HTTP API Protocol User Guide

For IP Media Device

Version 1.4

2015-12

Document History

No	Release Notes	Date	Version	Author
1	Draft	2014-2-14	1.0	
2	Draft	2014-4-15	1.1	
3	Draft	2014-9-30	1.2	
4	Release	2014-11-6	1.2	Luyugang
5	Draft	2014-12-5	1.3	Chenshiguang
6	Draft	2015-12-22	1.4	Ouyangming

Contents

1 Overview

1.1 Preface

This document details the API of IP media devices. Programmers can access and configure IP media devices following the API.

1.2 Transaction

The HTTP API transaction starts from a request from a client application, usually a web browser. Theweb server on the IP media devices processes the request and sends the response back to the client application. The HTTP requestis taken in POST form as described in the following paragraphs. If the request is successful, the IP media video device will return a HTTP header contains 200 OK. The HTTP Body will contain actual result or error message if an error occurs.

1.3 Protocol Description

The client application should use POST form to send requests to the IP media devices. Other forms are not supported in this specification.

1.3.1 URL

The URL scheme is used to specify a request to the device locate device resources via a specific protocol in the network.This section defines the syntax and semantics for HTTP URLs.

<protocol>://<host>[:port]</cmd name> [/channelId][/action name]

protocol: URL scheme for the particular request. The HTTP protocol is allowedin this specification.

host: The host field refer to the hostname, IP address, or the FQDN(Fully Qualified Domain Name) of an IP device.

port: The port field refer to the port number of that host on which the identified resource islocated at the IP device listening for TCP connections. If the port is empty or not given, thedefault port is assumed. For HTTP, the default port 80.

cmd name: The specific command to an IP device.

channelId: The channel identification for an IP device. For the IP camera, this field can be omitted, the default channelId is "1".

action name: This field is optional. It acts as a sub operation for complex commands.

1.3.2 Connection Header Filed

Requests from the video management system or the client application are packed in HTTP messages. A request message composed of three parts: the connection header field, the authorization header field, and the entity body field.

HTTP/1.1 is implemented and utilized according to RFC 2616 inthe IP devices. For a video management system or client application that uses persistentconnection for multiple transactions, it is required to implement "Connection: Keep-Alive"HTTP header field as follows.

```
POST http://192.168.6.37/PtzAddPreset
HTTP/1.1
…
Content-Length: 135
…
Connection: Keep-Alive
…
```

1.3.3 Authorization Header Field

When a video management system or client application sends any request to the IP device,it must be authenticated by means of Basic Accessaccording to RFC 2617.

Authorization header field needs to be sentalong with each request, and if a user is authenticated, the request will follow the normalexecution flow. For the request with no authentication credentials, unauthorizedHTTP response (401) will be returned with WWW-Authenticate header field.

For example:

1. An HTTP request from the client application should include the “Authorization” information as follows, the “YWRtaW46MTIzNDU2” is the encoded result of “admin:123456” by base64:

POST http://192.168.6.37/PtzAddPreset

HTTP/1.1

…

Authorization: Basic YWRtaW46MQ==

…

2. The device responses the following to a requestwith no authentication credentials:

401 Unauthorized

WWW-Authenticate: Basic realm=”XXXXXX”

Then the client application encodes the username and password with base64, and sends the following request:

Authorization: Basic VXZVXZ.

1.3.4 Entity Body Field

Some requests will include entity body field. The Content-Type entity-header field indicates the media type of the entity body. TheContent-Type may be designated as “application/xml; charset=’UTF-8’”. For example:

POST http://192.168.6.37/PtzAddPreset

HTTP/1.1

...

Content-Type: application/xml; charset=”UTF-8”

...

<?xml version="1.0" encoding="utf-8" ?>

<presetInfo>

<name>preset1</name>

</presetInfo>

1.3.5 Response Message

The response message from the IP device is a standard HTTP response, information can be included in the entity body field in XML format. This information includes the result to a request message, or the detailed parameters that required by a request message.

A successful response that includes the result is as follows:

HTTP/1.1 200 OK

…

Content-Type: application/xml; charset=”UTF-8”

Content-Length: 66

Connection: close

…

<?xml version="1.0" encoding="UTF-8"?>

<config status="success"/>

A successful response that includes the detailed parameters is as follows:

HTTP/1.1 200 OK

…

Content-Type: application/xml; charset=”UTF-8”

Content-Length: 66

Connection: close

…

<?xml version="1.0" encoding="UTF-8"?>

<config version="1.0" xmlns="http://www.ipc.com/ver10">

…

<deviceInfo>

<supportTalk type="boolean">true</supportTalk>

…

</deviceInfo>

</config>

When a request cannot be executed correctly, an application fail response that includes an error result in the entity body will be sent from the IP device. Meantime, the HTTP answer is 400 to indicate the client application. For example:

HTTP/1.1 400 Bad Request

...

Content-Type: application/xml

Content-Length: 66

Connection: close

<?xml version="1.0" encoding="utf-8" ?>

<config status="failed" errorCode="1"/>

The detailed “errorCode” will be described in the following section.

1.3.6 Error Code

Error Code	Description
1	“Invalid Request”: The request URL is not supported by the device. There is something wrong with “cmd name”, “channelId”, or “action name”.
2	“Invalid XML Format”: The entity’s XML format is not recognized by the system.
3	“Invalid XML Content”:An incomplete message or a message containing some out-of-range parameters.

1.4 Protocol Conventions

1.4.1 XML Element Name

There will be several words in one element name, in this case, the first letter of the first word should be in lower case, the first letter of other words should be in upper case, and all other letters should be in lower case.

1.4.2 XML Element Type

Each element has an attribute “type”, which defines the data type of the element. The basic data types are listed as follows:

Type	Description
boolean	The same as “bool” in C++, available value is “true” or “false”.
int8	8 bit integer, the same as “char” in C/C++.

Type	Description
uint8	Unsigned 8 bit integer, the same as “unsigned char” in C/C++.
int16	16 bit integer, the same as “short” in C/C++.
uint16	Unsigned 16 bit integer, the same as “unsigned short” in C/C++.
int32	32 bit integer, the same as “long” in C/C++.
uint32	Unsigned 32 bit integer, the same as “unsigned long” in C/C++.
int64	64 bit integer, the same as “long long” in C/C++.
uint64	Unsigned 64 bit integer, the same as “unsigned long long” in C/C++.
string	A string of characters, like the “string” in C++.
list	List of basic or advanced types.

