
 typedef struct _net_sdk_face_info_img_data_
{
    unsigned int                imgLen;//length of face picture
    unsigned char                *imgData;//face picture data
}NET_SDK_FACE_INFO_IMG_DATA;
```

Members

imgLen
   the length of face picture
imgData
   face picture data

Client SDK Instructions

NET_SDK_FACE_INFO_IMG_GET

area struct

```
 typedef struct _net_sdk_face_info_img_get_
{

    unsigned int    itemId;    //target id
    unsigned int    index;//start index 1 of faceImgCount

}NET_SDK_FACE_INFO_IMG_GET;
```

Members

itemId
   target Id
index
   The index of faceImgCount starts from 1.

Client SDK Instructions

DD_ENCODE_CONFIG_EX

struct of encoding configuration

```
struct _dd_encode_config_{
unsigned long       iSize;
unsigned short      resolution;
unsigned short      rate;
unsigned short      encodeType;
unsigned short      quality;
unsigned short      minBitrate;
unsigned short      maxBitrate;
unsigned short bitrate; //bitrate

unsigned short encodeFormat;  //H264or265 coding DD_VIDEO_ENCODE_FORMA
char recv[14];                  //reserved bytes
}DD_ENCODE_CONFIG;
```

Members

iSize

struct size

resolution

resolution. Refer to DD_VIDEO_SIZE_N9000:

rate

Frame rate

encodeType

encoding type. Refer to the following table.

Type	Value	Description
DD_VIDEO_ENCODE_MODE_VBR	0x01	VBR
DD_VIDEO_ENCODE_MODE_CBR	0x02	BR

quality

Refer to the following table.

Type	Value	Description
DD_IMAGE_QUALITY_LOWEST	0x01	lowest quality
DD_IMAGE_QUALITY_LOWER	0x02	lower quality
DD_IMAGE_QUALITY_LOW	0x03	low quality
DD_IMAGE_QUALITY_MEDIUM	0x04	middle quality
DD_IMAGE_QUALITY_HEIGHTER	0x05	higher quality
DD_IMAGE_QUALITY_HEIGHTEST	0x06	highest quality

minBitrate
  lower limit of bitrate; unit:kbps
maxBitrate
  upper limit of bitrate; unit: kbps
bitrate
  bitrate; unit: kbps
encodeFormat
  Encoding type: H264or265. Refer to DD_VIDEO_ENCODE_FORMAT

Client SDK Instructions

NET_SDK_CH_SNAP_FACE_IMG_LIST_SEARCH

View face pictures.

```
 typedef struct _net_sdk_ch_snap_face_img_list_sreach_
{
    DWORD               dwChannel;//camera/channel of snapshot
    DD_TIME_EX                  startTime; //time

    DD_TIME_EX                  endTime; //time

    DWORD               pageIndex;//page

    DWORD               pageSize;//the number of items in the page

    unsigned char       resv[8];
}NET_SDK_CH_SNAP_FACE_IMG_LIST_SEARCH;
```

Members

Client SDK Instructions

NET_SDK_SEARCH_IMAGE_BY_IMAGE

Search image by image

```
 typedef struct _net_sdk_search_image_by_image_
{
   unsigned int pageIndex; //compulsive 1、2、3...
   unsigned int    pageSize;   //compulsive

   unsigned int    similarity; //similarity

   unsigned int    resultCountLimit; //result limit
   DD_TIME_EX    startTime;
   DD_TIME_EX    endTime;
   unsigned int
searchType;//NET_SDK_SEARCH_IMAGE_BY_IMAGE_TYPE
   struct
   {
      unsigned int itemId; //target id
   } sfaceFeatures;//SEARCH_IMAGE_BY_FACE_FEATURES
   NET_SDK_FACE_IMG_INFO_CH
sfaceImgs;//SEARCH_IMAGE_BY_FACE_IMAGES

   struct
   {
      unsigned int groupsId; //GROUP Id
   }sfaceFeatureGroups
;//SEARCH_IMAGE_BY_FACE_FEATURE_GROUPS

   struct
   {
      unsigned int     isContainRecognized; //0 or 1
      unsigned int     isContainNotRecognized; //0 or 1
      unsigned int     groupsId; //GROUP Id
```

```
    }srecognizedFilter
;//SEARCH_IMAGE_BY_RECONGNIZED_FILTER
    struct
    {
        unsigned int                imgWidth;//
        unsigned int                imgHeight;//
        unsigned int                imgLen;//
        unsigned char                *imgData;//

}sfaceImgData;//SEARCH_IMAGE_BY_FACE_IMAGE_DATA

}NET_SDK_SEARCH_IMAGE_BY_IMAGE;
```

Members

Client SDK Instructions

DECODE_FRAME_INFO

Decode YUV frame information

