



**Web3.0 development kit**

**Interfaces description**

# Catalog

1.	Overview .....	4
1.1	Introduction .....	4
1.2	Supported devices.....	4
1.3	Running environment.....	4
1.4	Version.....	4
2.	Error code definition .....	5
2.1.	Abnormal event code.....	5
2.2.	Error code.....	5
3.	Function calling sequence.....	7
4.	Function description .....	8
4.1.	Plugin initialization.....	8
4.1.1.	Web plugin initialization (including plugin event registration).....	8
4.1.2.	Embed play plugin.....	9
4.1.3.	Write plugin in web .....	9
4.2.	Get device information .....	10
4.2.1.	Get IP basing on DNS.....	10
4.2.2.	Login device.....	10
4.2.3.	Logout device .....	11
4.2.4.	Get basic information of devices.....	11
4.2.5.	Get analog channels information .....	12
4.2.6.	Get digital channel infomation.....	12
4.2.7.	Get zero channel information .....	13
4.2.8.	Record search.....	14
4.2.9.	Get voice intercom channel .....	15
4.3.	Play and play control .....	16
4.3.1.	Start realplay .....	16
4.3.2.	Start playback.....	16
4.3.3.	Start reverse playback .....	17
4.3.4.	Stop play.....	18
4.3.5.	Single frame.....	18
4.3.6.	Pause .....	19
4.3.7.	Resume Play .....	19
4.3.8.	Slow forward .....	19
4.3.9.	Fast forward.....	19
4.3.10.	Get OSD time.....	20
4.3.11.	Enable sound .....	20
4.3.12.	Disable sound .....	20
4.3.13.	Set volume.....	20

4.3.14.	Capture pictures .....	20
4.3.15.	Split screen .....	21
4.4.	Recording .....	21
4.4.1.	Start recording .....	21
4.4.2.	Stop recording .....	21
4.5.	Download record .....	22
4.5.1.	Start downloading .....	22
4.5.2.	Get the records downloading status .....	22
4.5.3.	Get the records downloading progress .....	22
4.5.4.	Stop downloading records .....	22
4.6.	Voice intercom .....	23
4.6.1.	Start voice intercom .....	23
4.6.2.	Stop voice intercom .....	23
4.7.	PTZ control .....	23
4.7.1.	PTZ direction control .....	23
4.7.2.	Setting preset .....	23
4.7.3.	Calling preset .....	24
4.8.	Enlarging image .....	24
4.8.1.	Enable electronic zoom .....	24
4.8.2.	Disable electronic zoom .....	24
4.8.3.	Enable 3D zoom .....	25
4.8.4.	Disable 3D zoom .....	25
4.8.5.	Full-screen play .....	25
4.9.	Device maintaince .....	25
4.9.1.	Export device configuring parameters .....	25
4.9.2.	Import device configuring parameters .....	25
4.9.3.	Restore the default parameters .....	26
4.9.4.	Restart .....	26
4.9.5.	Start upgrading .....	26
4.9.6.	Get upgrading status .....	27
4.9.7.	Get upgrading progress .....	27
4.9.8.	Stop upgrading .....	27
4.9.9.	Open remote configuration .....	27
4.10.	Plugin information maintaince .....	28
4.10.1.	Plugin version comparison .....	28
4.10.2.	Get the local configuring parameters .....	28
4.10.3.	Set the local configuration of plugins .....	29
4.10.4.	Get playing window status .....	29
4.11.	Others .....	29
4.11.1.	Folder selection and files' path .....	29
4.11.2.	Get the last error code .....	29

# 1. Overview

## 1.1 Introduction

The web 3.0 development kit is developed basing on ActiveX and NPAPI, interfaces are encapsulated in javascript, so interfaces of javascript will be provided for integration. Functions of preview, playback, PTZ control and so on are supported via web. This development kit only can be used for B/S, but not C/S.

## 1.2 Supported devices

Most of devices of Hikvision are supported by this development kit, including DVR, NVR, DVS, network camera, network speed dome, etc., but devices should support PSIA or ISAPI protocol.

## 1.3 Running environment

OS: Windows XP, Windows7, Windows8, Windows8.1

Browser: IE6~IE11, Chrome 8+, Firefox3.5+, Safari5.0.2+, only 32-bit Browser is supported currently.

## 1.4 Version

Version	Description
V1.0.0	<ul style="list-style-type: none"><li>● The development kit of this version supports synchronous / asynchronous, cross-domain CGI command (PSIA / ISAPI), but only some basic commands can be supported, such as basic device information, channel acquisition, PTZ control, etc.</li><li>● Playing mode only support rtsp over tcp and rtsp over udp at present.</li></ul>
V1.0.1	<ul style="list-style-type: none"><li>● Modify the HTTP interaction course and solve the problem that old version devices can't login.</li><li>● Solve the Bug that NVR can't call preset</li></ul>
V1.0.2	<ul style="list-style-type: none"><li>● Solve the problem that RTSP port can't be obtained for some devices</li></ul>
V1.0.3	<ul style="list-style-type: none"><li>● Getting stream of private protocol is added, including preview, playback and reverse playback. The private protocol will be used in tcp mode by default.</li></ul>
V1.0.4	<ul style="list-style-type: none"><li>● Redesign PTZ operation interface (I_PTZControl), and focus, zoom, aperture</li></ul>

	functions are added. The flag of start and stop is also added in this interface.
V1.0.5	<ul style="list-style-type: none"> <li>● Remote configuration library is modified as modeless way(I_RemoteConfig) to avoid jamming when using firefox.</li> <li>● Extend remote configuration interface(I_RemoteConfig), add language choice(Chinese and English).</li> <li>● Add Digest Authentication</li> <li>● Solve the problem of 3D zoom</li> </ul>

