



HTTP API Protocol User Guide
For Viewtron IP Cameras

Version 2.0.0

2024-10

For more information on Viewtron IP cameras and all security products,
please visit www.Viewtron.com.

Email questions to
mike@viewtron.com

Contents

HTTP API PROTOCOL USER GUIDE	
FOR IP MEDIA DEVICE	
VERSION 2.0.0	
2024-10	
DOCUMENT HISTORY	ERROR! BOOKMARK NOT DEFINED.
1 OVERVIEW	1
1.1 PREFACE	1
1.2 TRANSACTION	1
1.3 PROTOCOL DESCRIPTION	1
1.3.1 URL	1
1.3.2 Connection Header Field	2
1.3.3 Authorization Header Field	3
1.3.4 Entity Body Field	3
1.3.5 Response Message	4
1.3.6 Error Code	6
1.4 PROTOCOL CONVENTIONS	9
1.4.1 XML Element Name	9
1.4.2 XML Element Type	9
1.4.3 The "types" Element	11
1.4.4 Command category	12
1.4.5 Annotation	12
1.5 DEVICE DISCOVERY	13
2 SYSTEM COMMANDS	14
2.1 DEVICE INFORMATION	14
2.1.1 GetSupportedAPIs	14
2.1.2 GetDeviceInfo	18
2.1.3 GetDiskInfo	22
2.1.4 GetChannelInfo	25
2.1.5 GetAlarmInInfo	28
2.1.6 GetAlarmOutInfo	29
2.2 DATE AND TIME	31
2.2.1 GetDateAndTime	31
2.2.2 SetDateAndTime	35
2.3 UPGRADE	38
2.3.1 UpdateState	38
2.3.2 UpdateSliceFirmware	39
3 IMAGE COMMANDS	42
3.1 STREAM CAPABILITIES	42
3.1.1 GetStreamCaps	42

3.2 IMAGE CONFIGURATION	49
3.2.1 GetImageConfig	49
3.2.2 SetImageConfig	67
3.2.3 GetImageConfigFileInfo	72
3.2.4 SetImageConfigFileInfo	75
3.2.5 GetSnapshot	76
3.2.6 GetSnapshotByTime	77
3.3 STREAM CONFIGURATION	79
3.3.1 GetAudioStreamConfig	79
3.3.2 SetAudioStreamConfig	81
3.3.3 GetVideoStreamConfig	83
3.3.4 SetVideoStreamConfig	87
3.3.5 RequestKeyFrame	89
3.4 OSD	90
3.4.1 GetImageOsdConfig	90
3.4.2 SetImageOsdConfig	98
3.5 PRIVACY MASK	104
3.5.1 GetPrivacyMaskConfig	104
3.5.2 AddPrivacyMaskConfig	106
3.5.3 ModifyPrivacyMaskConfig	108
3.5.4 DeletePrivacyMaskConfig	109
3.5.5 GotoPrivacyMaskLocation	110
3.6 ROI	ERROR! BOOKMARK NOT DEFINED.
3.6.1 GetROIConfig	Error! Bookmark not defined.
3.6.2 SetROIConfig	Error! Bookmark not defined.
4 PTZ COMMANDS	111
4.1 PROTOCOL	111
4.1.1 PtzGetCaps	111
4.1.2 GetPtzConfig	114
4.1.3 SetPtzConfig	116
4.2 PTZ CONTROL	118
4.2.1 PtzControl	118
4.2.2 PtzGotoPreset	120
4.2.3 PtzRunCruise	121
4.2.4 PtzStopCruise	122
4.3 PRESET	122
4.3.1 PtzGetPresets	122
4.3.2 PtzAddPreset	123
4.3.3 PtzModifyPresetName	124
4.3.4 PtzDeletePreset	125
4.3.5 PtzModifyPresetPosition	126
4.4 CRUISE	127
4.4.1 PtzGetCruises	127
4.4.2 PtzGetCruise	128
4.4.3 PtzAddCruise	130
4.4.4 PtzModifyCruise	132
4.4.5 PtzDeleteCruise	134
5 ALARM COMMANDS	135
5.1 MOTION DETECTION	135
5.1.1 GetMotionConfig	135
5.1.2 SetMotionConfig	140
5.2 ALARM	146

5.2.1	<i>GetAlarmInConfig</i>	146
5.2.2	<i>SetAlarmInConfig</i>	148
5.2.3	<i>ManualAlarmOut</i>	149
5.2.4	<i>GetAlarmOutConfig</i>	150
5.2.5	<i>SetAlarmOutConfig</i>	155
5.2.6	<i>AlarmOutputControl</i>	158
5.3	ALARMSTATUS	159
5.3.1	<i>GetAlarmStatusInfo</i>	159
5.3.2	<i>GetAlarmServerConfig</i>	167
5.3.3	<i>SetAlarmServerConfig</i>	169
5.4	ALARMTRIGGER	171
5.4.1	<i>GetAlarmTriggerConfig</i>	171
5.4.2	<i>SetAlarmTriggerConfig</i>	175
5.5	SOUND-LIGHT ALARM	179
5.5.1	<i>GetAudioAlarmOutConfig</i>	179
5.5.2	<i>SetAudioAlarmOutConfig</i>	181
5.5.3	<i>AddCustomizeAudioAlarm</i>	182
5.5.4	<i>DeleteCustomizeAudioAlarm</i>	184
5.5.5	<i>AuditionAudioAlarm</i>	185
5.5.6	<i>GetLightAlarmOutConfig</i>	186
5.5.7	<i>SetLightAlarmOutConfig</i>	187
5.6	DISARMING	189
5.6.1	<i>RequestDisarming</i>	189
5.6.2	<i>CancelDisarming</i>	189
5.6.3	<i>GetDisarmingStatus</i>	190
6	PLAYBACK	191
6.1	RECORD SEARCH	191
6.1.1	<i>GetRecordType</i>	191
6.1.2	<i>SearchRecordDate</i>	194
6.1.3	<i>SearchByTime</i>	195
6.2	RECORDSTATUS	198
6.2.1	<i>GetRecordStatusInfo</i>	198
7	NETWORK COMMANDS	201
7.1	TCP/IPv4	201
7.1.1	<i>GetNetBasicConfig</i>	201
7.1.2	<i>SetNetBasicConfig</i>	203
7.2	PPPoE	206
7.2.1	<i>GetNetPppoeConfig</i>	206
7.2.2	<i>SetNetPppoeConfig</i>	207
7.3	PORT	208
7.3.1	<i>GetPortConfig</i>	208
7.3.2	<i>SetPortConfig</i>	210
7.4	DDNS	211
7.4.1	<i>GetDdnsConfig</i>	211
7.4.2	<i>SetDdnsConfig</i>	213
8	SECURITY COMMANDS	215
8.1	USER MANAGEMENT	215
8.1.1	<i>ModifyPassword</i>	215
8.2	REBOOT	216
8.2.1	<i>Reboot</i>	216