```
struct decode_frameInfo
{
int nWidth;
int nHeight;
unsigned int time;
unsigned int dwLen;
unsigned char *pData;
}DECODE_FRAME_INFO;
```

Members

nWidth
  frame width
nHeight
  frame height
time
  frame time stamp
dwLen
  decode frame length
pData
  decode frame data

Client SDK Instructions

NET_SDK_USB_BACKUP_PROCESS_EX

the process and status of the saving record to USB device

```
typedef struct _usb_backup_process_ex
{
                 DD_TIME_EX                     startTime; // the start time of the recor
        DD_TIME_EX                      endTime; // the end time of the record
        unsigned int        dataSize;//MB  the size of the record
        unsigned char       backupPath[64]; // the usb path of the backup
        unsigned char       creator[36];// the creator of the backup task
        unsigned int        progress;//0-100, the process of the backup 0-100
        unsigned int        backupFileFormat;//
        unsigned int        status;//
        unsigned int        eventType;//
        unsigned char           chls[64];// the channel of the backup
        unsigned int        chlNum;// the actual number of the channels

}NET_SDK_USB_BACKUP_PROCESS_EX;
```

Members

startTime
the start time of the record

endTime
the end time of the record

dataSize
MB the size of the record

backupPath
the usb path of the backup

creator
the creator of the backup task

progress
0-100, the process of the backup 0-100

backupFileFormat
0 is avi,1 is private format

status
0 is backuping, 1 is complete

eventType
refer to DD_RECORD_TYPE

chls
the channels of the backup

chlNum
the number of the backup channels

Client SDK Instructions

NET_SDK_IVE_VEHICE_ITEM_INFO

Vehicle number information

```
typedef struct
{
  unsigned int  begin_flag;
  unsigned int  data_type;
  unsigned int  image_type;
  unsigned int plateId;
  unsigned int plateCharCount;
  char plate[32];
  char plateCharConfid[32];
  NET_SDK_IVE_RECT_T ptPlateCharRect[32];
  unsigned in ptWidth;
  unsigned int ptHeight;
  NET_SDK_IVE_POINT_T ptLeftTop;
  NET_SDK_IVE_POINT_T ptRightTop;
  NET_SDK_IVE_POINT_T ptLeftBottom;
  NET_SDK_IVE_POINT_T ptRightBottom;
  unsigned short plateWidth;
  unsigned short plateHeight;
  unsigned int plateConfidence;
  unsigned int plateIntensity;
  unsigned char      plateColor;
  unsigned char plateStyle;
  unsigned char PlateColorRate;
  unsigned char vehicleColor;
  unsigned int plateAngleH;
  unsigned int plateAngleV;
  unsigned in jpeg_len;
  unsigned int  jpeg_vir_len;
  char owner[32];
  int listType;
  unsigned long long beginTime;
  unsigned long long  endTime;
  unsigned char      iVehicleDirect;
  unsigned char resrv[11];
  unsigned  int  end_flag;
}NET_SDK_IVE_VEHICE_ITEM_INFO;
```

Members

begin_flag
start identification，0x5a5a5a5a
data_type
0：JPG,1:YUV
image_type
0:source image，1：vehicle number
plateId
ID，just for identification
plateCharCount
the number of the characters of the vehicle number
plate
the vehicle number, utf8 codec
plateCharConfid
the confidence of the vehicle number
ptPlateCharRect
the coordinate of the upleft of the vehicle number
ptWidth
the width of the vehicle number（for drawing rectangle to following the vehicle number plate）
ptHeight
the height of the vehicle number
ptLeftTop
coordinate of the upleft of the plate
ptRightTop
coordinate of the upright of the plate
ptLeftBottom
coordinate of the downleft of the plate
ptRightBottom
coordinate of the downright of the plate
plateWidth
width of the plate
plateHeight
height of the plate
plateConfidence
plate confidence
plateIntensity

plate intensity

plateColor
: the color of the vehicle number plate // 0-blue 1-black 2-yellow 3-white 4-green 5-red 6-gray 7-purple(KISE)

plateStyle
: plate style

PlateColorRate
: the similarity of the plate color

vehicleColor
: color of the vehicle

plateAngleH
: horizen angle of the plate

plateAngleV
: vertical angle of the plate

jpeg_len
: the lenth of the jpeg image data

jpeg_vir_len
: the total lenth of the jpeg image data

owner
: the name of the vehicle's owner

listType
: list type,0-comparision failed，1-strange，2-white list，3-black list，

beginTime
: start time

endTime
: end time

iVehicleDirect
: vehicle's direction, 1 unknown 2 near 3 far

resrv
: preserve

end_flag
: end identification，0xa5a5a5a5

Client SDK Instructions

NET_SDK_IVE_VEHICE_HEAD_INFO

Vehicle number detection alarm call back information header