## 2. Error code definition

### 2.1. Abnormal event code

Abnormal event callback is disposed in the call back function that users input, the first parameter is event code (abnormal playback, playback stop and not enough hard disk space), and the second parameter is the window number of events.

Event Name	Code	
PLUGIN_EVENTTYPE_PLAYABNORMAL	0	abnormal playback
PLUGIN_EVENTTYPE_PLAYBACKSTOP	2	playback stop
PLUGIN_EVENTTYPE_AUDIOTALKFAIL	3	voice intercom failure
PLUGIN_EVENTTYPE_NOFREESPACE	21	not enough hard disk space(record)

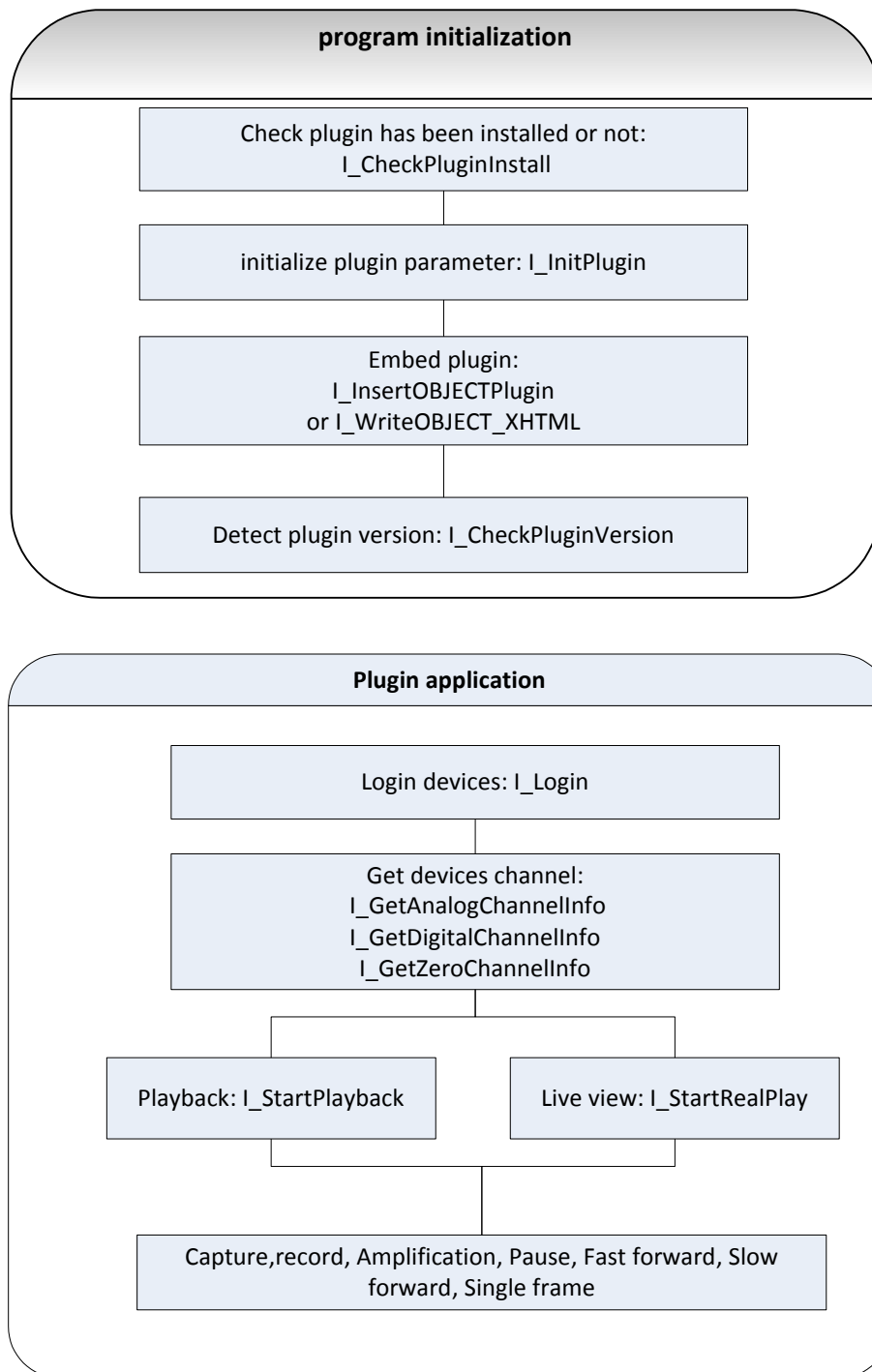
### 2.2. Error code

Error code is obtained by calling I\_GetLastError, it belongs to the bottom error code. The upper logical errors have no error code.