9 TALKBACK COMMANDS.....	217
9.1 TALKBACK.....	217
9.1.1 <i>profile_talk</i>	217
9.1.2 <i>channel_talk</i>	219
10 SMART COMMANDS	221
10.1 FACE DETECT.....	221
10.1.1 <i>GetSmartVfdConfig</i>	221
10.1.2 <i>SetSmartVfdConfig</i>	223
10.1.3 <i>SearchSnapFaceByTime</i>	226
10.1.4 <i>SearchSnapFaceByKey</i>	228
10.2 INTRUSION	232
10.2.1 <i>GetSmartPerimeterConfig</i>	232
10.2.2 <i>SetSmartPerimeterConfig</i>	245
10.3 LINE CROSSING	256
10.3.1 <i>GetSmartTripwireConfig</i>	256
10.3.2 <i>SetSmartTripwireConfig</i>	267
10.4 OBJECT REMOVAL	275
10.4.1 <i>GetSmartOscConfig</i>	275
10.4.2 <i>SetSmartOscConfig</i>	278
10.5 EXCEPTION	280
10.5.1 <i>GetSmartAvdConfig</i>	280
10.5.2 <i>SetSmartAvdConfig</i>	281
10.5.3 <i>GetSmartAsdConfig</i>	283
10.5.4 <i>SetSmartAsdConfig</i>	285
10.6 REGION ENTRANCE	287
10.6.1 <i>GetSmartAoiEntryConfig</i>	287
10.6.2 <i>SetSmartAoiEntryConfig</i>	296
10.7 REGION LEAVE	304
10.7.1 <i>GetSmartAoiLeaveConfig</i>	304
10.7.2 <i>SetSmartAoiLeaveConfig</i>	305
10.8 TARGET COUNTING.....	306
10.8.1 <i>GetSmartPassLineCountConfig</i>	306
10.8.2 <i>SetSmartPassLineCountConfig</i>	316
10.12.3 <i>GetPassLineCountStatistics</i>	324
10.9 HEAT MAP.....	326
10.9.1 <i>GetSmartHeatMapConfig</i>	326
10.9.2 <i>SetSmartHeatMapConfig</i>	334
10.10 REGION STATISTICS.....	343
10.10.1 <i>GetSmartTrafficConfig</i>	343
10.10.2 <i>SetSmartTrafficConfig</i>	351
10.10.3 <i>GetTrafficCountStatistics</i>	359
10.11 VIDEO METADATA DETECTION	361
10.11.1 <i>GetSmartVsdConfig</i>	361
10.11.2 <i>SetSmartVsdConfig</i>	375
10.12 ILLEGAL PARKING DETECTION	389
10.12.1 <i>GetSmartPvdConfig</i>	389
10.12.2 <i>SetSmartPvdConfig</i>	397
10.13 LOITERING DETECTION	404
10.13.1 <i>GetSmartLoiteringConfig</i>	404
10.13.2 <i>SetSmartLoiteringConfig</i>	411
10.14 CROWD GATHERING DETECTION	417
10.14.1 <i>GetSmartCrowdGatheringConfig</i>	417

10.14.2 <i>SetSmartCrowdGatheringConfig</i>	420
10.15 FIRE	424
10.15.1 <i>GetSmartFireConfig</i>	424
10.15.2 <i>SetSmartFireConfig</i>	425
10.16 TEMPERATURE MEASUREMENT.....	426
10.16.1 <i>GetTemperatureAlarmConfig</i>	426
10.16.2 <i>SetTemperatureAlarmConfig</i>	433
10.16.3 <i>GetDotTemperature</i>	439
11 SCHEDULE COMMANDS	441
11.1 SCHEDULE	441
11.1.1 <i>GetScheduleConfig</i>	441
11.1.2 <i>SetScheduleConfig</i>	445
ANNEX A.....	447
A.1 FTP CUSTOM FILE NAME	447
A.2 CHANGE LOG	447

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. The web server on the IP media devices processes the request and sends the response back to the client application. The HTTP requests 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 allowed in this specification.

host: The host field refer to the host name, 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 is located at the IP device listening for TCP connections. If the port is empty or not given, the default 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 in the IP devices. For a video management system or client application that uses persistent connection 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 Access according to RFC 2617.

Authorization header field needs to be sent along with each request, and if a user is authenticated, the request will follow the normal execution flow. For the request with no authentication credentials, unauthorized HTTP 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 YWRtaW46MTIzNDU2
```

```
...
```

2. The device responses the following to a request with 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. The Content-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="2.0.0" 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" errorDesc="Invalid Request"/>

The detailed "errorCode" will be described in the following section.

1.3.6 Error Code

Error Code	Description
0	Successful.
1	"Invalid Request": The "cmd name" in the request URL does not exist or is not supported by this device.
2	"Invalid XML Format": The entity's XML format is not recognized by the system.
3	"Invalid XML Content": The parameter content carried in XML cannot be recognized or processed by the system.
4	Permission Denied. No permission to execute this API.
5	Network limit. The client's request may be restricted by a blacklist or whitelist.
6	Sensor ID Error. Invalid sensor ID.
7	System Busy. The system is undergoing upgrades, importing configurations, etc. It is currently unable to provide services and can be tried again later.
8	Password Expired. You can change your password and try again.
9	Unauthorized. The username or password is incorrect.
10	User Locked. The user has been locked.
11	Unsupport Function. Function not supported.
12	Channel Error. Invalid channel ID.
13	SD Error. SD card status error.
14	Action Error. The "action name" in the URL is invalid.
15	Missing Required Parameters.
16	Range Error. The parameter value is out of range.

Error Code	Description
17	Service Not Enabled. API service not enabled.
18	Modification not allowed. Modifying the configuration will cause the system to restart.
19	Over Specifications. Exceeding system specifications.
79	Internal Error.Device processing error.
80	Upgrade Error. Device upgrade processing error.
81	Upgrade Version Same. The software version of the current device is the same as the upgrade target version.
82	Upgrade Package Error. The upgrade package does not match the device.
83	Upgrade Signature Error. Upgrade package signature verification failed.
84	Upgrade Incompatible. The upgrade package version is incompatible and does not support upgrading from the current version to the target version.
101	Audio Not Working. The audio of the device is not enabled.
102	Audio Param Error. The audio file length is 0 or does not exist.
103	Audio Not PCM. The voice file is not PCM encoded.
104	Audio Not Wave. The voice file is not in wave format.
105	Audio Sampling Rate Error. The sampling rate is not 8000.
106	Audio Save Fail. Failed to save audio file to Flash.
107	Audio File Over Limit .The number of audio files exceeds the maximum limit.
108	Audio File Too Large. The audio file size exceeds the limit
109	Audio File Not Exist.

Error Code	Description
110	Audio Alarming. The system is playing an alarm voice and currently cannot support trial listening voice.
111	Audio File Occupied. The voice file is in use and cannot be deleted.
150	General errors in face processing.
151	Exceeding the maximum quantity limit of the face sample library.
152	The picture format of the face is not supported.
153	There is no face or more than one face in the picture.
154	The picture is too large.
155	The same face already exists in the face sample library.
156	The face does not exist.
157	The face group already exists.
158	Exceeding the maximum quantity limit of the face groups.
159	The face group does not exist.
201	General errors in license plate processing.
202	Exceeding the maximum quantity limit of the license plate library.
203	The same license plate already exists in the license plate sample library.
204	The license plate not exist.
205	The license plate group already exists.
206	Exceeding the maximum quantity limit of the license plate groups
207	The license plate does not exist.

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++.
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++.

Type	Description
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, four more attributes "minLen", "maxLen", "minCharNum" and "maxCharNum" are optional, which mean the minimum and maximum length of the character string, and minimum and maximum number of characters 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>
```

```
<name type="string" maxCharNum="18"><![CDATA[123456789012345678]]> </name>
```