```
typedef struct
{
  unsigned int begin_flag;
  unsigned int item_cnt;
  unsigned int plate_cnt;
  long long relativeTime;
  long long absoluteTime;
  unsigned int softwareVersion;
  unsigned int softwareBuildDate;
  unsigned int resver[2];
  unsigned int  end_flag;
}NET_SDK_IVE_VEHICE_HEAD_INFO;
```

Members

begin_flag
start identification，0x5a5a5a5a

item_cnt
the number of NET_SDK_IVE_VEHICE_ITEM_INFO

plate_cnt
the number of vihicle number has detected

relativeTime
the relative time of the detected happened

absoluteTime
the absolute time of the detected happened

softwareVersion
the version of the software, 0xABCDEFGH,AB：Brand CD：main version EFGH：sub version Brand 1:OMRON version:V5.00

softwareBuildDate
the build time of the software,0xYYYYMMDD

resver
Preserve
end_flag
end identification，0xa5a5a5a5

Client SDK Instructions

NET_SDK_DEV_SUPPORT

the functions of the IPC

```
struct {
unsigned int   supportThermometry:1;                          //support mask and temperature
unsigned int   supportVfd:1;                          //support face detect
unsigned int   supportVfdMatch:1;                             //support face match
unsigned int   supportThermal:1;                              //thermal
unsigned int   supportPassLine:1;                             //pass line
unsigned int   supportresv:27;                        //
unsigned int   resv[15];                              //
}NET_SDK_DEV_SUPPORT;
```

Client SDK Instructions

REG_LOGIN_INFO

the information of the auto register device。

```
 typedef struct _reg_login_info
{
unsigned int deviceId;//register id
        char m_szUserName[36];//user name
        char m_szPasswd[36];//password
 }REG_LOGIN_INFO;
```

Client SDK Instructions

SEARCHED_DEVICE_INFO

searched device's information

```
struct  _searched_deviceInfo{
       char                    series[64];
       char                    devName[64];
       char            deviceType[16];
       char                    szproductModel[16];
       char                    szVersion[32];
       char                    szFactoryName[16];
       char            szEthName[16];
       unsigned short  netport;
       unsigned short  nHttpPort;
       unsigned int                    ipaddr;
       unsigned int                    gateway;
       unsigned int                    netmask;
       unsigned int                    dns1;
       unsigned int                    dns2;
       unsigned short  nChannelCount;  //NVR's channel count
       unsigned int    dwSecondIP;
       unsigned int    dwSecondMask;
}SEARCHED_DEVICE_INFO;
```

Members

series[64]
   series
devName[64]
   device name
deviceType[16]
   device type
szproductModel[16]
   product model
szVersion[32]
   version
szFactoryName[16]
   factory name
szEthName[16]

ethnet name

netport;

net port

nHttpPort;

http port

ipaddr

ip address

gateway

gate way

netmask

net mask

dns1

dns1

dns2

dns2

nChannelCount

NVR channel count

dwSecondIP

second ip

dwSecondMask

second mask

Client SDK Instructions

NET_SDK_IVE_FACE_MATCH_ADD_FACE_REPLY_T

return struct of the adding face to IPC

```
typedef struct _net_sdk_ive_face_match_add_face_reply_t
{
   unsigned int              dwResult;
        int                  iPersonId;                // person ID.
        char                 szRes[32];
 }NET_SDK_IVE_FACE_MATCH_ADD_FACE_REPLY_T;
```

Members

dwResult;
  result。

iPersonId
  person ID。

szRes
  reserve。

Client SDK Instructions

NET_SDK_IVE_PASSLINECOUNT_T

pass line information

```
typedef struct
{
        unsigned int            enterCarCount; //enter car count
        unsigned int            enterPersonCount;//enter person count
        unsigned int            enterBikeCount;//enter bike count
        unsigned int            leaveCarCount; //leave car count, it'll be 0 if single directio
        unsigned int            leavePersonCount;//leave person count, it'll be 0 if single dir
        unsigned int            leaveBikeCount;//leave bike count, it'll be 0 if single directi
        unsigned int            existCarCount;//exist car count, it'll be 0 if single direction
        unsigned int            existPersonCount;//exist person count, it'll be 0 if single dir
        int                                             existBikeCount;//exist bike count, it'
        unsigned int           count;          //count
        NET_SDK_IVE_PASSLINECOUNT_INFO_T  passLineInfo[32];      // pass line analyse result i
}NET_SDK_IVE_PASSLINECOUNT_T;
```

Client SDK Instructions

NET_SDK_IVE_PASSLINECOUNT_INFO_T

pass line analyse result information

```
typedef struct
{
        unsigned int         eventId;             // event id
        unsigned char          status;              // alarm status,0:none 1:start 2:end 3:proc
        unsigned char          reserve[3];          // reserve
        unsigned int         targetId;            // target ID
        NET_SDK_IVE_LINE_T      line;                // pass line rule
        NET_SDK_IVE_RECT_T      rect;                // target rectangle
}NET_SDK_IVE_PASSLINECOUNT_INFO_T;
```

Client SDK Instructions

NET_SDK_IVE_LINE_T

pass line rule