PLUGIN_ERROR_NOERROR	0	no error
PLUGIN_ERROR_LOAD_RTSP_FAILED	1	failed to load rtsp library
PLUGIN_ERROR_LOAD_PLAYCTRL_FAILED	2	failed to load play library
PLUGIN_ERROR_LOAD_SYSTRANSFORM_FAILED	3	failed to load code stream transform wrapping library
PLUGIN_ERROR_LOAD_HTTPCLIENT_FAILED	4	failed to load http library
PLUGIN_ERROR_PARAMETER_ERROR	5	parameter error
PLUGIN_ERROR_ORDER_ERROR	6	Calling sequence error
PLUGIN_ERROR_ALLOC_RESOURCE_FAILED	7	failed to allocate resources
PLUGIN_ERROR_NOT_INITLIB	8	not initialized
PLUGIN_ERROR_OPERTION_NOSUPPORT	9	operation is not supported
PLUGIN_ERROR_OPENFILE_ERROR	10	failed to open files

PLUGIN_ERROR_WRITEFILE_ERROR	11	failed to write files
PLUGIN_ERROR_READFILE_ERROR	12	failed to read files
PLUGIN_ERROR_INIT_HPR_FAILED	13	Failed to initialize hpr library
PLUGIN_ERROR_AUDIO_MONOPOLIZED	14	sound card is monopolized
PLUGIN_ERROR_CREATE_SOCKET_ERROR	15	failed to create socket
PLUGIN_ERROR_NETWORK_CONNECT_FAILED	16	connection failure
PLUGIN_ERROR_NETWORK_SEND_ERROR	17	failed to send
PLUGIN_ERROR_NETWORK_RECV_ERROR	18	failed to receive
PLUGIN_ERROR_NETWORK_SEND_TIMEOUT	19	send timeout
PLUGIN_ERROR_NETWORK_RECV_TIMEOUT	20	receive timeout
PLUGIN_ERROR_NETWORK_RESOLVE_FAILED	21	domain name resolution error
PLUGIN_ERROR_XML_PARSE_ERROR	22	xml parsing error
PLUGIN_ERROR_XML_NODE_ERROR	23	xml node error
PLUGIN_ERROR_NO_EXCEL_DRIVER_ERROR	24	Excel driver is not installed
PLUGIN_ERROR_PARSE_URL_FAILED	25	URL parsing failure
PLUGIN_ERROR_LOADRTSPSDKPROC_ERROR	26	can't find rtsp interface address
PLUGIN_ERROR_LOADPLAYERSDKPROC_ERROR	27	can't find play library interface address
PLUGIN_ERROR_LOADSYSTRANSFORMPROC_ERROR	28	can't find code stream transform wrapping library interface address
PLUGIN_ERROR_LOADHTTPSDKPROC_ERROR	29	can't find http library interface address
PLUGIN_ERROR_START_WAVEIN_FAILED	30	failed to start audio capture
PLUGIN_ERROR_START_WAVEOUT_FAILED	31	failed to start playing audio
PLUGIN_ERROR_INIT_G722_CODEC_FAILED	32	failed to initialize G722 codec
PLUGIN_ERROR_NOT_ENOUGH_DISK_FREESPACE	33	not enough hard disk
PLUGIN_ERROR_FILE_ALREADY_EXIST	34	file has already existed

### 3. Function calling sequence



## 4. Function description

### 4.1. Plugin initialization

#### 4.1.1. Web plugin initialization (including plugin event registration)

Function: `I_InitPlugin (szWidth, szHight, options)`

Instruction: initialize various properties of plugin

Parameter: `szWidth` plugin width (unit: px, 100% represents full occupy plugin container)

`szHight` plugin height (unit: px, 100% represents full occupy plugin container )

`options` Optional parameter objects:

`szContainerID` the container ID of plugin (the DOM node of HTML ), which can be input both when initialization and and when plugin is being embedded.

`szColorProperty` the color properties, which represents background color of plugin, background color of child window, child window border color, selected border color of child window frame. Plugin has its own default color.

`szOcxClassId` ocx plugin ID, corresponding ID can be modified when OEM to achieve development kit to bind different plugins, and default plug-in is Hikvision WEB3.0.

`szMimeType` non-IE plugin, corresponding ID can be modified when OEM to achieve development kit to bind different plugins, and default plug-in is Hikvision WEB3.0.

`iWndowType` split screen types, 1(1\*1), 2(2\*2), 3(3\*3), 4(4\*4). The default type is a single screen.

`iPlayMode` play mode, the default value is 2: normal play mode. Other modes are not supported at present.

`bDebugMode` JS debug mode, console prints debug information: true(enable), false(disable)

`cbSelWnd` window selects event callback function,



	contains only one string parameter, the value of which is XML.
cbEvent	plugin event callback function, has three parameters: the first is event type, and the second is window number.