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 category

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

System commands.

Image commands.

PTZ commands.

Alarm commands.

Playback commands

Network commands.

Security commands.

Maintain commands.

Talkback commands

Smart commands

Schedule commands

1.4.5 Annotation

We use the comments syntax of XML "`<!-- xxx -->`" to describe the protocol, in the following format: `<!--Required/Optional/Dependent; Description-->`.

For "Required/Optional/Dependent", explain as follows:

Required/Optional/Dependent	Description

Required	Required field. If a field marked as required is missing from the client to device request message, the API may not function correctly.
Optional	Optional field. If the request message sent by the client to the device does not carry optional fields, the device will handle it in the default way; If an optional field is carried but not supported by the device, the device should ignore it.
Dependent	This field depends on the values of other fields to determine whether it needs to be carried.

Note: Optional XML structures may contain required fields, meaning that the entire XML structure can be omitted, but if the XML structure appears, it must carry the required fields.

For "Description", it can include information such as functionality, value range, and so on. For example:

```
<mac><!--Optional; string, mac address of the device,like: 00:18:ae:98:38:fd--></mac>.
```

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/locate themselves. An UPnP compatible device will automatically announce its network address supported devices and services types when connected to a network, therefore becoming "plug-and-play" by allowing clients recognize those information and begin using this device immediately.

The UPnP architecture supports zero-configuration networking, and the device can dynamically join a network, obtain IP address, announce its name, convey its capabilities upon request, and gets the on-line status and capabilities of other devices. DHCP and DNS servers are optional and are only used if they are available on the network. Devices can leave the network

automatically without leaving any unwanted status information behind. UPnP was published as a 73-part International Standard, ISO/IEC 29341, in December, 2008 [6][7][8].

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

2 System commands

2.1 Device Information

2.1.1 GetSupportedAPIs

GetSupportedAPIs	
Description	To get the list of APIs supported by the device.
Typical URL	POST or GET http://<host>[:port]/GetSupportedAPIs
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<applicationInterfaces>
Applicable products	IPC,NVR
<applicationInterfaces> XML Block: <config version="" xmlns="http://www.ipc.com/ver10">	

```

<applicationInterfaces type="list" count=""> <! --Required; List of all API names supported by the
device-->

  <itemType type="string" maxLen=""/><! --Required; Define the data type of each item as string.
  -->

  <item><! --Required; The quantity of "item" is the same as the value of "count". --></item>

</applicationInterfaces>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <applicationInterfaces type="list" count="99">
    <itemType type="string" maxLen="63"/>
    <item><![CDATA[GetSupportedAPIs]]></item>
    <item><![CDATA[GetAlarmInInfo]]></item>
    <item><![CDATA[GetAlarmOutList]]></item>
    <item><![CDATA[GetAlarmStatus]]></item>
    <item><![CDATA[GetChannelInfo]]></item>
    <item><![CDATA[GetDeviceDetail]]></item>
    <item><![CDATA[GetDeviceInfo]]></item>
    <item><![CDATA[GetDiskInfo]]></item>
    <item><![CDATA[GetDotTemperature]]></item>
    <item><![CDATA[SetScheduleConfigEx]]></item>
    <item><![CDATA[ModifyPassword]]></item>
    <item><![CDATA[PtzGetCruises]]></item>
    <item><![CDATA[SetDateAndTime]]></item>
    <item><![CDATA[GetDateAndTime]]></item>
    <item><![CDATA[GetRecordStatusInfo]]></item>
    <item><![CDATA[GetRecordType]]></item>
    <item><![CDATA[SearchByTime]]></item>
    <item><![CDATA[SearchRecordDate]]></item>
    <item><![CDATA[SetScheduleConfig]]></item>
    <item><![CDATA[GetScheduleConfig]]></item>
    <item><![CDATA[SetAudioStreamConfig]]></item>
  </applicationInterfaces>
</config>