```
typedef struct
{
       unsigned int X1;   // start x coodinate
        unsigned int Y1;   // start y coodinate
        unsigned int X2;   // end x coodinate
        unsigned int Y2;   // end y coodinate
}NET_SDK_IVE_LINE_T;
```

Client SDK Instructions

NET_SDK_IVE_RECT_T

target rectangle

```
typedef struct
{
       unsigned int X1;   // top left x coodinate
        unsigned int Y1;   // top left y coodinate
        unsigned int X2;   // right down x coodinate
        unsigned int Y2;   // right down y coodinate
}NET_SDK_IVE_RECT_T;
```

Client SDK Instructions

NET_SDK_IVE_AVD_T

struct of abnormal video detection

```
    typedef struct
    {
        unsigned int          count;
        NET_SDK_IVE_AVD_INFO_T avdInfo[32];
}NET_SDK_IVE_AVD_T
```

Members

count
    count
avdInfo
    avd information struct array

Client SDK Instructions

NET_SDK_IVE_AVD_INFO_T

struct of abnormal video detection detail

```
    typedef struct
    {
        unsigned int      eventId;
        unsigned int     status;
        unsigned int     type;
    }NET_SDK_IVE_AVD_INFO_T
```

Members

eventId
  eventId
status
  0:none 1:start 2:end 3:procedure
type
  0:none 1:Scene 2:Clarity 3:Color

Client SDK Instructions

NET_DVR_IVE_VFD_RESULT_HEAD_T

struct of video face detection result head

```
typedef struct
{
   LONGLONG      time;
   LONGLONG      relativeTime;
   unsigned int     detectDataLen;
   unsigned int     softwareVersion;
   unsigned int     softwareBuildDate;
   unsigned int     faceCnt;
   unsigned int     faceDataLen[40];
}NET_DVR_IVE_VFD_RESULT_HEAD_T
```

Members

time
  current time
relativeTime
  relative time
detectDataLen
  detect data length
softwareVersion
  software version
  0xABCDEFGH,AB:manufacture,CD:major version
  EFGH:minor version
softwareBuildDate
  software build date
faceCnt
  face count, max is 40
faceDataLen
  face data length

Client SDK Instructions

NET_DVR_IVE_VFD_RESULT_DATA_INFO_T

struct of video face detection result data information

```
 typedef struct
{     unsigned  inttype;
        unsigned     int status;
        unsigned int     width;
        unsigned int     height;
        unsigned int     dataLen;
    }NET_DVR_IVE_VFD_RESULT_DATA_INFO_T
```

Members

type
: 0, JPG; 1, YUV

status
: 0, INVALID; 1, VALID; 2, SAVED

width
: width

height
: height

dataLen
: data Length

Client SDK Instructions

NET_DVR_IVE_VFD_RESULT_FACE_DATA_INFO_T

face data information

```
    typedef struct
{
    int                         faceId;
    unsigned int                        ptWidth;
    unsigned int                        ptHeight;
     NET_SDK_IVE_POINT_T ptLeftTop;
    NET_SDK_IVE_POINT_T ptRightTop;
   NET_SDK_IVE_POINT_T ptLeftBottom;
    NET_SDK_IVE_POINT_T ptRightBottom;
    int                         nPose;
    int                         nConfidence;
    int                         age;
    int                         sex;
    int                         dtFrames;
    int                            featureSize;
    NET_SDK_IVE_POINT_T stPosFaceImg;


        float                           feature_score;


        short                           eye_dist;
        short                           blur;

        char                            pose_est_score;
        char                            detect_score;
        char                            illumination;
        char                            faceliveness;

        char                            completeness;
        char                            glasses;
        char                            wearmask;
        char                            reserved1[1];

        float                           comprehensive_score;
        int                          temperature;

        int                          foreheadX;
        int                         foreheadY;