Return value: none

Note: the format of szColorProperty: "plugin-background:ffffff; sub-background:ffffff; sub-border:ffffff; sub-border-select:ffffff", which represents background color of plugin, background color of child window, child window border color, selected border color of child window frame.

cbSelWnd is window selects event callback function, users can input function, and development kit will automatically call this function after the window is selected. The parameter is XML, format is as follows:

```
<?xml version="1.0"?>
<RealPlayInfo>
<SelectWnd>0</SelectWnd>//the number of window that triggers event, start from 0
</RealPlayInfo>
```

cbEvent is a callback function of plugin's abnormal event, which has three parameters: the first parameter is event type (each value of abnormal events is introduced in **Abnormal event code**), the second represents the number of window that triggers events.

### 4.1.2. Embed play plugin

Function: I\_InsertOBJECTPlugin (szContainerID)

Instruction: embed play plugin in HTML DOM

Parameter: szContainerID the container ID of plugin, which is the DOM of HTML

Return value: 0-success, -1-failure

### 4.1.3. Write plugin in web

Function: I\_WriteOBJECT\_XHTML ()

Instruction: Playing plug is inserted in web directly

Parameter: none

Return value: 0-success, -1-failure

## 4.2. Get device information

### 4.2.1. Get IP basing on DNS

Function: `I_GetIPInfoByMode (iMode, szAddress, iPort, szDeviceInfo)`

Instruction: Get IP basing on DNS

Parameter: `iMode` DNS server mode, 0-IP\_Domain 1-IPServer 2-HIDDNS

`szAddress` DNS server IP

`iPort` DNS server port

`szDeviceInfo` device serial number or device name(or HiDDNS)

Return value: success: return "device IP address-device SDK port"("-" is used to serve as separator between IP and port); failure: return ""(null string). `I_Login` is called after getting the IP address of device.

### 4.2.2. Login device

Function: `I_Login (szIP, iPrototocol, iPort, szUserName, szPassword, options)`

Instruction: login device

Parameter: `szIP` device IP address

`iPrototocol` 1: http protocol, 2: https protocol

`iPort` login the http/https port of devices, choose different ports according to `iPrototocol`

`szUserName` username

`szPassword` password

`options` optional parameter objects:

`async` http interactive way, true: asynchronous, false: synchronous

`cgi` CGI protocol, 1:ISAPI, 2:PSIA. If this parameter is not input, a kind of protocol that devices support will be chosen.

`success` success callback function, there is one parameter that represents the content of XML.

`error` failure callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: login devices by calling this function; if login successfully, http/https protocol and PSIA/ISAPI protocol will be selected, and the following interaction with devices will adopt the selected protocol. The successful callback function will be called when interact successfully; otherwise, failed callback function will be called.

### 4.2.3. Logout device

Function: I\_Logout (szIP)  
Instruction: Logout device  
Parameters: szIP device IP address  
Return value: 0-success, -1-failure

### 4.2.4. Get basic information of devices

Function: I\_GetDeviceInfo (szIP, options)  
Instruction: get basic information of devices  
Parameters: szIP device IP address  
options optional parameter objects:  
    async http interactive way, true: asynchronous, false: synchronous  
    success successful callback function, there is one parameter that represents the content of XML.  
    error failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise, failed callback function will be called. The first parameter of callback function is the XML of devices information.

XML format is as follows:

```
<DeviceInfo>
  <deviceName></deviceName> //device name
  <deviceId></deviceId> //device ID
  <deviceType></deviceType> //device type(may be null)
  <model></model> //device model
  <serialNumber></serialNumber> //device serial number
  <macAddress></macAddress> //device mac address
  <firmwareVersion></firmwareVersion> //device firmware version
  <firmwareReleasedDate></firmwareReleasedDate> //release date of firmware
  <encoderVersion></encoderVersion> //encoder version
  <encoderReleasedDate></encoderReleasedDate> //release date of encoder
</DeviceInfo>
```

&lt;/DeviceInfo&gt;

#### 4.2.5. Get analog channels information

Function: `I_GetAnalogChannelInfo (szIP, options)`

Instruction: get analog channels information

Parameter:	szIP	device IP address
	options	optional parameter objects:
	async	http interactive way, true:asynchronous , false:synchronous
	success	successful callback function, there is one parameter that represents the content of XML.
	error	failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise, failed callback function will be called. The first parameter of callback function is the XML of devices information.

XML format is as follows:

<VideoInputChannelList>

## <VideoInputChannel>

<id></id> //channel ID

```
<inputPort></inputPort> //channel number
```

```
<videoInputEnabled></videoInputEnabled> //whether to enable
```

<name></name> //channel name

```
<videoFormat></videoFormat> //channel format
```

&lt;/VideoInputChannel&gt;

&lt;/VideoInputChannelList&gt;

#### 4.2.6. Get digital channel information

Function: I GetDigitalChannelInfo (szIP, options)

Instruction: get digital channel information

Parameter:	szIP	device IP address
	options	optional parameter objects:
	async	http interactive way, true: asynchronous, false: synchronous
	success	successful callback function, there is one

	parameter that represents the content of XML.
error	failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise, failed callback function will be called. The first parameter of callback function is the XML of devices information.

