

java sdk Interface instruction manual

catalog

1. cover	3
2. Introduction	4
3. Revision Record	5
4. programming instructions	6
4.1 The development environment	6
4.2 API call process	7
4.3 Note	8
5. interface definition	9
5.1 Connect device	9
5.1.1 Connect device	9
5.1.1.1 Connect camera	9
5.1.1.2 encrypt connection	10
5.1.1.3 disconnect	11
5.1.2 Device connection status	12
5.1.2.1 Get device status	12
5.1.2.2 Set device event callback	13
5.1.2.3 device event callback	14
5.2 Plate recognition	15
5.2.1 Software trigger	15
5.2.1.1 Software trigger	15
5.2.2 recognition data reception	16
5.2.2.1 Register callback for license plate recognition event	16
5.2.2.2 Set callback for license plate recognition event	17
5.2.2.3 License plate recognition data callback function	18
5.2.2.4 Unregister callback for license plate recognition event	22
5.3 real-time video and capture	23
5.3.1 real-time video	23
5.3.1.1 mjpeg bitstream callback	23
5.3.1.2 set mjpeg bitstream callback	24
5.3.2 capture	25
5.3.2.1 capture	25
5.4 Device management	26
5.4.1 Get mac of device	26
5.4.2 Set the camera network configuration	27
5.4.3 Get the camera network configuration	28
5.4.4 Get UID of camera	29
5.4.5 Get time of camera	30
5.4.6 Time synchronization	31
5.5 Peripherals management	32
5.5.1 open gate	32
5.5.2 Control alarmout	33

5.5.3	Get IO status	34
5.5.4	Register IO change event	35
5.5.5	Set event callback for IO state change.....	36
5.5.6	IO state change event	37
5.5.7	Unregister IO change event	38
5.6	Camera configuration	39
5.6.1	Black and white list parameters	39
5.6.1.1	Get the black and white list parameters	39
5.6.1.2	Set the black and white list parameters.....	42
5.6.2	DNS configuration	45
5.6.2.1	Set DNS configuration	45
5.6.2.2	Get DNS configuration.....	46
5.6.3	Alarm IO configuration.....	47
5.6.3.1	Get alarm IO configuration.....	47
5.6.3.2	Set alarm IO configuration	48
5.6.4	Serial port configuration	49
5.6.4.1	Get serial port configuration	49
5.6.4.2	Set serial port configuration.....	50
5.7	Transparent serial port.....	51
5.7.1	Send data through serial port in SerinalNET mode	51
5.7.2	Register serial port to receive events.....	52
5.7.3	Set transparent serial port data reporting event callback (RS485-1)	53
5.7.4	Set transparent serial port data reporting event callback (RS485-2/RS232)	54
5.7.5	Transparent serial port data reporting event.....	55
5.7.6	Unregister serial port to receive events.....	56
5.8	Log configuration	57
5.8.1	Log configuration	57
5.9	Black and white list	58
5.9.1	Note	58
5.9.2	Black and white list operation	59
5.9.2.1	Get a B & W list item.....	59
5.9.2.2	Edit a B & W list item	62
5.9.2.3	Add a B & W list item.....	63
5.9.2.4	Delete a B & W list item	64
5.9.2.5	Find a B & W list item	65
5.9.2.6	Delete all list items.....	67
5.9.3	Get the total number of B & W list items	68
5.9.3.1	Get the total number of B & W list items	68

1. cover

This manual is the API specification for IP Camera SDK.
Please refer to this manual for proper camera
integration.

2. Introduction

SDK

Name of SDK	Abbreviation	version
IP Camera SDK	ice_ipcsdk.jar	V1.4.0.82

Audience

This document is for the following engineers:

- SI developer

3. Revision Record

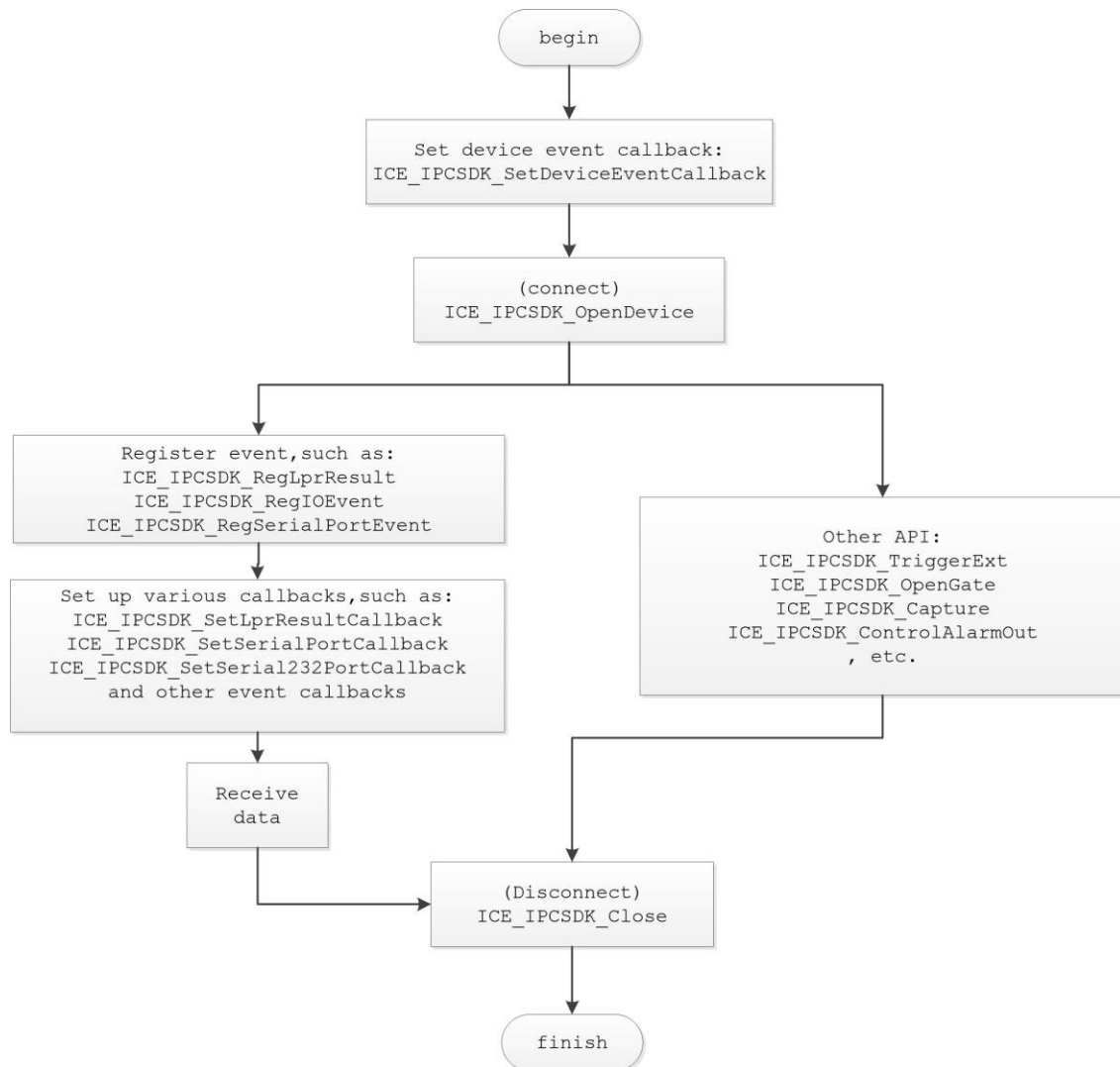
Date	Version	Revision Note
2021-10-09	V1.0.1	1. The first edition

4. programming instructions

4.1 The development environment

Support systems: windows32 or windows 64; Linux
Support languages: java, android

4.2 API call process



4.3 Note

When using the SDK, in addition to introducing ice_ In addition to ipcsdk.jar, please import the related library zip4j_synchronously_1.3.2.jar, the library on which JSON parsing depends (fastjson-1.2.70.Jar) and java-websocket-1.3.8.jar.

5. interface definition

5.1 Connect device

5.1.1 Connect device

5.1.1.1 Connect camera

```
int ICE_IPCSDK_OpenDevice(String strIP)
```

Parameters:

strIP - ip address

Return:

1 succee

0 fail

5.1.1.2 encrypt connection

```
int ICE_IPCSDK_OpenDevice_Passwd(String strIP, String strPasswd,  
String strReserve)
```

Parameters:

strIP - ip address
strPasswd - password
strReserve - Reserved parameter

Return:

1 success
0 fail

5.1.1.3 disconnect

```
void ICE_IPCSDK_Close()
```

Parameters:

void

Return:

void

5.1.2 Device connection status

5.1.2.1 Get device status

```
boolean ICE_IPCSDK_GetStatus()
```

Parameters:

void

Return:

false - offline

true - online

5.1.2.2 Set device event callback

```
public void ICE_IPCSDK_SetDeviceEventCallBack(IDeviceEventCallback  
callback)
```

Parameters:

callback - device event callback, see [IDeviceEventCallback](#) (See 5.1.2.3)

Return:

void

5.1.2.3 device event callback

```
public interface IDeviceEventCallback {  
    public void ICE_IPCSDK_OnDeviceEvent(  
        String strIP,  
        int nEventType,  
        int nEventData1,  
        int nEventData2,  
        int nEventData3,  
        int nEventData4  
    );  
}
```

Parameters:

strIP - camera's ip

nEventType - Event type 0: offline 1: online

nEventData1 - reserve parameter

nEventData2 - reserve parameter

nEventData3 - reserve parameter

nEventData4 - reserve parameter

5.2 Plate recognition

5.2.1 Software trigger

5.2.1.1 Software trigger

```
int ICE_IPCSDK_TriggerExt()
```

Parameters:

void

Return:

0 - fail

1 - success

5.2.2 recognition data reception

5.2.2.1 Register callback for license plate recognition event

```
int ICE_IPCSDK_RegLprEvent(boolean bNeedPic)
```

Parameters:

bNeedPic - Whether need picture

Return:

0 - fail

1 - success

5.2.2.2 Set callback for license plate recognition event

```
void ICE_IPCSDK_SetLprResultCallback(ILprResultCallback callback)
```

Parameters:

callback - plate recognition data callback, see [ILprResultCallback](#)