        NET_SDK_IVE_POINT_T         stHotLeftTop;
        NET_SDK_IVE_POINT_T         stHotRightBottom;
        char                        cTemperatureMode;
        char                        tempUnitsType;
        char                        cTemperatureStatus;
        char                         reserved[5];
    NET_DVR_IVE_VFD_RESULT_DATA_INFO_T stFaceImgData;
}NET_DVR_IVE_VFD_RESULT_FACE_DATA_INFO_T;
```

Members

faceId
face ID Number
ptWidth
width
ptHeight
height
ptLeftTop
Left-Top Face Coordinates
ptRightTop
Right-Top Face Coordinates
ptLeftBottom
Left-Bottom Face Coordinates
ptRightBottom
Right-Bottom Face Coordinates
nPose
Face Pose
nConfidence
Confidence Degree
age
age
sex
sex
dtFrames
dtFrames
featureSize

feature size

stPosFaceImg

the coodinate of the image left top

feature_score

feature score 0-100

eye_dist

distance of the eyes

blur

blur

pose_est_score

pose est_score 0-100

detect_score

detect score 0-100

illumination

illumination

faceliveness

faceliveness0~100

completeness

completeness 0~100

glasses

if wear glasses

wearmask

if wear mask

reserved1

reserved1

comprehensive_score

comprehensive score [90,100)best， [80,90)better，[70,80)good，[60,70)normal，[50,60)medium，[0,50)bad。

temperature

temperature

foreheadX

forehead X coordinate

foreheadY

forehead Y coordinate

stHotLeftTop
: hot left top Coordinates

stHotRightBottom
: hot right top Coordinates

cTemperatureMode
: 0:normal 1:validate

tempUnitsType
: 0:Celsius 1: Fahrenheit

cTemperatureStatus
: 0:normal，1:temperature too low，2:temperature too high

reserved
: reserved

stFaceImgData
: face image data

Client SDK Instructions

NET_SDK_IVE_FACE_MATCH_T

struct of face match

```
typedef struct
 {
        DD_TIME_EX frameTime;
        unsigned int dwRealFaceID;
        unsigned int dwGrpID;
        unsigned int dwLibFaceID;
        unsigned int dwSimilar;
        unsigned char byName[32];
        unsigned int Channel;
        unsigned int  imgLen;
    }NET_SDK_IVE_FACE_MATCH_T
```

Members

frameTime
    frameTime
dwRealFaceID
    snap face id
dwGrpID
    group id
dwLibFaceID
    library face id
dwSimilar
    similarity
byName
    name
Channel
    Channel
imgLen
    image length

Client SDK Instructions

NET_SDK_AVPSTORE_FACE_ABSTRACT_INFO

struct of face abstract

```
typedef  struct
         {
         char szName[32];
        unsigned int dwBirth;
        char szNativePlace[16];
        char szNote[16];
        unsigned char byPicNum;
        unsigned char byTypeCredential;
        unsigned char bySex;
        unsigned char byGroupCount;
        unsigned char byGroupID[4];
        union
        {
                struct
                {
                        unsigned int dwStartTime;
                        unsigned int dwReserve[3];
                        unsigned int dwEndTime;
                        unsigned char  byReserve[11];
                        unsigned char  byContentType;
                }PeriodV1;

                struct
                {
                        unsigned int  byWeekOrDate;
                        unsigned int  dwReserve[3];
                        unsigned short  wStartTime;
                        unsigned short  wEndTime;
                        unsigned short  wReserve[5];
                        unsigned char   byMode;
                        unsigned char   byContentType;
                }PeriodV2;

                struct
                {
                        unsigned char  dwReserve[31];
                        unsigned char  byContentType;
                }PlaceHolder;
        }TimeCycle;

        char szCredential[32];
        unsigned char   byPhoneNum[16];
        unsigned char   byIDParam[16];
    }NET_SDK_AVPSTORE_FACE_ABSTRACT_INFO
```

Members

szName
name
dwBirth
birthday like 19991220
szNativePlace
native place
szNote
note
byPicNum
number of picture
byTypeCredential
credential type
bySex
0:male 1:female
byGroupCount
group count
byGroupID
group id
ress
reserve
szCredential
credential id
byPhoneNum
phone number
byIDParam
id

Client SDK Instructions

NET_SDK_TLV_BUFFER_DESC

struct of buffer description

```
public struct NET_SDK_TLV_BUFFER_DESC
{
       unsigned char ucID;
       unsigned char ucVersion;
       unsigned short usNumber;
       unsigned int dwSize;

   }
```

Members

ucID
 id
ucVersion
 version
usNumber
 number
dwSize
 the source image's size

Client SDK Instructions

NET_SDK_IVE_BASE_INFO

struct of IPC face match base information

```
typedef struct NET_SDK_IVE_BASE_INFO_T
 {
        long long             i64SnapTime;
        unsigned int              iSnapPicId;

        int                   iSimilarity;
        int                   iPersonId;
        int                   iType;
        char                  szName[128];
        int                   iMale;
        int                   iAge;
        char                  szIdentifyNum[128];
        char                  szTel[64];
        char                  szRes[128];

        int                   iSnapPicQuality;
        int                   iSnapPicAge;
        int                   iSnapPicSex;

        char                  livingBody;
        char                  comparisonRes;
        char                  wearmask;
        char                  tempUnitsType;
        int                   temperature;

        char                  keyID[36];
        char                  szReserve[20];