For the element with type “int8/uint8/int16/uint16/int32/uint32/int64/uint64”, two more attributes “min” and “max” can be optional, which mean the minimum and maximum value of this element. For example:

```
<bright type="uint8" min="0" max="100" default="50">50</bright>
```

For the element with type “string” attribute, two more attributes “minLen” and “maxLen” are optional, which mean the minimum and maximum length of the character string. When the type “string”attribute is used, the string itself should be packed in the CDATA segment. For example:

<ntpServer type="string" minLen="0" maxLen="127" default="time.windows.com"><![CDATA[time.windows.com]]></ntpServer>

For the element with type “list” attribute, the attribute “maxCount” should be used for the variable list, which means the maximum item counts for this list, and the attribute “count” should be used for the list with constant items. There should be an “itemType” sub element after the element with type “list” attribute. Some “item” sub element should be included after the “itemType” sub element to indicate the value for the list. For example:

<content type="list" count="6">

<itemType type="string" minLen="0" maxLen="32" default="00000000000000000000000000000000"/>

<item><![CDATA[11111111111111111111]]></item>

<item><![CDATA[22222222222222222222]]></item>

<item><![CDATA[33333333333333333333]]></item>

<item><![CDATA[44444444444444444444]]></item>

<item><![CDATA[55555555555555555555]]></item>

<item><![CDATA[66666666666666666666]]></item>

</content>

1.4.3 The “types”Element

When the basic data types cannot meet the demands, the “types” element should be used to define advanced data types. We don’t define any advanced data types in this document. Either, all advanced data types that will be used in a message should be defined in the message body. This means**“The messages themselves are documents”**.

In the “types” element, only the “enum” type can be defined. For example, an “enum” type is defined as follows:

<types>

<userType>

```
            <enum>administrator</enum>
            <enum>advance</enum>
            <enum>normal</enum>
        </userType>
    </types>
```

It is not allowed for the client application to define advanced data types with the “types” element in request messages. The client application should study advanced data types from the response messages. Advanced data types defined in the corresponding response message can be used directly in a request message by the client application. The Client application can also study advanced data types from other elements except for “types” in the message entity from the device.

1.4.4 Command catagory

We divide all commands into different categories that will be detailed in the following paragraphs.

System commands.

Image commands.

PTZ commands.

Alarm commands.

Network commands.

Security commands.

Maintain commands.

1.5 Device discovery

The IP media devices support UPnP protocol for device discovery.

The IP devices support Universal Plug and Play (UPnP) technology to discovery/locatethemselves. An UPnP compatible device will automatically announce its network

address,supported devices and services types when connected to a network, thereforebecoming “plug-and-play” by allowing clients recognize those information and begin usingthis device immediately.

The UPnP architecture supports zero-configuration networking, and the device candynamically join a network, obtain IP address, announce its name, convey its capabilitiesupon request, and gets the on-line status and capabilities of other devices. DHCP andDNS servers are optional and are only used if they are available on the network. Devicescan leave the network automatically without leaving any unwanted status informationbehind. UPnP was published as a 73-part International Standard, ISO/IEC 29341, inDecember, 2008 [6][7][8].

After a control point has discovered a device, the control point still needs more operationsto request more information about the device or to interact with it.

2 System commands

2.1 Device Information

2.1.1 GetDeviceInfo

GetDeviceInfo	
Description	To get the IP media device’s information.
Typical URL	POST or GET http://<host>[:port]/GetDeviceInfo
Channel ID	None
Action name	None

GetDeviceInfo	
Entity Data	None
Successful Response	The device information will be included in the entity of the successful response. For example:

GetDeviceInfo

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
    <deviceInfo>
        <model type="string"><![CDATA[TD-9421M]]></model>
        <brand type="string"><![CDATA[TVT]]></brand>
        <deviceDescription type="string"><![CDATA[IPCamera]]></deviceDescription>
        <audioInCount type="uint32">1</audioInCount>
        <audioOutCount type="uint32">1</audioOutCount>
        <integratedPtz type="boolean">true</integratedPtz>
        <supportRS485Ptz type="boolean">false</supportRS485Ptz>
        <supportSDCard type="boolean">true</supportSDCard>
        <alarmInCount type="uint32">1</alarmInCount>
        <alarmOutCount type="uint32">1</alarmOutCount>
        <softwareVersion type="string"><![CDATA[4.0.0 beta1]]></softwareVersion>
        <softwareBuildDate type="string"><![CDATA[2013-12-24]]></softwareBuildDate>
        <kernelVersion type="string"><![CDATA[20111010]]></kernelVersion>
        <hardwareVersion type="string"><![CDATA[1.3]]></hardwareVersion>
        <mac type="string"><![CDATA[00:18:ae:98:38:fd]]></mac>
    </deviceInfo>
</config>
```

[Tips]:

This command is designed for the client application to obtain the basic information from the specific media device.

- For the fixed-channel devices such as IPC or DVR, the items “audioInCount”, “audioOutCount”, “alarmInCount” and “alarmOutCount” will be included in the successful response.
- For the variable-channel devices such as NVR, these items are optional. The client application can use “GetChannelList”, “GetAlarmInList”, “GetAlarmOutList”, “GetStreamCpas” commands to obtain the information.

2.1.2 GetChannelList

GetChannelList	
Description	To get the IP media device's channel list.
Typical URL	POST or GET http://<host>[:port]/GetChannelList
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The channel list will be included in the entity of the successful response. For example:

<?xml version="1.0" encoding="UTF-8"?>

<config version="1.0" xmlns="http://www.ipc.com/ver10">

<channelIDList type="list" count="4">

<itemTypetype="string" maxLen="20"/>

<item>1</item>

<item>2</item>

<item>3</item>

<item>4</item>

</channelList>

</config>

[Tips]:

This command is designed for multi-channel device and not mandatory for IP cameras. If the "deviceDescription" item is equal to "IPCamera" in the response message for "GetDeviceInfo" command, this command should not be sent to the device.

2.1.3 GetAlarmInList

GetAlarmInList	
Description	To get the IP media device's alarmin list.

GetAlarmInList	
Typical URL	POST or GET http://<host>[:port]/GetAlarmInList
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The alarmin list will be included in the entity of the successful response. For example:

<?xml version="1.0" encoding="UTF-8"?>

<config version="1.0" xmlns="http://www.ipc.com/ver10">

<alarmInIDList type="list" count="8">

<itemTypetype="string" maxLen="20"/>

<item>1</item>

<item>2</item>

<item>3</item>

<item>4</item>

<item>5</item>

<item>6</item>

<item>7</item>

<item>8</item>

</alarmInList>

</config>

[Tips]:

This command is designed for multi-channel device and not mandatory for IP cameras. If the "deviceDescription" item is equal to "IPCamera" in the response message for "GetDeviceInfo" command, this command should not be sent to the device.

2.1.4 GetAlarmOutList

GetAlarmOutList	
Description	To get the IP media device's alarmout list.

GetAlarmOutList	
Typical URL	POST or GET http://<host>[:port]/GetAlarmOutList
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The alarmout list will be included in the entity of the successful response. For example:

<?xml version="1.0" encoding="UTF-8"?>

<config version="1.0" xmlns="http://www.ipc.com/ver10">

<alarmOutIDList type="list" count="4">

<itemTypetype="string" maxLen="20"/>

<item>1</item>

<item>2</item>

<item>3</item>

<item>4</item>

</alarmOutList>

</config>

[Tips]:

This command is designed for multi-channel device and not mandatory for IP cameras. If the "deviceDescription" item is equal to "IPCamera" in the response message for "GetDeviceInfo" command, this command should not be sent to the device.

2.2 Date and Time

2.2.1 GetDateAndTime

GetDateAndTime	
Description	To get the IP media device's system date and time.
Typical URL	POST or GEThttp://<host>[:port]/GetDateAndTime

GetDateAndTime	
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The device time and date will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <synchronizeType>
            <enum>manually</enum>
            <enum>NTP</enum>
        </synchronizeType>
    </types>
    <time>
        <timezoneInfo>
            <timeZone type="string"><![CDATA[GMT0BST,M3.5.0/1,M10.5.0]]></timeZone>
            <daylightSwitch type="boolean">false</daylightSwitch>
        </timezoneInfo>
        <synchronizeInfo>
            <type type="synchronizeType">manually</type>
            <ntpServer type="string" maxLen="127"><![CDATA[time.windows.com]]></ntpServer>
            <currentTime type="string"><![CDATA[2014-01-09 15:07:28]]></currentTime>
        </synchronizeInfo>
    </time>
</config>
```