XML format is as follows:

```
<InputProxyChannelStatusList>
  <InputProxyChannelStatus>
    <id></id>    //channel ID
    <sourceInputPortDescriptor>
      <proxyProtocol></proxyProtocol>    //connection protocol
      <addressingFormatType></addressingFormatType>    //IP address format type
      <ipAddress></ipAddress>    //IP address
      <managePortNo></managePortNo>    //managing port number
      <srcInputPort></srcInputPort>    //IP channel number
      <userName></userName>    //connected user name
      <streamType></streamType> //code stream type
      <online></online> //online or not (true/false)
    </sourceInputPortDescriptor>
  </InputProxyChannelStatus>
</InputProxyChannelStatusList>
```

## 4.2.7. Get zero channel information

Function: I\_GetZeroChannelInfo (szIP, options)

Instruction: get zero channel information

Parameters: szIP	device IP address
options	optional parameter objects:
async	http interactive way, true:asynchronous , false:synchronous
success	successful callback function, there is one parameter that represents the content of XML.
error	failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise,

failed callback function will be called. The first parameter of callback function is the XML of devices information.

XML format is as follows:

```
<ZeroVideoChannelList>
  <ZeroVideoChannel>
    <id>1</id>    //channel ID
    <enabled>true</enabled>    //whether to enable
    <inputPort>1</inputPort> //input port
  </ZeroVideoChannel>
</ZeroVideoChannelList>
```

## 4.2.8. Record search

Function: `I_RecordSearch (szIP, iChannelID, szStartTime, szEndTime, options)`

Instruction: record search

Parameters:	szIP	device IP address
	iChannelID	channel ID
	szStartTime	start time, eg: 2013-12-23 00:00:00
	szEndTime	end time, eg:2013-12-23 23:59:59
	options	optional parameter objects:
	async	http interactive way, true:asynchronous, false:synchronous
	iSearchPos	search video location(the default value is 0),0 represents the 0-40 returned results, 40 represents 40-80 returned results, and so on.
	success	successful callback function, there is one parameter that represents the content of XML.
	error	failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise, failed callback function will be called. The first parameter of callback function is the XML of devices information.

The maximum number of returned results is 40, and if the number is over 40, you need to call this interface many times, and set up a search location.

XML format is as follows:

```
<CMSearchResult>
  <responseStatus>true</responseStatus>
  <responseStatusStrg>MORE</responseStatusStrg> // decide to search or not according to
```

this status flag. OK stands for search is finished.

```
<numOfMatches>40</numOfMatches> // The number of videos returned this search
<matchList>
  <searchMatchItem>
    <trackID>101</trackID> //record ID
    <startTime>2013-12-23T03:06:58Z</startTime> //the start time of record
    <endTime>2013-12-23T03:16:57Z</endTime> //the end time of record

    <playbackURI>rtsp://172.9.4.222/Streaming/tracks/101/?starttime=20131223T03
    0658Z&endtime=20131223T031657Z&name=02000000076000101&am
    p;size=115665012</playbackURI> // This node contains record start time, end time,
    video name, video size and other information, you need to input this value when
    download video.
    <metadataDescriptor>motion</metadataDescriptor> //record type: timing-timing
    record, motion-motion detection record, motionOrAlarm- motion detection or
    alarm, motionAndAlarm- motion detection and alarm, manual-manual recording,
    smart- intelligent
  </searchMatchItem>
</matchList>
</CMSearchResult>
```

## 4.2.9. Get voice intercom channel

Function: I\_GetAudioInfo (szIP, options)

Instruction: get voice intercom channel information

Parameters:	szIP	device IP address
	options	optional parameter objects:
	async	http interactive way, true: asynchronous, false: synchronous
	success	successful callback function, there is one parameter that represents the content of XML.
	error	failed callback function, there are two parameters, the first is http status code, and the second is the XML that devices return(may be null)

Return value: none

Note: The successful callback function will be called when interact successfully; otherwise, failed callback function will be called. The first parameter of callback function is the XML of devices information.

XML format is as follows:

```
<TwoWayAudioChannelList>
  <TwoWayAudioChannel>
    <id></id> //Channel ID
```

```

        <enabled></enabled> //whether to enable
        <audioCompressionType></audioCompressionType> //audio codec
    </TwoWayAudioChannel>
</TwoWayAudioChannelList>

```

## 4.3. Play and play control

### 4.3.1. Start realplay

Function: I\_StartRealPlay (szIP, options)

Instruction: start realplay

Parameters: szIP	device IP address
options	optional parameter objects:
iWndIndex	Play window, if you do not input this parameter, current window will be chosen by default (the default window is 0)
iStreamType	code stream type: 1-main stream, 2-sub stream, main stream is adopted to preview by default.
iChannelID	play channel number: the default channel is 1.
bZeroChannel	whether to play zero channel, and the default is false.
iPort	RTSP port number: if you do not input this parameter, the RTSP port will be detected automatically by development kit.

Return value: 0-success, -1-failure

Note: this function can't be called until you login.