Return:

void

5.2.2.3 License plate recognition data callback function

```
public interface ILprResultCallback {  
    public void ICE_IPCSDK_LprResult(  
        String strIP,  
        LprResult lprResult,  
        byte[] bData,  
        int iFullPicOffset,  
        int iFullPicLen,  
        int iPlatePicOffset,  
        int iPlatePicLen,  
        int iReserve1,  
        int iReserve2  
    );  
}
```

Parameters:

strIP - camera's ip
lprResult - recognition result
bData - data of picture
iFullPicOffset - Offset of full picture data
iFullPicLen - length of full picture data
iPlatePicOffset - Offset of plate picture data
iPlatePicLen - length of plate picture data
iReserve1 - reserved parameter
iReserve2 - reserved parameter

LprResult:

```
public class LprResult {  
    //plateInfo  
    private String strPlateNum = ""; // plate number  
    private String strPlateColor = ""; // License plate color  
    private boolean bFalsePlate = false; // Whether it is a fake license  
plate  
    private float fConfidence = 0; // Scoring value  
    private int iPlateType = 0; // License plate type  
    private int iRectLeft = 0; // Left X coordinate of license plate  
rectangle  
    private int iRectTop = 0; // Left Y coordinate of license plate  
rectangle  
    private int iRectRight = 0; // Right X coordinate of license plate  
rectangle
```

```
private int iRectBottom = 0; // Right Y coordinate of license plate rectangle
```

```
//vehicleInfo
```

```
private String strVehicleType = ""; // Vehicle Type
```

```
private String strVehicleColor = ""; // the color of car
```

```
private String strVehicleDir = ""; // Vehicle direction
```

```
private String strLogName = ""; // Brand
```

```
private String strSubLogName = ""; // Sub-brand
```

```
private String strProductYearName = ""; // Year of production
```

```
private String strLprState = ""; // Result status (online real-time result / disconnected network resume result)
```

```
private int iHadPlate = 0; // whether had plate
```

```
private String strTriggerType = ""; // Trigger type
```

```
private String strPlateState = ""; // License plate status
```

```
private String strBwList = ""; // Black and white list
```

```
private String strCapTime = ""; // Capture time
```

```
private int iFullPicLen = 0;
```

```
private int iPlatePicLen = 0;
```

```
public String getStrPlateNum() {
```

```
    return strPlateNum;
```

```
}
```

```
public String getStrPlateColor() {
```

```
    return strPlateColor;
```

```
}
```

```
public boolean isbFalsePlate() {
```

```
    return bFalsePlate;
```

```
}
```

```
public float getfConfidence() {
```

```
    return fConfidence;
```

```
}
```

```
public int getiPlateType() {
```

```
    return iPlateType;
```

```
}
```

```
public int getiRectLeft() {
```

```
    return iRectLeft;
```

```
}
```

```
public int getiRectTop() {
```

```
    return iRectTop;
```

```
}
```

```
public int getiRectRight() {
```

```
    return iRectRight;
```

```
    }
    public int getiRectBottom() {
        return iRectBottom;
    }
    public String getStrVehicleType() {
        return strVehicleType;
    }
    public String getStrVehicleColor() {
        return strVehicleColor;
    }
    public String getStrVehicleDir() {
        return strVehicleDir;
    }
    public String getStrLogName() {
        return strLogName;
    }
    public String getStrSubLogName() {
        return strSubLogName;
    }
    public String getStrProductYearName() {
        return strProductYearName;
    }
    public String getStrLprState() {
        return strLprState;
    }
    public int getiHadPlate() {
        return iHadPlate;
    }
    public String getStrTriggerType() {
        return strTriggerType;
    }
    public String getStrPlateState() {
        return strPlateState;
    }
    public String getStrBwList() {
        return strBwList;
    }
    public String getStrCapTime() {
        return strCapTime;
    }
    public int getiFullPicLen() {
        return iFullPicLen;
    }
    public int getiPlatePicLen() {
```

```
        return iPlatePicLen;
    }
}
```

Return:

void

5.2.2.4 Unregister callback for license plate recognition event

```
int ICE_IPCSDK_UnregLprEvent()
```

Parameters:

void

Return:

0 - fail

1 - success

5.3 real-time video and capture

5.3.1 real-time video

5.3.1.1 mjpeg bitstream callback

```
public interface IMJpegCallback_Static {  
    public void ICE_IPCSDK_MJpeg(  
        String strIP,  
        byte[] bData,  
        int length  
    );  
}
```

Parameters:

strIP - camera's ip
bData - mjpeg data
length - lebgth of mjpeg data

Return:

void

5.3.1.2 set mjpeg bitstream callback

```
public void ICE_IPCSDK_SetMJpegallback_Static(IMJpegCallback_Static  
callback)
```

Parameters:

callback - mjpeg bitstream callback, see [IMJpegCallback_Static](#) (See 5.3.1.1)

Return:

void

5.3.2 capture

5.3.2.1 capture

`CaptureResult ICE_IPCSDK_Capture()`

Parameters:

`void`

Return:

Capture results are defined as follows:

```
public class CaptureResult {  
    public byte[] picdata; //data of capture picture  
}
```

Null on failure

5.4 Device management

5.4.1 Get mac of device

```
public String ICE_IPCSDK_GetDevID()
```

Parameters:

void

Return:

mac of device, null on failure

5.4.2 Set the camera network configuration

```
public int ICE_IPCSDK_SetIPAddr(String strIP, String strMask, String  
strGateway)
```