    }NET_SDK_IVE_BASE_INFO
```

Members

i64SnapTime
  snap time
iSnapPicId
  snap picture id
iSimilarity

(0-100) similarity

iPersonId
: the person's id

iType
: 0:stranger 1:white list 2: black list

szName
: name

iMale
: 1:male 0:female.

iAge
: age

szIdentifyNum
: identify number

szTel
: telphone number

szRes
: reserve

iSnapPicQuality
: snap picture's quality

iSnapPicAge
: snap picture's age

iSnapPicSex
: snap picture's sex

livingBody
: 1:living body 0:not

comparisonRes
: comparision result 1:success 0:failed

wearmask
: if ware mask 0:not detect 1:not wear 2 wear mask

tempUnitsType
: temperature unit type 0:celsius 1:Fahrenheit

temperature
: temperature

keyID
: keyID

szReserve

reserve

Client SDK Instructions

NET_SDK_IVE_PICTURE_INFO

struct of IPC snap picture information

```
typedef struct NET_SDK_IVE_PICTURE_INFO_T
 {
        int                   iWidth;
        int                   iHeight;
        int                   iPicFormat;
        int                   iPicSize;
    }NET_SDK_IVE_PICTURE_INFO
```

Members

iWidth
: picture's width

iHeight
: picture's height

iPicFormat
: picture's format

iPicSize
: picture's size

Client SDK Instructions

NET_SDK_IVE_POINT_T

point

```
 typedef struct
{
    int X;
    int Y;
}NET_SDK_IVE_POINT_T
```

Members

X1
    x coodinate
Y1
    y coodinate

Client SDK Instructions

NET_SDK_NVR_DISKREC_DATE_ITEM

the structure of NVR record days information

```
typedef struct _net_sdk_nvr_diskrec_date_item
{
 unsigned int    diskCount;
 unsigned int    diskIndex;
 char             szDiskSizeGB[16];
 char          szStartDate[32];
 char          szEndDate[32];
}NET_SDK_NVR_DISKREC_DATE_ITEM;
```

Members

diskCount
  disk count.
diskIndex
  disk index.
szDiskSizeGB
  the size(GB) of the disk.
szStartDate
  recording start day.
szEndDate
  recording end day.

Client SDK Instructions

NET_SDK_DiscoverDevice

discover device automatically on LAN

```
long NET_SDK_DiscoverDevice(
  NET_SDK_DEVICE_DISCOVERY_INFO          *pDeviceInfo,
long   bufNum;
long   waitSeconds;
);
```

Parameters

**pDeviceInfo*
: [in] an array witch is needed to asign values,its size is **bufNum** ,if descovered device num is more than,the returned size is just **bufNUm**

bufNum
: [in] size of the array

waitSeconds
: [in] time to discover devices, unit is second,this interface will be returned after **waitSeconds**

Return Values

Returned value is the num of discovered devices,if no deivce is found or discovering device gets error,the value is 0. Get error info refer to NET_SDK_GetLastError

See Also

NET_SDK_GetDeviceInfo

Client SDK Instructions

FAQ

Q1 How to get alarm means & invoking method?

A1

What is called protection is SDK connects device actively,device starts loading alarm,once alarm happens alarm information is uploaded to SDK as soon as possible.So except that alarm inputs information & device invokes callback function,the interface NET_SDK_SetupAlarmChan also should be invoked to setup connection between SDK and device.

Q2 Alarm configuration has succeed and alarm signal can be received

A2

Reasons as follows:1)whether network connection is normal 2)when alarm type is protection,whether setup protection correctly.

Q3 Why is it that returned value is failure when calling NET_SDK_Set

A3

NET_SDK_EnterDVRConfig() must be called to lock config before calling NET_SDK_SetDVRConfig().

Q4 Why is it that the start time of playback and downloading is differ

A4

Playback & download start from the nearby key frame of the setting start time.

Q5 Why need to pass a group of channel numbers to NET_SDK_PlayB

A5

The passed group channels realize autosynchronous play, at the same time divide channels into groups to play but not play by itself, these can save device-side performance.

Q6 Why the time of record data index,playback and download is diffe

A6

If this problem appears,first check that whether device timezone and PC timezone is the same,and then check that the time of the two machines is the same.

Q7 What to notice when using configurated parameters in NET_SDK_

A7

Because NET_SDK_SetDVRConfig() needs struct with complete assignment,otherwise setting error comes out. So for fear of this error, popularly invoke NET_SDK_GetDVRConfig() to assign initial values to the struct which is needed modification before invoking NET_SDK_SetDVRConfig().

Q8 Why does control command of NET_SDK_PTZControl have no effec

A8

Device sends control code to PTZ according to decoder type and decoder address.If current decoder unmatchable,matching decoder setup is needed; if device doesnot support the decoder, control command from device has no effect on the PTZ.

Q9 Do audio talkback & forward aim at device or channel?

A9

Aim at device ,not channel.

Q10 Whether the callback function of audio talkback can set be null o

A10

Yes,if be null,vioce is still normal but user can't access data.

Q11 How to save record data into files?

A11

Get data through callback function LIVE_DATA_CALLBACK of NET_SDK_SetLiveDataCallBack,and then save the data into files,refer to the example in livedlg.cpp of SDKdemo,see L1170.You can play the saved files by Player.

Q12 How to get play progress?

A12

Get start time and end time by NET_SDK_PlayBackByTime,and then get current playing time by NET_SDK_GetPlayBackOsdTime.Play progress=current playing time/(end time-start time).

Q13 How to do when PlayerDemo gets error code 0XC0150002?

A13

Solution:install Microsoft Visual C++ 2005 SP1 Redistributable Package4,download the module in MSDN.

Q14 Why some function are invalid when palyback?

A14

2X and 4X speed is invalid in SDKDemo and SDK when backward,but other speed is OK.When forward all speed is OK except 1X.Before starting playback one frame by one frame,*Pause* should be enabled,and then one frame can be played by click *Next frame* button one time.

Q15 What is *wday* in DD_TIME struct?

A15

Start time is DD_TIME type in NET_SDK_FindFile,but *wday* is invalid in DD_TIME.*wday* can be empty but can not be cleared.You can search by *mday* or write a function to convert time into wday.

Q16 How to do when play file gets E_PLAYER_BAD_FORMAT_FILE err

A16

Check the following four qustions:

1the first frame is format frame when save record file,

2all structs in SDK are 4 bytes alignment,

3check interface calling order,

4the file in PlayerSDK should be :

*frame info£¨SDK_FRAME_INFO£©valid data in frame

*frame info£¨SDK_FRAME_INFO£©valid data in frame

* ......

*frame info£¨SDK_FRAME_INFO£©valid data in frame

* ......

* video info frame should be before video frame,audio info frame should be before audio frame too

Client SDK Instructions

NET_SDK_GetLastError

return the last error code of operation