```

```
<item><![CDATA[GetAudioStreamConfig]]></item>
<item><![CDATA[SetImageConfig]]></item>
<item><![CDATA[GetImageConfig]]></item>
<item><![CDATA[SetImageOsdConfig]]></item>
<item><![CDATA[GetImageOsdConfig]]></item>
<item><![CDATA[SetPrivacyMaskConfig]]></item>
<item><![CDATA[GetPrivacyMaskConfig]]></item>
<item><![CDATA[GetStreamCaps]]></item>
<item><![CDATA[SetVideoStreamConfig]]></item>
<item><![CDATA[GetVideoStreamConfig]]></item>
<item><![CDATA[RequestKeyFrame]]></item>
<item><![CDATA[AddCustomizeAudioAlarm]]></item>
<item><![CDATA[AlarmOutputControl]]></item>
<item><![CDATA[AuditionCustomizeAudioAlarm]]></item>
<item><![CDATA[DeleteCustomizeAudioAlarm]]></item>
<item><![CDATA[GetAlarmInConfig]]></item>
<item><![CDATA[SetAlarmInConfig]]></item>
<item><![CDATA[GetAlarmOutConfig]]></item>
<item><![CDATA[SetAlarmOutConfig]]></item>
<item><![CDATA[GetAlarmTriggerConfig]]></item>
<item><![CDATA[SetAudioAlarmOutConfig]]></item>
<item><![CDATA[GetAudioAlarmOutConfig]]></item>
<item><![CDATA[GetDisarmingConfig]]></item>
<item><![CDATA[SetDisarmingConfig]]></item>
<item><![CDATA[SetAlarmServerConfig]]></item>
<item><![CDATA[GetAlarmServerConfig]]></item>
<item><![CDATA[GetMotionConfig]]></item>
<item><![CDATA[SetMotionConfig]]></item>
<item><![CDATA[SetSmartAsdConfig]]></item>
<item><![CDATA[GetSmartAsdConfig]]></item>
<item><![CDATA[SetSmartAvdConfig]]></item>
<item><![CDATA[GetSmartAvdConfig]]></item>
```

```
<item><![CDATA[GetScheduleConfig]]></item>
<item><![CDATA[ManualAlarmOut]]></item>
<item><![CDATA[SetScheduleConfig]]></item>
<item><![CDATA[SetAlarmTriggerConfig]]></item>
<item><![CDATA[GetAlarmTriggerConfig]]></item>
<item><![CDATA[GetScheduleConfig]]></item>
<item><![CDATA[GetPassLineCountStatistics]]></item>
<item><![CDATA[GetSmartPassLineCountConfig]]></item>
<item><![CDATA[SetSmartPassLineCountConfig]]></item>
<item><![CDATA[GetSmartTrafficConfig]]></item>
<item><![CDATA[SetSmartTrafficConfig]]></item>
<item><![CDATA[GetTrafficCountStatistics]]></item>
<item><![CDATA[GetSmartAoiEntryConfig]]></item>
<item><![CDATA[SetSmartAoiEntryConfig]]></item>
<item><![CDATA[GetSmartAoiLeaveConfig]]></item>
<item><![CDATA[SetSmartAoiLeaveConfig]]></item>
<item><![CDATA[SetSmartHeatMapConfig]]></item>
<item><![CDATA[GetSmartHeatMapConfig]]></item>
<item><![CDATA[SetSmartLoiteringConfig]]></item>
<item><![CDATA[GetSmartLoiteringConfig]]></item>
<item><![CDATA[SetSmartPvdConfig]]></item>
<item><![CDATA[GetSmartPvdConfig]]></item>
<item><![CDATA[SetSmartOscConfig]]></item>
<item><![CDATA[GetSmartOscConfig]]></item>
<item><![CDATA[GetSmartPerimeterConfig]]></item>
<item><![CDATA[SetSmartPerimeterConfig]]></item>
<item><![CDATA[SetSmartTripwireConfig]]></item>
<item><![CDATA[GetSmartTripwireConfig]]></item>
<item><![CDATA[GetSmartVsdConfig]]></item>
<item><![CDATA[SetSmartVsdConfig]]></item>
<item><![CDATA[SetDdnsConfig]]></item>
<item><![CDATA[GetDdnsConfig]]></item>
```

```

<item><![CDATA[SetNetBasicConfig]]></item>
<item><![CDATA[GetNetBasicConfig]]></item>
<item><![CDATA[SetNetPppoeConfig]]></item>
<item><![CDATA[GetNetPppoeConfig]]></item>
<item><![CDATA[SetPortConfig]]></item>
<item><![CDATA[GetPortConfig]]></item>
<item><![CDATA[ModifyIntegrateUser]]></item>
<item><![CDATA[Reboot]]></item>
<item><![CDATA[UpdateSliceFirmware]]></item>
<item><![CDATA[UpdateState]]></item>
<item><![CDATA[Talkback]]></item>
<item><![CDATA[GetAudioStreamConfig]]></item>
<item><![CDATA[GetSnapshot]]></item>
<item><![CDATA[GetSnapshotByTime]]></item>
</applicationInterfaces>
</config>

```

[Tips]: According to the "API name" in each<item>**API name**</item>, detailed descriptions can be found in the protocol document.

2.1.2 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
Entity Data	None
Successful Response	<deviceInfo>
Applicable products	IPC,NVR

<deviceInfo> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <deviceInfo <!--Required.-->
    <deviceName type="string"><!--Required; string.--></deviceName>
    <deviceDescription type="string"><!--Optional,string.--></deviceDescription>
    <apiVersion type="string">
      <!--Required; string ,the version of API protocol supported by the device.-->
    </apiVersion>
    <httpPostVersion type="string">
      <!--Optional; string ,the version of Http post protocol supported by the device.-->
    </httpPostVersion>
    <softwareVersion type="string"><!--Required; string.--></softwareVersion>
    <softwareBuildDate type="string">
      <!--Required;string, The compilation date of the software.-->
    </softwareBuildDate>
    <kernelVersion type="string"><!--Required; string.--></kernelVersion>
    <hardwareVersion type="string"><!--Required; string.--></hardwareVersion>
    <model type="string"><!--Required; string, product model.--></model>
    <brand type="string"><!--Required; string, brand name.--></brand>
    <mac type="string"><!--Required; string, MAC address of the device.--></mac>
    <sn type="string"><!--Optional; string, serial number.--></sn>
    <chlMaxCount type="uint32">
      <!--Required; Number of channels, value greater than or equal to 1. -->
    </chlMaxCount>
    <audioInCount type="uint32"><!--Optional; the quantity of audio inputs. --></audioInCount>
    <audioOutCount type="uint32">
      <!--Optional; the quantity of audio inputs. -->
    </audioOutCount>
    <alarmInCount type="uint32"><!--Optional; the quantity of alarm in. --></alarmInCount>
    <alarmOutCount type="uint32"><!--Optional; the quantity of alarm out. --></alarmOutCount>
    <integratedPtz type="boolean">
      <!--Optional; boolean, support dome camera PTZ function. -->
    </integratedPtz>
  </deviceInfo>
</config>
```