Parameters:

strIP - camera's ip
strMask - camera's mask
strGateway - camera's gateway

Return:

0 - fail
1 - success

5.4.3 Get the camera network configuration

```
public NetworkParam ICE_IPCSDK_GetIPAddr()
```

Parameters:

void

Return:

network configuration, the definition is as follows:

```
public class NetworkParam {  
    private String ip = ""; //ip  
    private String mask = ""; //mask  
    private String gateWay = ""; //gateway  
    public String getIp() {  
        return ip;  
    }  
    public void setIp(String ip) {  
        this.ip = ip;  
    }  
    public String getMask() {  
        return mask;  
    }  
    public void setMask(String mask) {  
        this.mask = mask;  
    }  
    public String getGateWay() {  
        return gateWay;  
    }  
    public void setGateWay(String gateWay) {  
        this.gateWay = gateWay;  
    }  
}
```

Null on failure

5.4.4 Get UID of camera

```
public String ICE_IPCSDK_GetUID()
```

Parameters:

void

Return:

UID of device, null on failure

5.4.5 Get time of camera

```
public TimeCfgInfo ICE_IPCSDK_GetTime()
```

Parameters:

```
void
```

Return:

null on failure

The time information is returned upon success. The time parameters are defined as follows:

```
public class TimeCfgInfo{
    public int iYear;//year
    public int iMon;//month
    public int iDay;//day
    public int iHour;//hour
    public int iMin;//minute
    public int iSec;//second
}
```

5.4.6 Time synchronization

```
public int ICE_IPCSDK_SyncTime(  
    int sYear,  
    int bMonth,  
    int bDay,  
    int bHour,  
    int bMin,  
    int bSec  
)
```

Parameters:

sYear - year
bMonth - month
bDay - day
bHour - hour
bMin - minute
bSec - second

Return:

0 - fail
1 - success

Note: the camera is UTC time, and this interface needs to pass in UTC time

5.5 Peripherals management

5.5.1 open gate

```
public int ICE_IPCSDK_OpenGate()
```

Parameters:

void

Return:

0 - fail

1 - success

5.5.2 Control alarmout

```
public int ICE_IPCSDK_ControlAlarmOut(int nIndex)
```

Parameters:

nIndex - IO index(0: IO1: IO2: IO3: IO4)

Return:

0 - fail

1 - success

5.5.3 Get IO status

```
public ICE_IOState ICE_IPCSDK_GetIOState()
```

Parameters:

void

Return:

IO status, defined as follows:

```
public class ICE_IOState{  
    public int[] nIOState = new int[4];  
}
```

Array subscript: 0 is IO1 status, 1 is io2 status, 2 is IO3 status, and 3 is IO4 status

Null on failure

5.5.4 Register IO change event

```
int ICE_IPCSDK_RegGpioEvent()
```

Parameters:

void

Return:

0 - fail

1 - suuccess

5.5.5 Set event callback for IO state change

```
void ICE_IPCSDK_SetIOEventCallBack(IIOEventCallback callback)
```

Parameters:

callback - IO state change event, see [IIOEventCallback](#)

Return:

void

5.5.6 IO state change event

```
public interface IIOEventCallback {  
    public void ICE_IPCSDK_OnIOEvent(  
        String strIP,  
        int nEventType,  
        int nEventData1,  
        int nEventData2,  
        int nEventData3,  
        int nEventData4  
    );  
}
```

Parameters:

strIP - camera's ip
nEventType - Event type 0: IO state change
nEventData1 - reserve parameter
nEventData2 - reserve parameter
nEventData3 - reserve parameter
nEventData4 - reserve parameter

Return:

void

5.5.7 Unregister IO change event

```
int ICE_IPCSDK_UnregGpioEvent()
```

Parameters:

void

Return:

0 - fail

1 - success

5.6 Camera configuration

5.6.1 Black and white list parameters

5.6.1.1 Get the black and white list parameters

```
public WhiteListParam ICE_IPCSDK_GetWhiteListParam()
```

Parameters:

void

Return:

Null on failure

Black and white list configuration parameters are defined as follows:

```
public class WhiteListParam {  
    private int nWhiteListMode = 0;    // White list control barrier gate  
mode. 0:work offline, 1:real-time , 2:turn off/don't control barrier  
gate  
    private int nWhiteListMatch = 0;    // the degree of matching level,  
range (60%-100%)  
    private int nBlackListMode = 0;    // Blacklist control gate mode,  
0: don't control barrier gate, 1: open gate  
    private int nBlackListMatch = 0;    // the degree of matching level,  
range (60%-100%)  
    private int nTempListMode = 0;    // Temporary vehicle control  
barrier gate mode. 0: don't control barrier gate, 1: open the gate  
    private int ignoreHZ_flag = 0;    //reserve  
    private int allow_unmatch_chars_cnt = 0; // The number of mismatched  
characters ranges [0~2]  
    private int new_version = 1;    // New or old WhiteList, 1:new,  
0:old.  
    private int Jing_mode = 0;    // reserve  
    private int Army_mode = 0;    // reserve  
    private int NewEnergy_mode = 0;    // reserve  
    private int EmergencyMode = 0;    // reserve  
    private int Antifake_mode = 0;    // reserve  
    public int getnWhiteListMode() {  
        return nWhiteListMode;  
    }  
}
```



```
public void setnWhiteListMode(int nWhiteListMode) {
    this.nWhiteListMode = nWhiteListMode;
}
public int getnWhiteListMatch() {
    return nWhiteListMatch;
}
public void setnWhiteListMatch(int nWhiteListMatch) {
    this.nWhiteListMatch = nWhiteListMatch;
}
public int getnBlackListMode() {
    return nBlackListMode;
}
public void setnBlackListMode(int nBlackListMode) {
    this.nBlackListMode = nBlackListMode;
}
public int getnBlackListMatch() {
    return nBlackListMatch;
}
public void setnBlackListMatch(int nBlackListMatch) {
    this.nBlackListMatch = nBlackListMatch;
}
public int getnTempListMode() {
    return nTempListMode;
}
public void setnTempListMode(int nTempListMode) {
    this.nTempListMode = nTempListMode;
}
public int getIgnoreHZ_flag() {
    return ignoreHZ_flag;
}
public void setIgnoreHZ_flag(int ignoreHZ_flag) {
    this.ignoreHZ_flag = ignoreHZ_flag;
}
public int getAllow_unmatch_chars_cnt() {
    return allow_unmatch_chars_cnt;
}
public void setAllow_unmatch_chars_cnt(int
allow_unmatch_chars_cnt) {
    this.allow_unmatch_chars_cnt = allow_unmatch_chars_cnt;
}
public int getNew_version() {
    return new_version;
}
public void setNew_version(int new_version) {
```

```
        this.new_version = new_version;
    }
    public int getJing_mode() {
        return Jing_mode;
    }
    public void setJing_mode(int jing_mode) {
        Jing_mode = jing_mode;
    }
    public int getArmy_mode() {
        return Army_mode;
    }
    public void setArmy_mode(int army_mode) {
        Army_mode = army_mode;
    }
    public int getNewEnergy_mode() {
        return NewEnergy_mode;
    }
    public void setNewEnergy_mode(int newEnergy_mode) {
        NewEnergy_mode = newEnergy_mode;
    }
    public int getEmergencyMode() {
        return EmergencyMode;
    }
    public void setEmergencyMode(int emergencyMode) {
        EmergencyMode = emergencyMode;
    }
    public int getAntifake_mode() {
        return Antifake_mode;
    }
    public void setAntifake_mode(int antifake_mode) {
        Antifake_mode = antifake_mode;
    }
}
}
```

5.6.1.2 Set the black and white list parameters

```
public int ICE_IPCSDK_SetWhiteListParam(WhiteListParam param)
```

Parameters:

param - Black and white list configuration parameters , defined as follows:

```
public class WhiteListParam {
    private int nWhiteListMode = 0;    // White list control barrier gate
mode. 0:work offline, 1:real-time , 2:turn off/don't control barrier
gate
    private int nWhiteListMatch = 0;    // the degree of matching level,
range (60%-100%)
    private int nBlackListMode = 0;    // Blacklist control gate mode,
0: don't control barrier gate, 1: open gate
    private int nBlackListMatch = 0;    // the degree of matching level,
range (60%-100%)
    private int nTempListMode = 0;    // Temporary vehicle control
barrier gate mode. 0: don't control barrier gate, 1: open the gate
    private int ignoreHZ_flag = 0;    //reserve
    private int allow_unmatch_chars_cnt = 0; // The number of mismatched
characters ranges [0~2]
    private int new_version = 1;    // New or old WhiteList, 1:new,
0:old.
    private int Jing_mode = 0;    // reserve
    private int Army_mode = 0;    // reserve
    private int NewEnergy_mode = 0; // reserve
    private int EmergencyMode = 0; // reserve
    private int Antifake_mode = 0; // reserve
    public int getnWhiteListMode() {
        return nWhiteListMode;
    }
    public void setnWhiteListMode(int nWhiteListMode) {
        this.nWhiteListMode = nWhiteListMode;
    }
    public int getnWhiteListMatch() {
        return nWhiteListMatch;
    }
    public void setnWhiteListMatch(int nWhiteListMatch) {
        this.nWhiteListMatch = nWhiteListMatch;
    }
}
```

```
public int getnBlackListMode() {
    return nBlackListMode;
}

public void setnBlackListMode(int nBlackListMode) {
    this.nBlackListMode = nBlackListMode;
}

public int getnBlackListMatch() {
    return nBlackListMatch;
}

public void setnBlackListMatch(int nBlackListMatch) {
    this.nBlackListMatch = nBlackListMatch;
}

public int getnTempListMode() {
    return nTempListMode;
}

public void setnTempListMode(int nTempListMode) {
    this.nTempListMode = nTempListMode;
}

public int getIgnoreHZ_flag() {
    return ignoreHZ_flag;
}

public void setIgnoreHZ_flag(int ignoreHZ_flag) {
    this.ignoreHZ_flag = ignoreHZ_flag;
}

public int getAllow_unmatch_chars_cnt() {
    return allow_unmatch_chars_cnt;
}

public void setAllow_unmatch_chars_cnt(int
allow_unmatch_chars_cnt) {
    this.allow_unmatch_chars_cnt = allow_unmatch_chars_cnt;
}

public int getNew_version() {
    return new_version;
}

public void setNew_version(int new_version) {
    this.new_version = new_version;
}

public int getJing_mode() {
    return Jing_mode;
}

public void setJing_mode(int jing_mode) {
    Jing_mode = jing_mode;
}

public int getArmy_mode() {
```

```
        return Army_mode;
    }
    public void setArmy_mode(int army_mode) {
        Army_mode = army_mode;
    }
    public int getNewEnergy_mode() {
        return NewEnergy_mode;
    }
    public void setNewEnergy_mode(int newEnergy_mode) {
        NewEnergy_mode = newEnergy_mode;
    }
    public int getEmergencyMode() {
        return EmergencyMode;
    }
    public void setEmergencyMode(int emergencyMode) {
        EmergencyMode = emergencyMode;
    }
    public int getAntifake_mode() {
        return Antifake_mode;
    }
    public void setAntifake_mode(int antifake_mode) {
        Antifake_mode = antifake_mode;
    }
}
```