[Tips]:

The element “timeZone” announces the time zone information. “GMT0BST,M3.5.0/1,M10.5.0”, this time zone, standard time named GMT and daylight saving time named BST, has daylight saving time. The standard local time is GMT. Daylight saving time, 1 hour ahead of GMT, starts the last Sunday in March at

GetDateAndTime
01:00 and ends the last Sunday in October at 02:00.

2.2.2 SetDateAndTime

SetDateAndTime	
Description	To set the IP media device's system date and time.
Typical URL	POST http://<host>[:port]/SetDateAndTime
Channel ID	None
Action name	None
Entity Data	The device time and date will be included in the entity of request message. The whole "time" element in the "GetDataAndTime" should be included in entity of this message. Any attributes for the "time" element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

3 Image commands

3.1 Stream Capabilities

3.1.1 GetStreamCaps

GetStreamCaps

GetStreamCaps	
Description	To get the IP media device's streams' capabilities for specific channel.
Typical URL	POST or GEThttp://<host>[:port]/GetStreamCaps[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The stream capabilities will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<resolution>
    <enum>1920x1080</enum>
        <enum>1280x720</enum>
        <enum>704x576</enum>
        <enum>352x288</enum>
</resolution>
<encodeType>
<enum>h264</enum>
<enum>mpeg4</enum>
<enum>mjpeg</enum>
            ...
</encodeType>
<encodeLevel>
<enum>baseLine</enum>
<enum>mainProfile</enum>
<enum>highProfile</enum>
</encodeLevel>
</types>
```

GetStreamCaps

```
<rtspPort type="uint16">554</rtspPort>
<streamList type="list" count="4">
<item id="1">
<streamName type="string"><![CDATA[profile1]]></streamName>
<resolutionCaps type="list" count="1">
<itemType type="resolution"/>
<item maxFrameRate="25">1920x1080</item>
</resolutionCaps>
<encodeTypeCaps type="list" count="1">
<itemType type="encodeType"/>
<item>h264</item>
</encodeTypeCaps>
<encodeLevelCaps type="list" count="3">
<itemType type="encodeLevel"/>
<item>baseLine</item>
<item>mainProfile</item>
<item>highProfile</item>
</encodeLevelCaps>
</item>
<item id="2">
<streamName type="string"><![CDATA[profile2]]></streamName>
<resolutionCaps type="list" count="3">
<itemType type="resolution"/>
<item maxFrameRate="10">1920x1080</item>
<item maxFrameRate="25">1280x720</item>
<item maxFrameRate="25">704x480</item>
</resolutionCaps>
<encodeTypeCaps type="list" count="1">
<itemType type="encodeType"/>
<item>h264</item>
</encodeTypeCaps>
```

GetStreamCaps
<encodeLevelCaps type="list" count="3"> <itemType type="encodeLevel"/> <item>baseLine</item> <item>mainProfile</item> <item>highProfile</item> </encodeLevelCaps> </item> … </streamList> </config>

[Tips]:

The “count=4” means the channel supports 4 streams at the same time. Each stream’s capability is announced in the “item” sub element. The “streamName” announces the name of each stream. The client application can obtain the specific stream by the following URL.

rtsp://<host><:port>/chID=[channelId]&streamType=[streamType]&linkType=tcp

//streamtype =main|sub

The “resolutionCaps” announces optional combinations for frame rate and resolution. The “encodeTypeCaps” announces optional compression types. The “encodeLevelCaps” optional compression levels.

For the reason that the capabilities for each stream are not the same, we omit the “itemType” element after the “streamList” element.

The “id” attribute for each item starts from “1”.

3.2 Image Configuration

3.2.1 GetImageConfig

GetImageConfig	
Description	To get the IP media device’s image configuration for specific channel.
Typical URL	POST or GEThttp://<host>[:port]/GetImageConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.

GetImageConfig	
Action name	None
Entity Data	None
Successful Response	The image configuration will be included in the entity of the Successful response. For example:

<?xml version="1.0" encoding="UTF-8"?>

<config version="1.0" xmlns="http://www.ipc.com/ver10">

<types>

<frequency>

<enum>60HZ</enum>

<enum>50HZ</enum>

</frequency>

<whitebalanceMode>

<enum>auto</enum>

<enum>manual</enum>

<enum>outdoor</enum>

<enum>indoor</enum>

</whitebalanceMode>

<IRCutMode>

<enum>auto</enum>

<enum>day</enum>

<enum>night</enum>

</IRCutMode>

</types>

<image>

<frequency type="frequency" default="50HZ">50HZ</frequency>

<bright type="uint8" min="0" max="100" default="50">50</bright>

<contrast type="uint8" min="0" max="100" default="55">55</contrast>

<hue type="uint8" min="0" max="100" default="50">50</hue>

<saturation type="uint8" min="0" max="100" default="50">50</saturation>

<mirrorSwitch type="boolean" default="false">false</mirrorSwitch>

GetImageConfig
<flipSwitch type="boolean" default="false">false</flipSwitch> <WDR> <switch type="boolean" default="false">false</switch> <value type="uint8" default="128">128</value> </WDR> <whiteBalance> <mode type="whitebalanceMode" default="auto">auto</mode> <red type="uint32" min="0" max="100" default="50">50</red> <blue type="uint32" min="0" max="100" default="50">50</blue> </whiteBalance> <denoise> <switch type="boolean" default="false">false</switch> <value type="uint8" default="24">24</value> </denoise> <irisSwitch type="boolean" default="false">false</irisSwitch> <sharpen> <switch type="boolean" default="true">true</switch> <value type="uint8" default="80">80</value> </sharpen> <IRCutMode type="IRCutMode" default="auto">auto</IRCutMode> </image> </config>

3.2.2 SetImageConfig

SetImageConfig	
Description	To set the IP media device's image configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetImageConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.

SetImageConfig	
Action name	None
Entity Data	The image configuration for specific channel should be included in the entity of request message. The whole “image” element in the “GetImagConfig” or some parameters that need to be changed can be included in entity of this message. Any attributes for the “image” element or sub elements should not be included. The following example changes the “saturation” parameter.
<?xml version="1.0"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <image> <saturation>65</saturation> </image> </config>	
Successful Response	The standard successful result response that described in 1.3.5.

3.2.3 GetSnapshot

GetSnapshot	
Description	To get a picture encoded by jpg for specific channel.
Typical URL	POST or GEThttp://<host>[:port]/GetSnapshot[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	A picture encoded by jpg.

3.3 Stream Configuration

3.3.1 GetAudioStreamConfig

GetAudioStreamConfig	
Description	To get the IP media device's audio stream configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetAudioStreamConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The audio stream configuration will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<audioEncode>
<enum>G711</enum>
<enum>G726</enum>
</audioEncode>
</types>
<audioEncode type="audioEncode">G711</audioEncode>
</config>
```