```
</integratedPtz>
<supportRS485Ptz type="boolean">
    <!--Optional; boolean, support external RS485 PTZ function or not. -->
</supportRS485Ptz>
<supportSDCard type="boolean">
    <!--Optional; boolean, support SD card or not. -->
</supportSDCard>
</deviceInfo>
</config>
```

Example of Successful Response(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <deviceInfo>
    <deviceName type="string"><![CDATA[IPC]]></deviceName>
    <deviceDescription type="string"><![CDATA[E4_4MP_]]></deviceDescription>
    <apiVersion type="string"><![CDATA[2.0.0]]></apiVersion>
    <softwareVersion type="string">
      <![CDATA[5.3.0.12288B240820.IG1.U1(08A10).beta]]>
    </softwareVersion>
    <softwareBuildDate type="string"><![CDATA[2024-10-31]]></softwareBuildDate>
    <kernelVersion type="string"><![CDATA[20241010]]></kernelVersion>
    <hardwareVersion type="string"><![CDATA[1.5]]></hardwareVersion>
    <model type="string"><![CDATA[E4_4MP_]]></model>
    <brand type="string"><![CDATA[Customer]]></brand>
    <mac type="string"><![CDATA[70:ab:c8:1b:5d:dc]]></mac>
    <sn type="string"><![CDATA[2E323D9463D5]]></sn>
    <chlMaxCount type="uint32">1</chlMaxCount>
    <audioInCount type="uint32">2</audioInCount>
    <audioOutCount type="uint32">1</audioOutCount>
    <alarmInCount type="uint32">1</alarmInCount>
    <alarmOutCount type="uint32">1</alarmOutCount>
    <integratedPtz type="boolean">false</integratedPtz>
```

```
<supportRS485Ptz type="boolean">false</supportRS485Ptz>
<supportSDCard type="boolean">true</supportSDCard>
</deviceInfo>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <deviceInfo>
    <deviceName type="string">
      <![CDATA[Device Name]]>
    </deviceName>
    <apiVersion type="string">
      <![CDATA[2.0.0]]>
    </apiVersion>
    <model type="string">
      <![CDATA[TD-3308H1-8P-B1]]>
    </model>
    <brand type="string">
      <![CDATA[IP CAM]]>
    </brand>
    <deviceDescription type="string">
      <![CDATA[TD-3308H1-8P-B1]]>
    </deviceDescription>
    <audioInCount type="uint32">0</audioInCount>
    <audioOutCount type="uint32">1</audioOutCount>
    <integratedPtz type="boolean">true</integratedPtz>
    <supportRS485Ptz type="boolean">true</supportRS485Ptz>
    <supportSDCard type="boolean">false</supportSDCard>
    <alarmInCount type="uint32">16</alarmInCount>
    <alarmOutCount type="uint32">4</alarmOutCount>
    <softwareVersion type="string">
```

```

        <![CDATA[1.4.12.71946B240815.N0W.U1(8A418).beta]]>
    </softwareVersion>
    <softwareBuildDate type="string">
        <![CDATA[1.4.12.69452B240510.N2P.U1(16A840).beta]]>
    </softwareBuildDate>
    <kernelVersion type="string"><![CDATA[N8G8-N38C-O5A5]]></kernelVersion>
    <hardwareVersion type="string"><![CDATA[300112-V1]]></hardwareVersion>
    <mac type="string">
        <![CDATA[00:18:ae:00:88:15]]>
    </mac>
    <sn type="string"><![CDATA[N018AEA8A9C2]]></sn>
    <chlMaxCount type="uint32">8</chlMaxCount>
</deviceInfo>
</config>

```

[Tips]:

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

2.1.3 GetDiskInfo

GetDiskInfo	
Description	To get the IP media device's disk information.
Typical URL	POST or GET http://<host>[:port]/GetDiskInfo
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<diskInfo>
Applicable products	IPC,NVR

<diskInfo> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <types><!--Required; enumeration type definition.-->
    <diskStatus><!--Required; define the status of the disk.-->
      <enum>read</enum>
      <enum>read/write</enum>
      <enum>unformat</enum>
      <enum>formatting</enum>
      <enum>exception</enum>
      <enum>locked</enum>
    </diskStatus>
    <diskType><!--Required; define the type of the disk.-->
      <enum>SD</enum><!--Optional; SD Card(Secure Digital Card).-->
      <enum>HDD</enum><!--Optional; Hard Disk Drive.-->
    </diskType>
  </types>
  <diskInfo type="list" count=""><!--Required; Disk Information List.-->
    <item> <!--Required.-->
      <id type="string"><!--Required; string, identification of disk.--></id>
      <totalSpace type="uint32"><!--Required; uint32, Total disk space size.--></totalSpace>
      <freeSpace type="uint32"><!--Required; uint32, Free disk space size.--></freeSpace>
      <imageFreeSpace type="uint32">
        <!--Optional(NVR is not supported); Free space size for storing images.-->
      </imageFreeSpace>
      <status type="diskStatus"><!--Required; diskStatus, The status of the disk.--></status>
      <storageType type="diskType">
        <!--Optional; diskType, type of storage media.-->
      </storageType>
    </item>
  </diskInfo>
</config>
```

Example of Successful Response(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <diskStatus>
      <enum>read</enum>
      <enum>read/write</enum>
      <enum>unformat</enum>
      <enum>formatting</enum>
      <enum>exception</enum>
      <enum>locked</enum>
    </diskStatus>
  </types>
  <diskInfo type="list" count="1">
    <item>
      <id type="string"><![CDATA[disk1]]></id>
      <totalSpace type="uint32">30371</totalSpace>
      <freeSpace type="uint32">0</freeSpace>
      <imageFreeSpace type="uint32">2985</imageFreeSpace>
      <status type="diskStatus">read/write</status>
      <storageType type="diskType">SD</storageType>
    </item>
  </diskInfo>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <diskStatus>
      <enum>read</enum>
      <enum>read/write</enum>
```

```

        <enum>unformat</enum>
        <enum>locked</enum>
    </diskStatus>
</types>
<diskInfo type="list" count="1">
    <item>
        <id><![CDATA[{7665642F-732F-6264-0000-000000000000}]]></id>
        <totalSpace>29560</totalSpace>
        <freeSpace>0</freeSpace>
        <diskStatus>unformat</diskStatus>
        <storageType type="diskType">HDD</storageType>
    </item>
</diskInfo>
</config>

```

[Tips]:

The "totalSpace" and "freeSpace" are in MB.

There is empty "diskInfo" node if there is no disk on device.

The enums, "read", "read/write" and "unformat", are supported by NVR and DVR.

The enums, "read/write", "unformat", "formatting" and "exception", are supported by IPC.

The "imageFreeSpace" is supported by IPC only.