Return:

0 - fail
1 - success

5.6.2 DNS configuration

5.6.2.1 Set DNS configuration

```
public int ICE_IPCSDK_SetDNSAddr(String strDNS, String strDNSReserve)
```

Parameters:

strDNS - Preferred DNS

strDNSReserve - Alternate DNS

Return:

0 - fail

1 - success

5.6.2.2 Get DNS configuration

```
public DNSParam ICE_IPCSDK_GetDNSAddr()
```

Parameters:

void

Return:

The DNS obtained is defined as follows:

```
public class DNSParam {  
    private String strDNS = ""; // Preferred DNS  
    private String strDNSReserve = ""; // Alternate DNS  
    public String getStrDNS() {  
        return strDNS;  
    }  
    public void setStrDNS(String strDNS) {  
        this.strDNS = strDNS;  
    }  
    public String getStrDNSReserve() {  
        return strDNSReserve;  
    }  
    public void setStrDNSReserve(String strDNSReserve) {  
        this.strDNSReserve = strDNSReserve;  
    }  
}
```

Null is returned when obtaining failed

5.6.3 Alarm IO configuration

5.6.3.1 Get alarm IO configuration

```
public RelayOutputResult ICE_IPCSDK_GetRelayOutput(int nIndex)
```

Parameters:

nIndex - IO index (0:IO1, 1:IO2, 2:IO3, 3:IO4)

Return:

The obtained alarm IO configuration, defined as follows:

```
public class RelayOutputResult{
    public byte bIdleState; // IO state (0:on, 1:off)
    public int nDelayTime; //Time to switch state (-1: no state switch.
unit: second, value: -1,1-10)
}
```

Null is returned when obtaining failed

5.6.3.2 Set alarm IO configuration

```
public int ICE_IPCSDK_SetRelayOutputSettings(  
    byte bIdleState,  
    int nDelayTime,  
    int nIndex  
)
```

Parameters:

bIdleState - IO state (0:on, 1:off)

nDelayTime - Time to switch state (-1: no state switch. unit: second,
value: -1,1-10)

nIndex - IO index (0:IO1, 1:IO2, 2:IO3, 3:IO4)

Return:

0 - fail

1 - success

5.6.4 Serial port configuration

5.6.4.1 Get serial port configuration

```
public UartParam ICE_IPCSDK_GetUARTCfg()
```

Parameters:

void

Return:

Serial port configuration, defined as below:

```
public class Uart{
    // Whether the serial port is enabled.0:disable 1:enable
    public int uartEn;
    public int uartWorkMode;           //reserve
    // baudRate 0:1200,1:2400,2:4800,3:9600,4:19200,5:38400,6:115200
    public int baudRate;
    public int dataBits;               // databit
    // parity check,0:none,1:odd,2:even
    public int parity;
    public int stopBits;              // stopBits,0:1,1:2
    // flow control,0:none,1:hardware,2:Xon,3: Xoff
    public int flowControl;
    public int LEDBusinessType;       // reserve
    //Retransmission times [0~3]
    public int u32UartProcOneReSendCnt;
    //Screen display mode. 1: screen display, 2: passthrough
    public byte screen_mode;
}

public class UartParam{
    public Uart uartParam1 = new Uart(); //RS485-1
    public Uart uartParam2 = new Uart(); //RS485-2/RS232
}
```

Null on failure.

5.6.4.2 Set serial port configuration

```
public int ICE_IPCSDK_SetUARTCfg(UartParam param)
```

Parameters:

param - Serial port configuration, defined as below:

```
public class Uart{
    // Whether the serial port is enabled.0:disable 1:enable
    public int uartEn;
    public int uartWorkMode;           //reserve
    // baudRate 0:1200,1:2400,2:4800,3:9600,4:19200,5:38400,6:115200
    public int baudRate;
    public int dataBits;               // databit
    // parity check,0:none,1:odd,2:even
    public int parity;
    public int stopBits;              // stopBits,0:1,1:2
    // flow control,0:none,1:hardware,2:Xon,3: Xoff
    public int flowControl;
    public int LEDBusinessType;       // reserve
    //Retransmission times [0~3]
    public int u32UartProcOneReSendCnt;
    //Screen display mode. 1: screen display, 2: passthrough
    public byte screen_mode;
}

public class UartParam{
    public Uart uartParam1 = new Uart(); //RS485-1
    public Uart uartParam2 = new Uart(); //RS485-2/RS232
}
```

Return:

0 - fail
1 - success

matters needing attention:

Because the UART class contains private member variables, setting is not supported temporarily,
Please call the acquisition interface first, and then assign a value to the member variable to be set.