```
DWORD NET_SDK_GetLastError(
);
```

Return Values

return value is pointer to error code information. error message has two main types,error message of network communication library and error message of soft and hard decoding library,list the first type as follows:

error message of network communication library

type of errors	error value	
NET_SDK_SUCCESS	0	no error
NET_SDK_PASSWORD_ERROR	1	user's name or pa
NET_SDK_NOENOUGH_AUTH	2	no right for this o
NET_SDK_NOINIT	3	SDK is not initial
NET_SDK_CHANNEL_ERROR	4	error of channel n
NET_SDK_OVER_MAXLINK	5	the client connect
NET_SDK_LOGIN_REFUSED	6	SDK login is refu
NET_SDK_VERSION_NOMATCH	7	version doesn't m
NET_SDK_NETWORK_FAIL_CONNECT	8	failed to connect
NET_SDK_NETWORK_NOT_CONNECT	9	network isn't con
NET_SDK_NETWORK_SEND_ERROR	10	failed to send dat
NET_SDK_NETWORK_RECV_ERROR	11	failed to receive t
NET_SDK_NETWORK_RECV_TIMEOUT	12	timeout when rec
NET_SDK_NETWORK_ERRORDATA	13	send illegal data t
NET_SDK_ORDER_ERROR	14	the called order e
NET_SDK_OPER_BY_OTHER	15	operation method
NET_SDK_OPER_NOPERMIT	16	the privileged use
NET_SDK_COMMAND_TIMEOUT	17	DVR command ti
NET_SDK_ERROR_SERIALPORT	18	error of serial por
NET_SDK_ERROR_ALARMPORT	19	error of alarm por
NET_SDK_PARAMETER_ERROR	20	parameter error
NET_SDK_CHAN_EXCEPTION	21	server's channel i
NET_SDK_NODISK	22	no hard disk
NET_SDK_ERROR_DISKNUM	23	hard disk no. erro
NET_SDK_DISK_FULL	24	server hark disk i
NET_SDK_DISK_ERROR	25	server hard disk e
NET_SDK_NOSUPPORT	26	server does not su
NET_SDK_BUSY	27	server is busy
NET_SDK_MODIFY_FAIL	28	failed to modify i
NET_SDK_PASSWORD_FORMAT_ERROR	29	the password inpu
NET_SDK_DISK_FORMATING	30	hard disk is forma
NET_SDK_DVR_NORESOURCE	31	DVR no resource
NET_SDK_DVR_OPRATE_FAILED	32	DVR failed to op
NET_SDK_OPEN_HOSTSOUND_FAIL	33	failed open PC vo
NET_SDK_DVR_VOICEOPENED	34	server voice dialo
NET_SDK_TIME_INPUTERROR	35	time input is not c
NET_SDK_NOSPECFILE	36	there is no appoin
NET_SDK_CREATEFILE_ERROR	37	failed to create a
NET_SDK_FILEOPENFAIL	38	faile to open a fil
NET_SDK_OPERNOTFINISH	39	the last operation
NET_SDK_GETPLAYTIMEFAIL	40	faile to get the cu
NET_SDK_PLAYFAIL	41	failed to play

NET_SDK_FILEFORMAT_ERROR	42	the file input forn
NET_SDK_DIR_ERROR	43	path error
NET_SDK_ALLOC_RESOURCE_ERROR	44	resources allottin
NET_SDK_AUDIO_MODE_ERROR	45	display card mod
NET_SDK_NOENOUGH_BUF	46	buffer is not enou
NET_SDK_CREATESOCKET_ERROR	47	establish SOCKE
NET_SDK_SETSOCKET_ERROR	48	set SOCKET erro
NET_SDK_MAX_NUM	49	the max number
NET_SDK_USERNOTEXIST	50	user doest not exi