3.3.2 SetAudioStreamConfig

SetAudioStreamConfig	
Description	To set the IP media device's audio stream configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetAudioStreamConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.

SetAudioStreamConfig	
Action name	None
Entity Data	The audio stream configuration for specific channel should be included in the entity of request message. The whole “audioEncode” element in the “GetAudioStreamConfig” can be included in entity of this message. Any attributes for the “audioEncode” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

3.3.3 GetVideoStreamConfig

GetVideoStreamConfig	
Description	To get the IP media device’s video stream configuration for specific channel.
Typical URL	POST or GEThttp://<host>[:port]/GetVideoStreamConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The video stream configuration will be included in the entity of the successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<bitRateType>
<enum>VBR</enum>
<enum>CBR</enum>
</bitRateType>
<quality>
<enum>lowest</enum>
<enum>lower</enum>
<enum>medium</enum>
```

GetVideoStreamConfig

```
<enum>higher</enum>
<enum>highest</enum>
</quality>
</types>
<streams type="list" count="4">
<item id="1">
<name type="string" maxLen="32"><![CDATA[profile1]]></name>
<resolution>1920x1080</resolution>
<frameRate type="uint32">25</frameRate>
<bitRateType type="bitRateType">CBR</bitRateType>
<maxBitRate type="uint32" min="64" max="12288">4096</maxBitRate>
<encodeType>h264</encodeType>
<encodeLevel>baseLine</encodeLevel>
<quality type="quality">highest</quality>
<GOP type="uint32" min="30" max="200">100</GOP>
</item>
<item id="2">
<name type="string" maxLen="32"><![CDATA[profile2]]></name>
<resolution>1280x720</resolution>
<frameRate type="uint32">25</frameRate>
<bitRateType type="bitRateType">CBR</bitRateType>
<maxBitRate type="uint32" min="64" max="10240">2048</maxBitRate>
<encodeType>h264</encodeType>
<encodeLevel>baseLine</encodeLevel>
<quality type="quality">highest</quality>
<GOP type="uint32" min="30" max="200">100</GOP>
</item>
        ...
</streams>
</config>
```

GetVideoStreamConfig
[Tips]: The “count=4” means the channel supports 4 streams at the same time. Each stream’s current video configuration is announced in the “item” sub element. The value of each stream’s “resolution”, “framRate”, “encodeType”, and “encodeLevel” should be in the scope of the corresponding capability announced in the “GeteStreamCaps” successful respond message. The “maxBitRate” element means the bitrate in kbps. The “id” attribute for each item starts from “1”.

3.3.4 SetVideoStreamConfig

SetVideoStreamConfig	
Description	To set the IP media device’s video stream configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetVideoStreamConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The video stream configuration for specific channel should be included in the entity of request message. The whole “streams” element in the “GetVideoStreamConfig” can be included in entity of this message. Any attributes for the “streams” element or sub elements should not be included. The value of each stream’s “resolution”, “framRate”, “encodeType”, and “encodeLevel” should be in the scope of the corresponding capability announced in the “GeteStreamCaps” successful respond message.
Successful Response	The standard successful result response that described in 1.3.5.

3.3.5 RequestKeyFrame

RequestKeyFrame	
Description	It is used to request the device to encode a key frame for specific channel.
Typical URL	POST or GEThttp://<host>[:port]/ RequestKeyFrame [/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None

RequestKeyFrame	
Entity Data	None
Successful Response	The standard successful result response that described in 1.3.5.

3.4 OSD

3.4.1 GetImageOsdConfig

GetImageOsdConfig	
Description	To get the IP media device's image OSD configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetImageOsdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The image OSD configuration will be included in the entity of the Successful response. For example:

GetImageOsdConfig

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<dateFormat>
<enum>year-month-day</enum>
<enum>month-day-year</enum>
<enum>day-month-year</enum>
</dateFormat>
</types>
<imageOsd>
<time>
<switch type="boolean">true</switch>
<X type="uint32">0</X>
<Y type="uint32">0</Y>
<dateFormat type="dateFormat">year-month-day</dateFormat>
</time>
<channelName>
<switch type="boolean">false</switch>
<X type="uint32">0</X>
<Y type="uint32">0</Y>
<name type="string" maxLen="19"><![CDATA[name]]></name>
</channelName>
</imageOsd>
</config>
```

[Tips]:

The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution.

3.4.2 SetImageOsdConfig

SetImageOsdConfig	
Description	To set the IP media device's image OSD configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetImageOsdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The image OSD configuration for specific channel should be included in the entity of request message. The whole "imageOsd" element in the "GetImagOsdConfig" or some parameters that need to be changed can be included in entity of this message. Any attributes for the "imageOsd" element or sub elements should not be included. The following example changes the "channelName" element:
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <imageOsd> <channelName> <switch>true</switch> <X>100</X> <Y>100</Y> <name><![CDATA[camera01]]></name> </channelName> </imageOsd> </config>	
Successful Response	The standard successful result response that described in 1.3.5.