### 4.3.2. Start playback

Function: I\_StartPlayback (szIP, options)

Instruction: start playback

Parameters: szIP	device IP address
options	optional parameter objects:
iWndIndex	Play window, if you do not input this parameter, current window will be chosen by default (the default window is 0)



szStartTime start time; the default time is the very day 00:00:00, format is: 2013-12-23 00:00:00

szEndTime end time, the default time is the very day 23:59:59, format is: 2013-12-23 23:59:59

iChannelID play channel number, the default channel is 1.

iPort RTSP port number: if you do not input this parameter, the RTSP port will be detected automatically by development kit.

oTransCodeParam Transcoding playback parameters object, if this parameter is input, transcoding playback will be executed according to this object(transcoding playback need support by device, if not support, this parameter doesn't need to be input )

Return value: 0-success, -1-failure

Note: this interface is for playback by time. Playback by time is supported by development kit, but playback by files is not supported at present. You can search out the video, and then playback by start time and end time.

oTransCodeParam is a javascript object:

```
{
  TransFrameRate: "16",
  TransResolution: "2",
  TransBitrate: "23"
}
```

**TransFrameRate: Frame rate**

Value range: 0-all, 5-1, 6-2, 7-4, 8-6, 9-8, 10-10, 11-12, 12-16, 13-20, 14-15, 15-18, 16—22, 255-Auto

**TransResolution: Resolution**

Value range: 1-CIF(352\*288/352\*240) , 2-QCIF(176\*144/176\*120) , 3-4CIF(704\*576/704\*480) or D1(720\*576/720\*486) , 255-Auto(Using current resolution)

**TransBitrate: Bit rate**

Value range: 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K, 24-3072K, 25-4096K, 26-8192K, 255-Auto

### 4.3.3. Start reverse playback

Function: I\_ReversePlayback (szIP, options)

Instruction: start reverse playback

Parameters: szIP                      device IP address

options                      optional parameters objects:

iWndIndex	Play window number, if you do not input this parameter, current window will be chosen by default (the default window is 0)
szStartTime	start time; the default time is the very day 00:00:00, format is: 2013-12-23 00:00:00
szEndTime	end time, the default time is the very day 23:59:59, format is: 2013-12-23 23:59:59
iChannelID	play channel number, the default channel is 1.
iPort	RTSP port number: if you do not input this parameter, the RTSP port will be detected automatically by development kit.

Return value: 0-success, -1-failure

Note: reverse playback starts from end time. Many devices can't support reverse playback at present, if the interface is called, it will return failure.

#### 4.3.4. Stop play

Function: I\_Stop (iWndIndex)

Instruction: stop play (stop realplay and stop playback will call this function):

Parameters: iWndIndex    play window number: it is optional to input or not input this parameter. It represents the current selected window.

Return value: 0-success, -1-failure

#### 4.3.5. Single frame

Function: I\_Frame (iWndIndex)

Instruction: single-frame play: one frame will be played when this function is called one time. It can be called when playback and reverse playback.

Parameters: iWndIndex    Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### 4.3.6. Pause

Function: I\_Pause (iWndIndex)

Instruction: Pause: it can be called when playback and reverse playback.

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### 4.3.7. Resume Play

Function: I\_Resume (iWndIndex)

Instruction: Resume Play: resume playing status from the single-frame / Pause to normal playback

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### 4.3.8. Slow forward

Function: I\_PlaySlow (iWndIndex)

Instruction: Slow forward: the playing speed will reduce one level when you call this interface one time, this plugin supports maximum 1/8 speed, and the device itself may also have restrictions.

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### 4.3.9. Fast forward

Function: I\_PlayFast (iWndIndex)

Instruction: Fast forward, the playing speed will increase one level when you call this interface one time, this plugin supports maximum 8x speed, and the device itself may also have restrictions.

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.3.10. Get OSD time**

Function: I\_GetOSDTime (iWndIndex)

Instruction: Get the OSD time of current code stream to realize playback progress

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.3.11. Enable sound**

Function: I\_OpenSound (iWndIndex)

Instruction: enable sound

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.3.12. Disable sound**

Function: I\_CloseSound (iWndIndex)

Instruction: disable sound

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.3.13. Set volume**

Function: I\_SetVolume (iVolume, iWndIndex)

Instruction: Set the volume, the volume range :0-100

Parameters: iVolume volume  
iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.3.14. Capture pictures**

Function: I\_CapturePic (szPicName, iWndIndex)

Instruction: Capture preview / playback picture, and save it to the local PC, the

storage path will be displayed in local configuration

Parameters: szPicName: file name of picture

iWndIndexPlay window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

Note: The format of captured picture is related to the file name that input when call interface. If the suffix is with '.bmp', then capture bmp picture, and if there is no suffix, the format will be jpg. You can call I\_GetLocalCfg() to get the saving path of picture.