5.7 Transparent serial port

5.7.1 Send data through serial port in SerinalNET mode

1. Transparent serial port,RS485-1

```
int ICE_IPCSDK_TransSerialPort(byte[] data)
```

Parameters:

data - data of need send

Return:

0 - fail

1 - success

2. Transparent serial port ,RS232/RS485-2

```
int ICE_IPCSDK_TransSerial232Port(byte[] data)
```

Parameters:

data - data of need send

Return:

0 - fail

1 - success

5.7.2 Register serial port to receive events

```
int ICE_IPCSDK_RegRS485Event(int iRS485Index)
```

Parameters:

iRS485Index - rs485 index, 1:RS485-1, 2:RS485-2, 3: RS485-1 + RS485-2

Return:

0 - fail

1 - success

5.7.3 Set transparent serial port data reporting event callback (RS485-1)

```
void ICE_ICPSDK_SetSerialPortCallback(ISerialPortCallback callback)
```

Parameters:

callback-Transparent serial port data reporting event callback , see [ISerialPortCallback](#) (See 5.7.5)

Return:

void

5.7.4 Set transparent serial port data reporting event callback (RS485-2/RS232)

```
void ICE_ICPSDK_SetSerial232PortCallback(  
    ISerialPortCallback callback)
```

Parameters:

callback-Transparent serial port data reporting event callback , see
[ISerialPortCallback](#) (See 5.7.5)

Return:

void

5.7.5 Transparent serial port data reporting event

```
public interface ISerialPortCallback {  
    public void ICE_IPCSDK_SerialPort(  
        String strIP,  
        byte[] bData,  
        int nOffset,  
        int nLen  
    );  
}
```

Parameters:

strIP - camera's ip
bData - data
nOffset - data offset
nLen - length of data

Return:

void

5.7.6 Unregister serial port to receive events

```
int ICE_IPCSDK_UnregRS485Event()
```

Parameters:

void

Return:

0 - fail

1 - success

5.8 Log configuration

5.8.1 Log configuration

```
public void ICE_IPCSDK_LogConfig(int nEnableLog, String filePath)
```

Parameters:

nEnableLog - Log or not, 0: not log, 1: log

filePath - The path of the log record. The full path of the log is
filepath + ipclog.txt

Return:

void

5.9 Black and white list

5.9.1 Note

1. Whitelist format

Format: license plate number +TAB + start date +TAB + end date +TAB + start time +TAB + end time +TAB + white list type (B/W) +TAB + note + enter

Example: AP889Q 2015/09/21 2015/10/20 00:00:00 23:59:59 w remarks

Note: the start time can only be set to 00:00:00 while the end time can only be set to 23:59:59.

2. The way of getting all black and white list items

A. Call ICE_IPCSDK_WhiteListGetCount to get the total number of the black and white list item

B. for(int i = 0; i < totalNum; i++)

```
{  
    ICE_IPCSDK_WhiteListGetItem; (use i as the second parameter, i.e.  
index number)  
}
```

3. The maximum length for remarks of license plate is 127 bytes

5.9.2 Black and white list operation

5.9.2.1 Get a B & W list item

WhiteListItem ICE_IPCSDK_WhiteListGetItem(int index)

Parameters:

index - index of list

Return:

Black and white list item, defined as follows:

```
public class WhiteListItem {
    private String strPlate = "";    // plate number
    //begin date(format is yyyy/MM/dd)
    private String strDateBegin = "";
    //end date(format is yyyy/MM/dd)
    private String strDateEnd = "";
    //begin time(format is HH:mm:ss)
    private String strTimeBegin = "";
    //end time(format is HH:mm:ss)
    private String strTimeEnd = "";
    //type('W'- white list /'B'- black list)
    private String strType = "";
    private String strRemark = "";    //remark
    private String strRsrv3 = "";
    private String strRsrv4 = "";

    private int iStatus = 0;

    public String getStrPlate() {
        return strPlate;
    }
    public void setStrPlate(String strPlate) {
        this.strPlate = strPlate;
    }
    public String getStrDateBegin() {
        return strDateBegin;
    }
    public void setStrDateBegin(String strDateBegin) {
        this.strDateBegin = strDateBegin;
    }
    public String getStrDateEnd() {
```

```
        return strDateEnd;
    }
    public void setStrDateEnd(String strDateEnd) {
        this.strDateEnd = strDateEnd;
    }
    public String getStrTimeBegin() {
        return strTimeBegin;
    }
    public void setStrTimeBegin(String strTimeBegin) {
        this.strTimeBegin = strTimeBegin;
    }
    public String getStrTimeEnd() {
        return strTimeEnd;
    }
    public void setStrTimeEnd(String strTimeEnd) {
        this.strTimeEnd = strTimeEnd;
    }
    public String getStrType() {
        return strType;
    }
    public void setStrType(String strType) {
        this.strType = strType;
    }
    public String getStrRemark() {
        return strRemark;
    }
    public void setStrRemark(String strRemark) {
        this.strRemark = strRemark;
    }
    public String getStrRsrv3() {
        return strRsrv3;
    }
    public void setStrRsrv3(String strRsrv3) {
        this.strRsrv3 = strRsrv3;
    }
    public String getStrRsrv4() {
        return strRsrv4;
    }
    public void setStrRsrv4(String strRsrv4) {
        this.strRsrv4 = strRsrv4;
    }
    public int getiStatus() {
        return iStatus;
    }
}
```

```
public void setiStatus(int iStatus) {  
    this.iStatus = iStatus;  
}  
}
```