3.5 Privacy Mask

3.5.1 GetPrivacyMaskConfig

GetPrivacyMaskConfig

GetPrivacyMaskConfig	
Description	To get the IP media device's privacy mask configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetPrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The privacy mask configuration will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<color>
<enum>black</enum>
<enum>white</enum>
<enum>gray</enum>
</color>
</types>
<privacyMask type="list" count="4">
<itemType>
<switch type="boolean"/>
<rectangle>
<X type="uint32"/>
<Y type="uint32"/>
<width type="uint32"/>
<height type="uint32"/>
</rectangle>
<color type="color"/>
</itemType>
<item>
```

GetPrivacyMaskConfig
<switch>false</switch> <rectangle> <X>0</X> <Y>0</Y> <width>0</width> <height>0</height> </rectangle> <color>black</color> </item> ... </privacyMask> </config> [Tips]: The “X” and “Y” element announce the horizontal and vertical position based in the 640*480 resolution.

3.5.2 SetPrivacyMaskConfig

SetPrivacyMaskConfig	
Description	To set the IP media device’s privacy mask configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetPrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The privacy mask configuration for specific channel should be included in the entity of request message. The whole “privacyMask” element in the “GetPrivacyMaskConfig” should be included in entity of this message. Any attributes for the “privacyMask” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

4 PTZ commands

4.1 Protocol

4.1.1 PtzGetCaps

PtzGetCaps	
Description	To get the IP media device's PTZ capabilities mask information for specific channel.
Typical URL	POST or GET http://<host>[:port]/PtzGetCaps[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The PTZ capabilities will be included in the entity of the Successful response. For example:

PtzGetCaps
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <caps> <controlMinSpeed type="uint32">1</controlMinSpeed> <controlMaxSpeed type="uint32">8</controlMaxSpeed> <presetMaxCount type="uint32">255</presetMaxCount> <cruiseMaxCount type="uint32">8</cruiseMaxCount> <cruisePresetMinSpeed type="uint32">1</cruisePresetMinSpeed> <cruisePresetMaxSpeed type="uint32">8</cruisePresetMaxSpeed> <cruisePresetMaxHoldTime type="uint32">240</cruisePresetMaxHoldTime> <cruisePresetMaxCount type="uint32">16</cruisePresetMaxCount> </caps> </config>
[Tips]: The sub elements in the “caps” element announce the scope of each parameter. For example, the “ptzControlMinSpeed” announce the minimum speed for the PTZ control command, the “ptzControlMaxSpeed” announce the maximum speed for the PTZ control command.

4.1.2 PtzGetProtocolConfig

PtzGetProtocolConfig	
Description	To get the IP media device’s PTZ protocol configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/PtzGetProtocolConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The PTZ protocol configuration will be included in the entity of the Successful response. For example:

PtzGetProtocolConfig

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<baudRate>
<enum>110</enum>
<enum>300</enum>
<enum>600</enum>
<enum>1200</enum>
<enum>2400</enum>
<enum>4800</enum>
<enum>9600</enum>
<enum>19200</enum>
<enum>38400</enum>
<enum>57600</enum>
<enum>115200</enum>
<enum>230400</enum>
<enum>460800</enum>
<enum>921600</enum>
</baudRate>
<ptzProtocol>
<enum>PELCOP</enum>
<enum>PELCOD</enum>
<enum>LILIN</enum>
<enum>MINKING</enum>
<enum>NEON</enum>
<enum>STAR</enum>
<enum>VIDO</enum>
<enum>DSCP</enum>
<enum>VISCA</enum>
</ptzProtocol>
```

PtzGetProtocolConfig
</types> <ptzInfo> <baudRate type="baudRate">4800</baudRate> <protocol type="ptzProtocol">PELCOP</protocol> <address type="uint8">2</address> </ptzInfo> </config>

4.1.3 PtzSetProtocolConfig

PtzSetProtocolConfig	
Description	To set the IP media device's PTZ protocol configuration for specific channel.
Typical URL	POST http://<host>[:port]/PtzSetProtocolConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ protocol configuration for specific channel should be included in the entity of request message. The whole "ptzInfo" element in the "PtzGetProtocolConfig" should be included in entity of this message. Any attributes for the "ptzInfo" element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

4.2 PTZ Control

4.2.1 PtzControl

PtzControl	
Description	To start control PTZ for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzControl[/channelId]</action_name>
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.

PtzControl	
Action name	Up: to move up Down: to move down Left: to move left Right: to move right LeftUp: to move left and up LeftDown: to move left and down RightUp: to move right and up RigthDown: to move right and down Near: to focus near Far: to focus far ZoomIn: to zoom in ZoomOut: to zoom out IrisOpen: to open the iris IrisClose: to close the iris Stop: to stop current action
Entity Data	The PTZ's action information that needs to be executed will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <actionInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <speed>4</speed> </actionInfo> [Tips]: The value of "speed" should be in the scope of the corresponding capability announced in the "PtzGetCaps" successful respond message.	
Successful Response	The standard successful result response that described in 1.3.5.

4.2.2 PtzGotoPreset

PtzGotoPreset

PtzGotoPreset	
Description	To run the PTZ to one preset for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzGotoPreset[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The destination PTZ preset's ID will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <presetInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>2</id> </presetInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.2.3 PtzRunCruise

PtzRunCruise	
Description	To run one PTZ's cruise for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzRunCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ cruise's ID that needs to be run will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <cruiseInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> </cruiseInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.2.4 PtzStopCruise

PtzStopCruise	
Description	To stop the PTZ cruise for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzStopCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The standard successful result response that described in 1.3.5.

4.3 Preset

4.3.1 PtzGetPresets

PtzGetPresets	
Description	To get the IP media device's PTZ presets list for specific channel.
Typical URL	POST or GET http://<host>[:port]/PtzGetPresets[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The PTZ presets list will be included in the entity of the Successful response. For example:

PtzGetPresets
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <presetInfo type="list" maxCount="255"> <itemTypetype="string" maxLen="11"/> <item id="1"><![CDATA[DDD]]></item> </presetInfo> </config> [Tips]: The “id” attribute for each item starts from “1”.

4.3.2 PtzAddPreset

PtzAddPreset	
Description	To add one preset for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzAddPreset[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ preset name for current position will be included in the entity of the request message. The “name” should accord with the “name” type in the “itemType” that announced in “PtzGetPresets” message. For example:
<?xml version="1.0" encoding="utf-8" ?> <presetInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <name><![CDATA[dd]]></name> </presetInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.3.3 PtzModifyPresetName

PtzModifyPresetName	
Description	To modify one preset's name for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzModifyPresetName[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ preset's ID and new name will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <presetInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> <name><![CDATA[aa1]]></name> </presetInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.3.4 PtzDeletePreset

PtzDeletePreset	
Description	To delete one preset for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzDeletePreset[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ preset's ID that needs to be deleted will be included in the entity of the request message. For example:

PtzDeletePreset	
<?xml version="1.0" encoding="utf-8" ?> <presetInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> </presetInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.3.5 PtzModifyPresetPosition

PtzModifyPresePosition	
Description	To modify one preset's position to current position for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzModifyPresetPosition[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ preset's ID that needs to be modified to current position will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <presetInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>3</id> </presetInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

4.4 Cruise

4.4.1 PtzGetCruises

PtzGetCruises	
Description	To get the IP media device's PTZ cruises list for specific channel.