2.1.4 GetChannelInfo

GetChannelInfo	
Description	To get the IP media device's channel list.
Typical URL	POST or GET http://<host>[:port]/GetChannelInfo
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<channelList>

Applicable products

IPC,NVR

<channelList> XML Block:

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

```
  <types> <!--Required; enumeration type definition.-->
```

```
    <channelStatus><!--Required; define the status of the channel.-->
```

```
      <enum>online</enum>
```

```
      <enum>offline</enum>
```

```
      <enum>videoOn</enum>
```

```
      <enum>videoLoss</enum>
```

```
    </channelStatus>
```

```
    <channelType><!--Optional; define the type of the channel.-->
```

```
      <enum>Normal</enum>
```

```
      <enum>Thermal</enum>
```

```
      <enum>Fisheye</enum>
```

```
      <enum>Panoramic</enum>
```

```
      <enum>PTZ</enum>
```

```
      <enum>4PTZFusion</enum>
```

```
    </channelType>
```

```
  </types>
```

```
  <channelList type="list" count=""> <!--Required; channel List.-->
```

```
    <item> <!--Required; The quantity of "item" is the same as the value of "count". -->
```

```
      <channelId type="uint32"><!--Required; channel Id.--></channelId>
```

```
      <name type="string"><!--Required; string, channel name.--></name>
```

```
      <status type="channelStatus"><!--Required; channel work status.--></status>
```

```
      <attribute type="channelType">
```

```
        <!--Optional(NVR is not supported); channel work status.-->
```

```
      </attribute>
```

```
    </item>
```

```
  </channelList>
```

```
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <channelStatus>
      <enum>online</enum>
      <enum>offline</enum>
      <enum>videoOn</enum>
      <enum>videoLoss</enum>
    </channelStatus>
    <channelType>
      <enum>Normal</enum>
      <enum>Thermal</enum>
      <enum>Fisheye</enum>
      <enum>Panoramic</enum>
      <enum>PTZ</enum>
      <enum>4PTZFusion</enum>
    </channelType>
  </types>
  <channelList type="list" count="1">
    <item>
      <channelId type="uint32">1</channelId>
      <name type="string"><![CDATA[channel1]]></name>
      <status type="channelStatus">online</status>
      <attribute type="channelType">Normal</attribute>
    </item>
  </channelList>
</config>
```

[Tips]:

1. For IPC: The count of channelList should be the same as the field "chlMaxCount" returned by "GetDeviceInfo". For NVR: It is the number of channels configured on the NVR.
2. Channel id starts from 1.

2.1.5 GetAlarmInInfo

GetAlarmInInfo	
Description	To get the IP media device's alarmin list.
Typical URL	POST or GET http://<host>[:port]/GetAlarmInInfo
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<alarmInInfoList>
Applicable products	IPC,NVR
<p><alarmInInfoList> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <!--Required; enumeration type definition.--> <alarmType> <!--Required; define the type of the alarmIn.--> <enum>local</enum> <enum>virtual</enum><!--Only valid for NVR.--> <enum>remote</enum><!--Only valid for NVR.--> </alarmType> </types> <alarmInInfoList type="list" count=""> <!--Required; alarm in list.--> <item><!--Required; The quantity of "item" is the same as the value of "count". --> <id type="uint32"><!--Required; identification of alarm in--></id> <alarmInType type="alarmType"><!--Required; type of alarm in.--></alarmInType> </item> </alarmInInfoList> </config> </pre>	

GetAlarmInInfo

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <alarmType>
      <enum>local</enum>
      <enum>virtual</enum>
      <enum>remote</enum>
    </alarmType>
  </types>
  <alarmInInfoList type="list" count="1">
    <item>
      <id type="uint32">1</id>
      <alarmInType type="alarmType">local</alarmInType>
    </item>
  </alarmInInfoList>
</config>
```

[Tips]:

1. For IPC, the alarm type is local.
2. For NVR, the local alarm in is local, and "remote" indicates the alarm in of the IPC that has been connected to the NVR. and "virtual" represents a virtual alarm in.
3. "id" starts from 1.

2.1.6 GetAlarmOutInfo

GetAlarmOutInfo	
Description	To get the IP media device's alarmout list.
Typical URL	POST or GET http://<host>[:port]/GetAlarmOutInfo
Channel ID	None

Action name	None
Entity Data	None
Successful Response	<alarmOutInfoList>
Applicable products	IPC,NVR

<alarmOutInfoList> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <alarmOutInfoList type="list" count=""><!--Required; alarm out list.-->
    <item><! --Required; The quantity of "item" is the same as the value of "count". -->
      <id type="uint32"><!--Required; identification of alarm out--></id>
    </item>
  </alarmOutInfoList>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmOutInfoList type="list" count="2">
    <item>
      <id type="uint32">1</id>
    </item>
    <item>
      <id type="uint32">2</id>
    </item>
  </alarmOutInfoList >
</config>
```

[Tips]:

1. "id" starts from 1.

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 GET http://<host>[:port]/GetDateAndTime
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<time>
Applicable products	IPC,NVR
<time> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <types> <!--Required; enumeration type definition.--> <synchronizeType><!--Required--> <enum>manually</enum> <enum>NTP</enum> </synchronizeType> <daylightType><!--Required; types of daylight saving time.--> <enum>offDaylight</enum> <enum>autoDaylight</enum> <enum>manuallyDaylight</enum> <!--Optional;NVR not supported.--> </daylightType> </types> <time><!--Required.--> <timezoneInfo><!--Required. Time zone info.--> <timeZone type="string"> <!--Required;POSIX time zone string. Refer to Tips for details --></pre>	

```

</timeZone>
<daylightMode type="daylightType">
    <!--Required. Daylight saving time mode.-->
</daylightMode>
<startTime type="string" maxLen="64">
    <!--Dependent. When 'daylightMode' is set to 'manuallyDaylight', it indicates the start
    time of daylight saving time.-->
</startTime>
<endTime type="string" maxLen="64">
    <!--Dependent. When 'daylightMode' is set to 'manuallyDaylight', it indicates the end
    time of daylight saving time.-->
</endTime>
<offset type="uint16" min="30" max="120">
    <!--Dependent. When 'daylightMode' is set to 'manually daylight', it indicates the
    offset time of daylight saving time.-->
</offset>
</timezoneInfo>
<synchronizeInfo> <!--Required, time synchronization information.-->
    <type type="synchronizeType"><!--Required.--></type>
    <ntpServer type="string" maxLen="">
        <!--Dependent. Valid when type='NTP', NTP server address. -->
    </ntpServer>
    <ntpSyncInterval type="uint32" min="" max="">
        <!--Dependent. Valid when type='NTP', NTP synchronization interval, Unit: minutes.
        -->
    </ntpSyncInterval>
    <currentTime type="string"><!--Required, current device time.--></currentTime>
</synchronizeInfo>
</time>
</config>

```

Example of Successful Response(IPC):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">

```