### **4.3.15. Split screen**

Function: I\_ChangeWndNum (iWndType)

Instruction: modify the type of split screen

Parameters: iWndType the type includes:1-1\*1, 2-2\*2, 3-3\*3, 4-4\*4

Return value: 0-success, -1-failure

## **4.4. Recording**

### **4.4.1. Start recording**

Function: I\_StartRecord (szFileName, iWndIndex)

Instruction: save video to the local PC, and the storage path will be displayed in local configuration

Parameters: szFileName recording files name

iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.4.2. Stop recording**

Function: I\_StopRecord (iWndIndex)

Instruction: stop recording

Parameters: iWndIndex Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

## 4.5. Download record

### 4.5.1. Start downloading

Function: I\_StartDownloadRecord (szIP, szPlaybackURI, szFileName)

Instruction: The records stored in devices will be downloaded when this function is called.

Parameters: szIP                      device IP address  
              szPlaybackURI   Record URL, this URL can be obtained by searching record.  
              szFileName        the name of files that need downloading

Return value: It will return a downloading ID that greater than or equal to 0 when success; otherwise, return -1.

### 4.5.2. Get the records downloading status

Function: I\_GetDownloadStatus (iDownloadID)

Instruction: Get the records downloading status to determine whether the download is in progress.

Parameters: iDownloadID   Download ID: the return value of 'Start downloading'.

Return value: it returns 0(stands for downloading is in progress) when success; otherwise, it returns -1 (stands for download failure)

### 4.5.3. Get the records downloading progress

Function: I\_GetDownloadProgress (iDownloadID)

Instruction: Get the downloading progress

Parameters: iDownloadID   download ID: the return value of '**Start downloading**'.

Return value: it will return a downloading progress value that greater than or equal to 0 when success; otherwise, it will return -1.

### 4.5.4. Stop downloading records

Function: I\_StopDownloadRecord (iDownloadID)

Instruction: stop downloading record

Parameters: iDownloadID   download ID: the return value of 'Start downloading'.

Return value: 0-success, -1-failure

## 4.6. Voice intercom

### 4.6.1. Start voice intercom

Function: `I_StartVoiceTalk (szIP, iAudioChannel)`

Instruction: start voice intercom

Parameters: `szIP` device IP address  
`iAudioChannel` voice intercom channel

Return value: 0-success, -1-failure

### 4.6.2. Stop voice intercom

Function: `I_StopVoiceTalk ()`

Instruction: Stop voice intercom

Parameters: none

Return value: 0-success, -1-failure

## 4.7. PTZ control

### 4.7.1. PTZ direction control

Function: `I_PTZControl(iPTZIndex, bStop, options)`

Instruction: PTZ direction control

Parameters: `iPTZIndex` operation type(1-up, 2-down, 3-left, 4-right, 5:up-left, 6:down-left, 7:up-right, 8:down-right, 9-auto, 10-zoom+,11-zoom-,12-focus+,13-focus-,14-aperture+,15-aperture-)

`bStop` Whether to stop the operation of `iPTZIndex`

`options` optional parameter objects

`iWndIndex` window number, the default window is the selected by default.

`iPTZSpeed` PTZ speed, the default is 4

Return value: 0-success, -1-failure

### 4.7.2. Setting preset

Function: `I_SetPreset (iPresetID, options)`

Instruction: Set preset

Parameters: iPresetID      Preset ID  
                 options      optional parameter objects  
                 iWndIndex   window number, the default window is the  
                                         selected by default

Return value: 0-success, -1-failure

### 4.7.3. Calling preset

Function:    I\_GoPreset (iPresetID, options)

Instruction:

Parameters: iPresetID      preset ID  
                 options      optional parameter objects  
                 iWndIndex   window number, the default window is the  
                                         selected by default

Return value: 0-success, -1-failure

## 4.8. Enlarging image

### 4.8.1. Enable electronic zoom

Function:    I\_EnableEZoom (iWndIndex)

Instruction: enable electronic zoom

Parameters: iWndIndex    Play window number: if you do not input this parameter,  
                                         current window will be chosen by default.

Return value: 0-success, -1-failure

### 4.8.2. Disable electronic zoom

Function:    I\_DisableEZoom (iWndIndex)

Instruction: disable electronic zoom

Parameters: iWndIndex    Play window number: if you do not input this parameter,  
                                         current window will be chosen by default.

Return value: 0-success, -1-failure



### **4.8.3. Enable 3D zoom**

Function: I\_Enable3DZoom (iWndIndex)

Instruction: enable 3D zoom

Parameters: iWndIndex    Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.8.4. Disable 3D zoom**

Function: I\_Disable3DZoom (iWndIndex)

Instruction: disable 3D zoom

Parameters: iWndIndex    Play window number: if you do not input this parameter, current window will be chosen by default.

Return value: 0-success, -1-failure

### **4.8.5. Full-screen play**

Function: I\_FullScreen (bFull)

Instruction: full-screen play

Parameters: bFull    full screen or not: true-full screen, false-quit full screen

Return value: none

## **4.9. Device maintainece**

### **4.9.1. Export device configuring parameters**

Function: I\_ExportDeviceConfig (szIP)

Instruction: export the configuring parameters of devices: this interface will automatically pop up the path selection box

Parameters: szIP    device IP address

Return value: 0-success, -1-failure

### **4.9.2. Import device configuring parameters**

Function: I\_ImportDeviceConfig (szIP)

Instruction: import the configuring parameters: this interface will automatically pop up file selection box. After you import the configuring parameters, devices may restart.