PtzGetCruises	
Typical URL	POST or GET http://<host>[:port]/PtzGetCruises[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The PTZ cruises list will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<cruiseInfo type="list" maxCount="8">
<itemType type="string" maxLen="31"/>
<item id="1"><![CDATA[SSS]]></item>
</cruiseInfo>
</config>
```

[Tips]:

The “id” attribute for each item starts from “1”.

4.4.2 PtzGetCruise

PtzGetCruise	
Description	To get one cruise configuration of the IP media device’s specific channel.
Typical URL	POST http://<host>[:port]/GetPtzCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ cruise’s ID that needs to be queried will be included in the entity of the request message. For example:

PtzGetCruise	
<?xml version="1.0" encoding="utf-8" ?> <cruiseInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> </cruiseInfo>	
Successful Response	The PTZ cruise's information will be included in the entity of the Successful response. For example:
<?xml version="1.0" encoding="UTF-8"?> <cruiseInfo> <id type="uint32">1</id> <name type="string" maxLen="31"><![CDATA[SSS]]></name> <presetInfo type="list" maxCount="16"> <itemType > <name type="string" maxLen="11"/> <speed type="uint32" min="1" max="8"/> <holdTime type="uint32" min="5" max="240"/> </itemType> <item id="1"> <name><![CDATA[DDD]]></name> <speed>5</speed> <holdTime>5</holdTime> </item> </presetInfo> </cruiseInfo> [Tips]: The "id" attribute for each item starts from "1".	

4.4.3 PtzAddCruise

PtzAddCruise

PtzAddCruise	
Description	To add one cruise for a specific channel of the IP media device.
Typical URL	POST http://<host>[:port]/PtzAddCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ cruise configuration for specific channel should be included in the entity of request message. The whole “cruiseInfo” element in the “GetPtzCruise” should be included in entity of this message. Any attributes for the “cruiseInfo” element or sub elements should not be included. For example:

```
<?xml version="1.0"?>
<cruiseInfo version="1.0" xmlns="http://www.ipc.com/ver10">
<name type="string"><![CDATA[c2]]></name>
<presetInfo>
<item id="2">
<speed>5</speed>
<holdTime>5</holdTime>
</item>
        ...
</presetInfo>
</cruiseInfo>
```

[Tips]:

The “id” attribute for each item starts from “1”.

Successful Response	The standard successful result response that described in 1.3.5.

4.4.4 PtzModifyCruise

PtzModifyCruise	
Description	To modify one cruise information of the IP media device’s specific channel.

PtzModifyCruise	
Typical URL	POST http://<host>[:port]/PtzModifyCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ cruise configuration for specific channel should be included in the entity of request message. The whole “cruiseInfo” element in the “GetPtzCruise” should be included in entity of this message. Any attributes for the “cruiseInfo” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

4.4.5 PtzDeleteCruise

PtzDeleteCruise	
Description	To delete one cruise of the IP media device’s specific channel.
Typical URL	POST http://<host>[:port]/PtzDeleteCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The PTZ cruise’s ID that needs to be deleted will be included in the entity of the request message. For example:
<?xml version="1.0" encoding="utf-8" ?> <cruiseInfo version="1.0" xmlns="http://www.ipc.com/ver10"> <id>2</id> </cruiseInfo>	
Successful Response	The standard successful result response that described in 1.3.5.

5 Alarm commands

5.1 Motion Detection

5.1.1 GetMotionConfig

GetMotionConfig	
Description	To get the IP media device's motion configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetMotionConfig [/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The motion configuration information will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<motion>
<switch type="boolean">false</switch>
<sensitivity type="int32" min="0" max="8">4</sensitivity>
<alarmHoldTime type="uint32">20</alarmHoldTime>
<area type="list" count="18">
<itemType type="string" minLen="22" maxLen="22"/>
<item><![CDATA[1111111111111111111111]]></item>
<item><![CDATA[1111111111111111111111]]></item>
<item><![CDATA[1111111111111111111111]]></item>
<item><![CDATA[1111111111111111111111]]></item>
<item><![CDATA[1111111111111111111111]]></item>
```

GetMotionConfig
<item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> <item><![CDATA[1111111111111111111111]]></item> </area> <triggerAlarmOut type="list" count="1"> <itemTypetype="boolean"/> <item id="1">false</item> </triggerAlarmOut> </motion> </config> [Tips]: There are 18 sub items in the “area”element, each item is a string with fixed length 22. This means a 22x18 motion detection areas, if corresponding character is “1”, the switch for this detection area is on. The “id” attribute for each item starts from “1”.

5.1.2 SetMotionConfig

SetMotionConfig	
Description	To set the IP media device’s motion configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetMotionConfig [/channelId]

SetMotionConfig	
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The motion detection configuration for specific channel should be included in the entity of request message. The whole “motion” element in the “GetMotionConfig” should be included in entity of this message. Any attributes for the “motion” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