5.9.2.2 Edit a B & W list item

```
public boolean ICE_IPCSDK_WhiteListEditItem_ByNumber(  
    String strPlate,  
    String strDateBegin,  
    String strDateEnd,  
    String strTimeBegin,  
    String strTimeEnd,  
    String strType,  
    String strRemark,  
    String strRsrv1,  
    String strRsrv2  
)
```

Parameters:

strPlate - plate number
strDateBegin - begin date(format is yyyy/MM/dd)
strDateEnd - end date(format is yyyy/MM/dd)
strTimeBegin - begin time(format is HH:mm:ss)
strTimeEnd - end time(format is HH:mm:ss)
strType - type('W':whitelist/'B':blacklist)
strRemark - remark
strRsrv1 - reserve
strRsrv2 - reserve

Return:

false - fail
true - success

5.9.2.3 Add a B & W list item

```
public boolean ICE_IPCSDK_WhiteListInsertItem_ByNumber(  
    String strPlate,  
    String strDateBegin,  
    String strDateEnd,  
    String strTimeBegin,  
    String strTimeEnd,  
    String strType,  
    String strRemark,  
    String strRsrv1,  
    String strRsrv2  
)
```

Parameters:

strPlate - plate number
strDateBegin - begin date(format is yyyy/MM/dd)
strDateEnd - end date(format is yyyy/MM/dd)
strTimeBegin - begin time(format is HH:mm:ss)
strTimeEnd - end time(format is HH:mm:ss)
strType - type('W':whitelist/'B':blacklist)
strRemark - remark
strRsrv1 - reserve
strRsrv2 - reserve

Return:

false - fail
true - success

5.9.2.4 Delete a B & W list item

```
public    boolean    ICE_IPCSDK_WhiteListDeleteItem_ByNumber(String  
strPlate)
```

Parameters:

strPlate - plate number

Return:

false - fail

true - success

5.9.2.5 Find a B & W list item

```
public WhiteListItem ICE_IPCSDK_WhiteListFindItem_ByNumber(String  
strPlate)
```

Parameters:

strPlate - plate number

Return:

Black and white list item, defined as follows:

```
public class WhiteListItem {  
    private String strPlate = "";    // plate number  
    //begin date(format is yyyy/MM/dd)  
    private String strDateBegin = "";  
    //end date(format is yyyy/MM/dd)  
    private String strDateEnd = "";  
    //begin time(format is HH:mm:ss)  
    private String strTimeBegin = "";  
    //end time(format is HH:mm:ss)  
    private String strTimeEnd = "";  
    //type('W' - white list / 'B' - black list)  
    private String strType = "";  
    private String strRemark = "";    //remark  
    private String strRsrv3 = "";  
    private String strRsrv4 = "";  
    private int iStatus = 0;  
  
    public String getStrPlate() {  
        return strPlate;  
    }  
    public void setStrPlate(String strPlate) {  
        this.strPlate = strPlate;  
    }  
    public String getStrDateBegin() {  
        return strDateBegin;  
    }  
    public void setStrDateBegin(String strDateBegin) {  
        this.strDateBegin = strDateBegin;  
    }  
    public String getStrDateEnd() {  
        return strDateEnd;  
    }  
    public void setStrDateEnd(String strDateEnd) {
```

```
        this.strDateEnd = strDateEnd;
    }
    public String getStrTimeBegin() {
        return strTimeBegin;
    }
    public void setStrTimeBegin(String strTimeBegin) {
        this.strTimeBegin = strTimeBegin;
    }
    public String getStrTimeEnd() {
        return strTimeEnd;
    }
    public void setStrTimeEnd(String strTimeEnd) {
        this.strTimeEnd = strTimeEnd;
    }
    public String getStrType() {
        return strType;
    }
    public void setStrType(String strType) {
        this.strType = strType;
    }
    public String getStrRemark() {
        return strRemark;
    }
    public void setStrRemark(String strRemark) {
        this.strRemark = strRemark;
    }
    public String getStrRsrv3() {
        return strRsrv3;
    }
    public void setStrRsrv3(String strRsrv3) {
        this.strRsrv3 = strRsrv3;
    }
    public String getStrRsrv4() {
        return strRsrv4;
    }
    public void setStrRsrv4(String strRsrv4) {
        this.strRsrv4 = strRsrv4;
    }
    public int getiStatus() {
        return iStatus;
    }
    public void setiStatus(int iStatus) {
        this.iStatus = iStatus;
    }
}
```

5.9.2.6 Delete all list items

```
boolean ICE_IPCSDK_WhiteListDeleteAllItems()
```

Parameters:

void

Return:

false - fail

true - success

5.9.3 Get the total number of B & W list items

5.9.3.1 Get the total number of B & W list items

```
int ICE_IPCSDK_WhiteListGetCount()
```

Parameters:

void

Return:

- 1 on failure

Total number of B & W list items