```
<types>
  <synchronizeType>
    <enum>manually</enum>
    <enum>NTP</enum>
  </synchronizeType>
  <daylightType>
    <enum>offDaylight</enum>
    <enum>autoDaylight</enum>
    <enum>manuallyDaylight</enum>
  </daylightType>
</types>
<time>
  <timezoneInfo>
    <timeZone type="string"><![CDATA[CST-8]]></timeZone>
    <daylightMode type="daylightType">manuallyDaylight</daylightMode>
    <startTime type="string" maxLen="64">
      <![CDATA[M1.1.0/0]]>
    </startTime>
    <endTime type="string" maxLen="64">
      <![CDATA[M2.1.1/0]]>
    </endTime>
    <offset type="uint16" min="30" max="120">120</offset>
  </timezoneInfo>
  <synchronizeInfo>
    <type type="timeFormatModeType">NTP</type>
    <ntpServer type="string" maxLen="127"><![CDATA[time.windows.com]]></ntpServer>
    <ntpSyncInterval type="uint32" min="30" max="10080">1440</ntpSyncInterval>
    <currentTime type="string"><![CDATA[2024-08-21 15:05:31]]></currentTime>
  </synchronizeInfo>
</time>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <synchronizeType>
      <enum>manually</enum>
      <enum>NTP</enum>
    </synchronizeType>
    <daylightType>
      <enum>offDaylight</enum>
      <enum>autoDaylight</enum>
    </daylightType>
  </types>
  <time>
    <timezoneInfo>
      <timeZone type="string"><![CDATA[CST-8]]></timeZone>
      <daylightMode type="daylightType">autoDaylight</daylightMode>
    </timezoneInfo>
    <synchronizeInfo>
      <type type="timeFormatModeType">NTP</type>
      <ntpServer type="string" maxLen="127"><![CDATA[time.windows.com]]></ntpServer>
      <currentTime type="string"><![CDATA[2024-08-21 15:05:31]]></currentTime>
    </synchronizeInfo>
  </time>
</config>
```

[Tips]:

1. POSIX time zone string: STD<offset>DST<offset>,start[/time],end[/time].

example: "GMT0BST,M3.5.0/1,M10.5.0".

this time zone, standard time named GMT and daylight saving time named BST(British Summer Time), 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 01:00 and ends the last Sunday in October at 02:00.

2. Examples of <startTime> and <endTime>. M represents the month. e.g:

M5.1.0/5: May, first week, day 0 (Sunday), 5:00.

M8.2.2/23: August, second week, second day (Tuesday), 23:00.

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	< time >
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<time> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> < time ><!--Required.--> <timezoneInfo><!--Optional; Time zone info.--> <timeZone> <!--Optional;string. POSIX time zone string. Refer to Tips for details. --> </timeZone> <daylightMode> <!--Optional;daylightType, Daylight saving time mode.--> </daylightMode> <startTime> <!--Dependent;string, when 'daylightMode' is set to 'manuallyDaylight', it indicates the start time of daylight saving time. time format: "Month.Week.Day/Hour".eg. M1.1.0/5, For more details, please refer to Tips.--> </startTime> <endTime> <!--Dependent;string, when 'daylightMode' is set to 'manuallyDaylight', it indicates the end time of daylight saving time. time format: "Month.Week.Day/Hour".eg. M1.1.0/5, For more details, please refer to Tips.-->	

```
</endTime>
<offset>
    <!--Dependent;uint16,range[30,120],when 'daylightMode' is set to 'manually daylight',
    it indicates the offset time of daylight saving time.-->
</offset>
</timezoneInfo>
<synchronizeInfo> <!--Optional;time synchronization information.-->
    <type><!--Required; synchronizeType.--></type>
    <ntpServer>
        <!--Dependent; string,maxLen=127, domain name of NTP server, when 'type' is set to
        'NTP' ,the field 'ntpServer' needs to be carried. -->
    </ntpServer>
    <ntpSyncInterval>
        <!--Dependent. Valid when type='NTP', NTP synchronization interval, Unit: minutes.
        -->
    </ntpSyncInterval>
    <currentTime>
        <!--Optional; string, set the time for the device, time format: YYYY-MM-DD
        HH:MM:SS, 24-hour format.-->
    </currentTime>
</synchronizeInfo>
</time>
</config>
```

Example of Entity Data(for NTP):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <time>
    <timezoneInfo>
      <timeZone><![CDATA[CST-8]]></timeZone>
    </timezoneInfo>
    <synchronizeInfo>
      <type>NTP</type>
      <ntpServer><![CDATA[time.windows.com]]></ntpServer>
      <ntpSyncInterval>1440</ntpSyncInterval>
    </synchronizeInfo>
  </time>
</config>
```