5.2 Alarm

5.2.1 GetAlarmInConfig

GetAlarmInConfig	
Description	To get the IP media device’s alarm input configuration for specific alarm input channel.
Typical URL	POST or GET http://<host>[:port]/GetAlarmInConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default alarm input channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The alarm inputs configuration will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
<alarmInVoltage>
<enum>NO</enum>
<enum>NC</enum>
</alarmInVoltage>
```

GetAlarmInConfig
</types> <sensor> <id type="uint32">1</id> <sensorName type="string" maxLen="11"><![CDATA[Sensor1]]></sensorName> <switch type="boolean">true</switch> <voltage type="alarmInVoltage">NO</voltage> <alarmHoldTime type="uint32">10</alarmHoldTime> <triggerAlarmOut type="list" count="1"> <itemTypetype="boolean"/> <item id="1">true</item> </triggerAlarmOut> </sensor> </config> [Tips]: The “id” attribute for each item starts from “1”.

5.2.2 SetAlarmInConfig

SetAlarmInConfig	
Description	To set the IP media device’s alarm inputs configuration for specific alarm input channel.
Typical URL	POST http://<host>[:port]/SetAlarmInConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default alarm input channel ID is 1.
Action name	None
Entity Data	The alarm input configuration for specific channel should be included in the entity of request message. The whole “sensor” element in the “GetAlarmInConfig” should be included in entity of this message. Any attributes for the “sensor” element or sub elements should not be included.

SetAlarmInConfig	
Successful Response	The standard successful result response that described in 1.3.5.

5.2.3 ManualAlarmOut

ManualAlarmOut	
Description	To manually set the IP media device's alarm output status for specific alarm output channel.
Typical URL	POST http://<host>[:port]/ManualAlarmOut[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default alarm output channel ID is 1.
Action name	None
Entity Data	The new status for the specific alarm output will be included in the entity of request message. For example:
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <action> <status>true</status> </action> </config> [Tips]: The "status" element is Boolean type.	
Successful Response	The standard successful result response that described in 1.3.5.

5.2.4 GetAlarmOutConfig

GetAlarmOutConfig	
Description	To get the IP media device's alarm output configuration for specific alarm output channel.

GetAlarmOutConfig	
Typical URL	POST or GET http://<host>[:port]/GetAlarmOutConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default alarm output channel ID is 1.
Action name	None
Entity Data	None
Successful Response	The specific alarm output configuration will be included in the entity of the Successful response. For example:
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <alarmOut> <id type="uint32">1</id> <alarmOutName type="string" maxLen="11"><![CDATA[alarmOut1]]></alarmOutName> <alarmHoldTime type="uint32">20</alarmHoldTime> </alarmOut> </config>	

5.2.5 SetAlarmOutConfig

SetAlarmOutConfig	
Description	To set the IP media device's alarm output configuration for specific alarm output channel.
Typical URL	POST http://<host>[:port]/SetAlarmOutConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default alarm output channel ID is 1.
Action name	None

SetAlarmOutConfig	
Entity Data	The alarm output configuration for specific channel should be included in the entity of request message. The whole “alarmOut” element in the “GetAlarmOutConfig” should be included in entity of this message. Any attributes for the “alarmOut” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

5.3 AlarmStatus

5.3.1 GetAlarmStatus

GetAlarmStatus	
Description	To get the IP media device’s alarm trigger status.
Typical URL	POST or GET http://<host>[:port]/GetAlarmStatus
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The alarm trigger status information will be included in the entity of the Successful response. For example:

GetAlarmStatus
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <alarmStatusInfo> <motionAlarm type="boolean" id="1">false</motionAlarm> <motionAlarm type="boolean" id="2">true</motionAlarm> <motionAlarm type="boolean" id="3">false</motionAlarm> <motionAlarm type="boolean" id="4">false</motionAlarm> <sensorAlarmIn type="list" count="4"> <itemType type="boolean"/> <item id="1">false</item> <item id="2">false</item> <item id="3">false</item> <item id="4">false</item> </sensorAlarmIn> </alarmStatusInfo> </config> [Tips]: The “id” attribute for each item starts from “1”.

批注 [W1]: 为了保持和原来一致，采用加 id 的方式表示每个通道的 Motion 状态

5.3.2 GetAlarmServerConfig

GetAlarmServerConfig	
Description	To get the alarm server configuration
Typical URL	POST or GET http://<host>[:port]/GetAlarmServerConfig
Channel ID	None
Action name	None
Entity Data	None

GetAlarmServerConfig	
Successful Response	The alarm server configuration will be included in the entity of the Successful response. For example:
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <alarmServer> <serverAddr type="string"></serverAddress> <serverPort type=" uint16">80</ serverPort > </ alarmServer> </config>	

5.3.3 SetAlarmServerConfig

SetAlarmServerConfig	
Description	To set the alarm server configuration.
Typical URL	POST http://<host>[:port]/SetAlarmServerConfig
Channel ID	None
Action name	None
Entity Data	The alarm server configuration should be included in the entity of request message. The whole “alarmServer” element in the “GetAlarmServerConfig” should be included in entity of this message. Any attributes for the “alarmServer” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

5.3.4 SendAlarmStatus

SendAlarmStatus	
Description	To send the alarm status to the alarm server when an alarm happens. This command will be used by the device. The alarm server should provide HTTP service to receive this command.

SendAlarmStatus	
Typical URL	POST http://<alarm server>[:port]/SendAlarmStatus
Channel ID	None
Action name	None
Entity Data	The alarm status should be included in the entity of request message. The whole “alarmStatusInfo” element in the response for “GetAlarmStatus” should be included in entity of this message.
Successful Response	None

6 Playback

6.1 Record Search

6.1.1 SearchRecordDate

SearchByDate	
Description	To search the date list with record data for specific channel.
Typical URL	POST or GET http://<host>[:port]/SearchRecordDate[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	None

SearchByDate	
Successful Response	The date list with record data will be included in the entity of the successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<dateList type="list" count="6">
<itemType type="string"/>
<item>2014-01-09</item>
<item>2014-02-09</item>
<item>2014-03-08</item>
<item>2014-04-02</item>
<item>2014-04-03</item>
<item>2014-04-04</item>
</datelist>
</config>
```

6.1.2 SearchByTime

SearchByTime	
Description	To search record data segments for the specific channel by time.
Typical URL	POST or GET http://<host>[:port]/SearchByTime[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	The start time and end time should be included in the entity of the request message as search condition. For example:

SearchByTime

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<search>
<starttime type="string">![CDATA[2014-01-09 15:07:28]]</starttime>
<endtime type="string">![CDATA[2014-01-09 18:07:28]]</endtime>
</search>
</config>
```

Successful Response	The searched record data segments will be included in the entity of the successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<timesectionList type="list" count="5">
<itemType type="string"/>
<item type="string" seconds="200">![CDATA[2014-01-09 15:07:28]]</item>
<item type="string" seconds="8000">![CDATA[2014-01-09 16:07:28]]</item>
<item type="string" seconds="200">![CDATA[2014-01-09 18:07:28]]</item>
<item type="string" seconds="200">![CDATA[2014-01-11 15:07:28]]</item>
<item type="string" seconds="200">![CDATA[2014-01-12 15:07:28]]</item>
</ datelist>
</config>
```

[Tips]:

The client application can playback one specific record data segment through RTSP protocol. For example:

rtsp://<host><:rtspPort>/channelID/ date=2014-01-09&time=15:07:28&timelen=200

When this URL is invoked by the client application, the first record data segment searched by the device will be playback through RTSP.

7 Network commands

7.1 TCP/Ipv4

7.1.1 GetNetBasicConfig

GetNetBasicConfig	
Description	To get the IP media device's basic network configuration.
Typical URL	POST or GET http://<host>[:port]/GetNetBasicConfig
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The basic network configuration will be included in the entity of the Successful response. For example:

GetNetBasicConfig
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <types> <ipSettingMode> <enum>staticIp</enum> <enum>dhcp</enum> </ipSettingMode> </types> <tcpIp> <ipSettingMode type="ipSettingMode">staticIp</ipSettingMode> <staticIp type="string" minLen="7" maxLen="15"><![CDATA[192.168.6.36]]></staticIp> <staticIpRoute type="string" minLen="7" maxLen="15"><![CDATA[192.168.6.1]]></staticIpRoute> <staticIpMask type="string" minLen="7" maxLen="15"><![CDATA[255.255.255.0]]></staticIpMask> <dnsFromDhcpSwitch type="boolean">false</dnsFromDhcpSwitch> <dnsServer1 type="string" minLen="7" maxLen="15"><![CDATA[192.168.226.1]]></dnsServer1> <dnsServer2 type="string" minLen="7" maxLen="15"><![CDATA[8.8.8.8]]></dnsServer2> </tcpIp> </config>

7.1.2 SetNetBasicConfig

SetNetBasicConfig	
Description	To set the IP media device's basic network configuration.
Typical URL	POST http://<host>[:port]/SetNetBasicConfig
Channel ID	None
Action name	None
Entity Data	The basic network configuration should be included in the entity of request message. The whole "tcpIp" element in the "GetNetBasicConfig" should be included in entity of this message. Any attributes for the "tcpIp" element or sub elements should not be included.