NET_SDK_WRITEFLASHERROR	51	wirte FLASH err
NET_SDK_UPGRADEFAIL	52	failed to upgrade
NET_SDK_CARDHAVEINIT	53	the decode card i
NET_SDK_PLAYERFAILED	54	player failed
NET_SDK_MAX_USERNUM	55	the max user no.
NET_SDK_GETLOCALIPANDMACFAIL	56	failed to get the I end or physical a
NET_SDK_NOENCODEING	57	the channel is not
NET_SDK_IPMISMATCH	58	IP address not ma
NET_SDK_MACMISMATCH	59	MAC address not
NET_SDK_UPGRADELANGMISMATCH	60	the language of u
NET_SDK_MAX_PLAYERPORT	61	reach to the max
NET_SDK_NOSPACEBACKUP	62	no enough space
NET_SDK_NODEVICEBACKUP	63	no backup device
NET_SDK_PICTURE_BITS_ERROR	64	the bits of picture
NET_SDK_PICTURE_DIMENSION_ERROR	65	the dimension is
NET_SDK_PICTURE_SIZ_ERROR	66	the size of picture
NET_SDK_LOADPLAYERSDKFAILED	67	failed to load pla
NET_SDK_LOADPLAYERSDKPROC_ERROR	68	not find some fun
NET_SDK_LOADDSSDKFAILED	69	failed to load Ds
NET_SDK_LOADDSSDKPROC_ERROR	70	not find some fun
NET_SDK_DSSDK_ERROR	71	failed to call func
NET_SDK_VOICEMONOPOLIZE	72	voice card is mon
NET_SDK_JOINMULTICASTFAILED	73	failed join to mul
NET_SDK_CREATEDIR_ERROR	74	failed to create lo
NET_SDK_BINDSOCKET_ERROR	75	failed to bind soc
NET_SDK_SOCKETCLOSE_ERROR	76	socket is closed
NET_SDK_USERID_ISUSING	77	the user ID is ope
NET_SDK_PROGRAM_EXCEPTION	78	sdk program exce
NET_SDK_WRITEFILE_FAILED	79	write file failed
NET_SDK_FORMAT_READONLY	80	failed to format r
NET_SDK_WITHSAMEUSERNAME	81	there is same use
NET_SDK_DEVICETYPE_ERROR	82	device type no m
NET_SDK_LANGUAGE_ERROR	83	language no matc
NET_SDK_PARAVERSION_ERROR	84	soft version no m
NET_SDK_FILE_SUCCESS	85	file has been crea
NET_SDK_FILE_NOFIND	86	file isn't found

NET_SDK_NOMOREFILE	87	there is no more f
NET_SDK_FILE_EXCEPTION	88	file exception
NET_SDK_TRY_LATER	89	Try again later
NET_SDK_DEVICE_OFFLINE	90	Device offline
NET_SDK_CREATEJPEGSTREAM_FAIL	91	Failed to create JI
NET_SDK_USER_ERROR_NO_USER	92	No such user!
NET_SDK_USER_ERROR_USER_OR_PASSWORD_IS_NULL	93	No username or p
NET_SDK_USER_ERROR_ALREDAY_LOGIN	94	The user has been
NET_SDK_USER_ERROR_SYSTEM_BUSY	95	The device is bus
NET_SDK_DEVICE_NOT_SUPPROT	96	The device don n
NET_SDK_USER_ERROR_SYSTEM_NO_READY	97	Do not complete
NET_SDK_CHANNEL_OFFLINE	98	Camera is offline
NET_SDK_GETREADYINFO_FAIL	99	It fails to get devi
NET_SDK_NORESOURCE	100	SDK resources is
NET_SDK_DEVICE_QUERYSYSTEMCAPS_FAIL	101	The device fails t
NET_SDK_INBUFFER_TOSMALL	102	The input buffer a
NET_SDK_NO_PASSWORD_STRENGTH	103	The password stre

Remarks

Get error number through NET_SDK_GetErrorMsg

See Also

NET_SDK_GetErrorMsg