Parameters: szIP device IP address

Return value: 0-success, -1-failure

### 4.9.3. Restore the default parameters

Function: I\_RestoreDefault(szIP, szMode, options)

Instruction: Restore the default parameters

Parameters: szIP device IP address

szMode restoring type basic-simply restore, full-fully restore

options optional parameter objects

success: success function: it has one parameter that represents the contents of the returned XML.

Error: failure function: it has two parameters, and the first one is http status code, the second is the returned XML of devices (may be null).

Return value: 0-success, -1-failure

Note: devices need rebooting after restore the default parameters. All users' information will be restored to default if you choose fully restore.

### 4.9.4. Restart

Function: I\_Restart (szIP, options)

Instruction: restart

Parameters: szIP device IP address

options optional parameter objects

success: success function: it has one parameter that represents the contents of the returned XML.

Error: failure function: it has two parameters, and the first one is http status code, the second is the returned XML of devices (may be null).

Return value: none

Note: success only stands for the device is restarted.

### 4.9.5. Start upgrading

Function: I\_StartUpgrade (szIP, szFileName)

Instruction: Start upgrading; the device will restart after upgrade

Parameters: szIP                device IP address  
              szFileName   the path of upgrading files  
Return value: 0-success, -1-failure

#### **4.9.6. Get upgrading status**

Function:    I\_UpgradeStatus ()  
Instruction:  Get upgrading status to determine whether upgrading is in progress.  
Parameters: none  
Return value: It returns 0 when success, which represents upgrade is in progress. Otherwise, it returns -1, which stands for upgrade failure.

#### **4.9.7. Get upgrading progress**

Function:    I\_UpgradeProgress ()  
Instruction:  Get upgrading progress  
Parameters: none  
Return value: It will return upgrade ID that greater than or equal to 0 when success; otherwise, return -1.

#### **4.9.8. Stop upgrading**

Function:    I\_StopUpgrade ()  
Instruction:  stop upgrading  
Parameters: none  
Return value: 0-success, -1-failure

#### **4.9.9. Open remote configuration**

Function:    I\_RemoteConfig (szIP , options)  
Instruction:  open remote configuration  
Parameters: szIP            device IP address  
              options   optional parameter objects:  
                              iDevicePort: the SDK port of devices, if you do not input this parameter, the plugin will obtain it automatically from devices.  
                              iLan:            Configure lib language remotely, 0: English, 1: Chinese. The default is English.  
Return value: 0-success, -1-failure

## 4.10. Plugin information maintenance

### 4.10.1. Plugin version comparison

Function: I\_CheckPluginVersion ()  
Instruction: Plugin version comparison: it can detect whether the plugin is installed before the plug-in is embedded  
Parameters: none  
Return value: -2: no plugins -1: need upgrading 0: no need upgrading

### 4.10.2. Get the local configuring parameters

Function: I\_GetLocalCfg ()  
Instruction: Get the local configuring parameters  
Parameters: none  
Return value: return the local configuring parameters of plugin (XML format)  
Note: the format is as follows:

```
<LocalConfigInfo>
    <ProtocolType></ProtocolType> //Protocol type: 0-TCP 2-UDP
    <PackageSize></PackageSize> //record package size: 0-256M 1-512M 2-1G
    <PlayWndType></PlayWndType> //play window type: 0-full 1-4:3 2-16:9
    <BuffNumberType></BuffNumberType> // Play Library Buffer Size
    <RecordPath></RecordPath> //the saving path of record
    <CapturePath> </CapturePath> //the saving path of captured pictures
during live view
    <PlaybackFilePath> </PlaybackFilePath> //the saving path of playback video
    <PlaybackPicPath> </PlaybackPicPath> //the saving path of captured
pictures during playback
    <DownloadPath> </DownloadPath> //the saving path of downloaded files
during playback
    <IVSMode></IVSMode> //whether to enable rules
    <CaptureFileFormat></CaptureFileFormat> //the capture pictures' format
</LocalConfigInfo>
```

### 4.10.3. Set the local configuration of plugins

Function: I\_SetLocalCfg ()  
Instruction: set the local configuration parameters  
Parameters: szLocalCofing the string of local configuration  
Return value: 0:success -1:failure

### 4.10.4. Get playing window status

Function: I\_GetWindowStatus (iWndIndex)  
Instruction: get the current window status  
Parameters: iWndIndex window index  
Return value: return window status if success; otherwise it returns null.

iIndex	window index
szIP	the IP address of device that is playing in window
iChannelID	the channel ID that is playing in window
iPlayStatus	the playing status of window(0-no play,1-live view, 2-playback, 3-pause, 4-single frame, 5-reverse playback, 6-pause when reverse playback)

## 4.11. Others

### 4.11.1. Folder selection and files' path

Function: I\_OpenFileDialog (iType)  
Instruction: Open folders or files' path  
Parameters: iType Type: 1-folder, 2-file  
Return value: It will return the selected folder or files' path

### 4.11.2. Get the last error code

Function: I\_GetLastError ()  
Instruction: get the error code  
Parameters: none  
Return value: It will return error code.