SetNetBasicConfig	
Successful Response	The standard successful result response that described in 1.3.5.

7.2 PPPoE

7.2.1 GetNetPppoeConfig

GetNetPppoeConfig	
Description	To get the IP media device's network PPPOE configuration.
Typical URL	POST or GET http://<host>[:port]/GetNetPppoeConfig
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The network PPPOE configuration will be included in the entity of the Successful response. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<pppoe>
<switch type="boolean">false</switch>
<userName type="string"maxLen="63"><![CDATA[aaa]]></userName>
<password type="string"maxLen="63"><![CDATA[bbb]]></password>
</pppoe>
</config>
```

[Tips]:

The value of the "password" element will be none, for the reason that the "password" element is write-only.

7.2.2 SetNetPppoeConfig

SetNetPppoeConfig	
Description	To set the IP media device's network PPPOE configuration.
Typical URL	POST http://<host>[:port]/SetNetPppoeConfig
Channel ID	None
Action name	None
Entity Data	The network PPPOE configuration should be included in the entity of request message. The whole "pppoe" element in the "GetNetPppoeConfig" should be included in entity of this message. Any attributes for the "pppoe" element or sub elements should not be included. If the user doesn't need to change password, please omit the "password" element.
Successful Response	The standard successful result response that described in 1.3.5.

7.3 Port

7.3.1 GetPortConfig

GetPortConfig	
Description	To get the IP media device's network service ports configuration.
Typical URL	POST or GET http://<host>[:port]/GetPortConfig
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The network service ports configuration will be included in the entity of the Successful response. For example:

GetPortConfig
<?xml version="1.0" encoding="UTF-8"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <port> <httpPort type="uint16">80</httpPort> <netPort type="uint16">9008</netPort> <rtspPort type="uint16">554</rtspPort> </port> </config> [Tips]: The “httpPort” element announces the port for HTTP service. The “netPort” element announces the port for protocol service. The “rtspPort” element announces the port for RTSP service.

7.3.2 SetPortConfig

SetPortConfig	
Description	To set the IP media device’s network service ports configuration.
Typical URL	POST http://<host>[:port]/SetPortConfig
Channel ID	None
Action name	None
Entity Data	The network service ports configuration should be included in the entity of request message. The whole “port” element in the “GetPortConfig” should be included in entity of this message. Any attributes for the “port” element or sub elements should not be included.
Successful Response	The standard successful result response that described in 1.3.5.

7.4 DDNS

7.4.1 GetDdnsConfig

GetDdnsConfig	
Description	To get the IP media device's network DDNS configuration.
Typical URL	POST or GET http://<host>[:port]/GetDdnsConfig
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The network DDNS configuration will be included in the entity of the Successful response. For example:

GetDdnsConfig

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="1.0" xmlns="http://www.ipc.com/ver10">
<types>
    <ddnsServerType>
        <enum requireParameters="userName,password">www.88ip.net</enum>
        <enum requireParameters="userName,password">www.dns2p.net</enum>
        <enum requireParameters="userName,password">www.meibu.com</enum>
        <enum requireParameters="userName,password,domainName">www.dyndns.com</enum>
        <enum requireParameters="userName,password,domainName">www.no-ip.com</enum>
        <enum
requireParameters="userName,password,domainName,serverName">mintdns</enum>
        <enum requireParameters="userName,password,domainName">www.3322.org</enum>
    </ddnsServerType>
</types>
<ddns>
<switch type="boolean">false</switch>
<servertype type="ddnsServerType">www.88ip.com</servertype>
<userName type="string" maxLen="63"><![CDATA[aaa]]></userName>
<password type="string" maxLen="63"><![CDATA[]]></password>
<domainName type="string"maxLen="63"><![CDATA[ipc.88ip.com]]></domainName>
<serverName type="string" maxLen="63"><![CDATA[111]]></serverName>
</ddns>
</config>
```

[Tips]:

The value of the “password” element will be none, for the reason that the “password” element is write-only.

7.4.2 SetDdnsConfig

SetDdnsConfig

SetDdnsConfig	
Description	To set the IP media device's network DDNS configuration.
Typical URL	POST http://<host>[:port]/SetDdnsConfig
Channel ID	None
Action name	None
Entity Data	The network DDNS configuration should be included in the entity of request message. The whole "ddns" element in the "GetDdnsConfig" should be included in entity of this message. Any attributes for the "ddns" element or sub elements should not be included.If the user doesn't need to change password, please omit the "password" element.
Successful Response	The standard successful result response that described in 1.3.5.

8 Security commands

8.1 User Management

8.1.1 ModifyPassword

ModifyPassword	
Description	To modify the current login user's password for the IP media device.
Typical URL	POST http://<host>[:port]/ModifyPassword
Channel ID	None
Action name	None

ModifyPassword	
Entity Data	The new password will be included in the entity of request message. Any attributes for the “userPassword” element or sub elements should not be included. For example:
<?xml version="1.0"?> <config version="1.0" xmlns="http://www.ipc.com/ver10"> <userPassword> <oldPassword><![CDATA[YWFh]]></oldPassword> <password><![CDATA[YmJi]]></password> </userPassword> </config> [Tips]: The “oldPassword” and “password” elements are all “string” type with maxLen “19”. They should be encoded by base64, the “YWFh” and “YmJi” are the encoded result for “aaa” and “bbb”.	
Successful Response	The standard successful result response that described in 1.3.5.

9 Maintain commands

9.1 Reboot

9.1.1 Reboot

Reboot	
Description	To reboot the IP media device.

Reboot	
Typical URL	POST or GET http://<host>[:port]/Reboot
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The standard successful result response that described in 1.3.5.

10 Talkback commands

10.1 Talkback

10.1.1 Talkback

Reboot	
Description	To request device talkback stream.
Typical URL	GET http://<host>[:port]/profile_talk
Channel ID	None
Action name	None
Entity Data	None

Reboot	
Tips	When this URL is invoked by the client application, the talkback data stream pass through RTSP.