Example of Entity Data(for manually):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <time>
    <timezoneInfo>
      <timeZone><![CDATA[CST-8]]></timeZone>
    </timezoneInfo>
    <synchronizeInfo>
      <type>manually</type>
      <currentTime><![CDATA[2024-08-21 15:05:31]]></currentTime>
    </synchronizeInfo>
  </time>
</config>
```

[Tips]:

1. POSIX time zone string: STD<offset>DST<offset>,start[/time],end[/time].

example: "GMT0BST,M3.5.0/1,M10.5.0".

this time zone, standard time named GMT and daylight saving time named BST(British Summer Time), 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 01:00 and ends the last Sunday in October at 02:00.

2. Examples of <startTime> and <endTime>. M represents the month. e.g:

M5.1.0/5: May, first week, day 0 (Sunday), 5:00.

M8.2.2/23: August, second week, second day (Tuesday), 23:00.

2.3 Upgrade

2.3.1 UpdateState

UpdateState	
Description	To set the IP media device's start or stop upgrade .
Typical URL	POST or GET http://<host>[:port]/UpdateState
Channel ID	None
Action name	None
Entity Data	<upgradeInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<upgradeInfo> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <upgradeInfo> <!--Required.--> <upgradeState> <!--Required;value value in {start,stop}, Request to start or stop the firmware upgrade of the device.--> </upgradeState> <md5sumBuffer>	

`<!--Dependent; string, MD5(upgrade packet), 32 length. When upgradeState is set to 'start', 'md5sumBuffer' needs to be included. -->`

`</md5sumBuffer>`

`</upgradeInfo>`

`</config>`

Example of Entity Data (start upgrade):

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

`<config version="2.0.0" xmlns="http://www.ipc.com/ver10">`

`<upgradeInfo>`

`<upgradeState>start</upgradeState>`

`<md5sumBuffer>`

`<![CDATA[68b8423b3535f97b88a7264b0870bca6]]>`

`</md5sumBuffer>`

`</upgradeInfo>`

`</config>`

Example of Entity Data (stop upgrade):

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

`<config version="2.0.0" xmlns="http://www.ipc.com/ver10">`

`<upgradeInfo>`

`<upgradeState>stop</upgradeState>`

`</upgradeInfo>`

`</config>`

[Tips]:

- 1.The upgrade must start with the upgradestate as start, the upgradepacketsize as the upgrade package size, and the md5sumbuffer as the md5sum value of the entire upgrade package file.
2. After sending the upgrade package file, you need to send the upgradestate as stop to end upgrading the firmware.
- 3.NVR is not supported.

2.3.2 UpdateSliceFirmware

UpdateSliceFirmware

Description	To update the IP media device's firmware, Recommended upgrade package fragment size 1M.
Typical URL	POST http://<host>[:port]/UpdateSliceFirmware
Channel ID	None
Action name	None
Entity Data	binary upgrade file.
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC

Example of Entity Data:

POST /UpdateSliceFirmware HTTP/1.1

Accept: */*

If-Modified-Since: 0

Authorization: Basic YWRtaW46MTIzNDU2

Content-Type: multipart/form-data; boundary=-----7e43865e10634

Accept-Language: zh-CN

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko

Host: 10.20.19.242

Content-Length: 536964

DNT: 1

Connection: Keep-Alive

Cache-Control: no-cache

-----7e43865e10634

Content-Disposition: form-data; name="file"; filename="blob"

Content-Type: application/octet-stream

binary upgrade file

-----7e43865e10634--

[Tips]:

1.NVR is not supported.

3 Image commands

3.1 Stream Capabilities

3.1.1 GetStreamCaps

GetStreamCaps	
Description	To get the IP media device's streams capabilities for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetStreamCaps[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1. The value of channelId should not be greater than chlMaxCount in GetDeviceInfo.
Action name	None
Entity Data	None
Successful Response	<streamList>
Applicable products	IPC,NVR
<streamList> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <encodeTypeE><!--Required;define video encoding types.--> <enum>h264</enum> <enum>h265</enum> <enum>h264plus</enum> <enum>h265plus</enum> <enum>h264smart</enum>	

```

        <enum>h265smart</enum>
        <enum>mjpeg</enum>
    </encodeTypeE>
    <encodeLevelE> <!--Required;define H.264/H.265 profile.-->
        <enum>baseLine</enum>
        <enum>mainProfile</enum>
        <enum>highProfile</enum>
    </encodeLevelE>
</types>
<streamList type="list" count=""><!--Required;channel stream information, usually including main
stream, sub stream and third stream.-->
    <item id="1"> <!--Required;id="1" indicates the main stream.-->
        <streamName type="string">
            <!--Required;string,the name of stream.-->
        </streamName>
        <resolutionCaps type="list" count=""><!--Required; List of supported resolutions.-->
            <itemType type="string"/>
            <item maxFrameRate=""><!--Optional; video width*height.--></item>
        </resolutionCaps>
        <encodeTypeCaps type="list" count=""><!--Required; List of supported encode types.-->
            <itemType type="encodeTypeE"/>
            <item>h264</item>
            <item>h265</item>
        </encodeTypeCaps>
        <encodeLevelCaps type="list" count="3">
            <!--Optional; List of profiles supported by H.264/H.265.-->
            <itemType type="encodeLevelE"/>
            <item>baseLine</item>
            <item>mainProfile</item>
            <item>highProfile</item>
        </encodeLevelCaps>
        <encodeLevelCaps265 type="list" count="">
            <!--Optional; list of supported h.265 profile.-->

```

```

        <itemType type="encodeLevelE"/>
        <item>mainProfile</item>
    </encodeLevelCaps265>
    <bitRateLists type="list" count=""><!--Optional; list of supported bit rate.-->
        <itemType type="uint32"/>
        <item>8192</item>
    </bitRateLists>
</item>
</streamList>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0"
  xmlns="http://www.ipc.com/ver10">
  <types>
    <encodeTypeE>
      <enum>h264</enum>
      <enum>h265</enum>
      <enum>h264plus</enum>
      <enum>h265plus</enum>
      <enum>h264smart</enum>
      <enum>h265smart</enum>
      <enum>mjpeg</enum>
    </encodeTypeE>
    <encodeLevelE>
      <enum>baseLine</enum>
      <enum>mainProfile</enum>
      <enum>highProfile</enum>
    </encodeLevelE>
  </types>
  <streamList type="list" count="3">
    <item id="1">

```

```
<streamName type="string">
  <![CDATA[profile1]]>
</streamName>
<resolutionCaps type="list" count="3">
  <itemType type="string"/>
  <item maxFrameRate="25"><![CDATA[2688x1520]]</item>
  <item maxFrameRate="25"><![CDATA[2560x1440]]</item>
  <item maxFrameRate="25"><![CDATA[1920x1080]]</item>
</resolutionCaps>
<encodeTypeCaps type="list" count="6">
  <itemType type="encodeTypeE"/>
  <item>h264</item>
  <item>h265</item>
  <item>h264plus</item>
  <item>h265plus</item>
  <item>h264smart</item>
  <item>h265smart</item>
</encodeTypeCaps>
<encodeLevelCaps265 type="list" count="1">
  <itemType type="encodeLevelE"/>
  <item>mainProfile</item>
</encodeLevelCaps265>
<encodeLevelCaps type="list" count="3">
  <itemType type="encodeLevelE"/>
  <item>baseLine</item>
  <item>mainProfile</item>
  <item>highProfile</item>
</encodeLevelCaps>
<bitRateLists type="list" count="5">
  <itemType type="uint32"/>
  <item>8192</item>
  <item>6144</item>
```

```
<item>3072</item>
<item>256</item>
<item>32</item>
</bitRateLists>
</item>
<item id="2">
  <streamName type="string">
    <![CDATA[profile2]]>
  </streamName>
  <resolutionCaps type="list" count="7">
    <itemType type="string"/>
    <item maxFrameRate="25"><![CDATA[1920x1080]]></item>
    <item maxFrameRate="25"><![CDATA[1280x720]]></item>
    <item maxFrameRate="25"><![CDATA[704x576]]></item>
    <item maxFrameRate="25"><![CDATA[640x480]]></item>
    <item maxFrameRate="25"><![CDATA[640x360]]></item>
    <item maxFrameRate="25"><![CDATA[480x240]]></item>
    <item maxFrameRate="25"><![CDATA[352x288]]></item>
  </resolutionCaps>
  <encodeTypeCaps type="list" count="5">
    <itemType type="encodeType"/>
    <item>h264</item>
    <item>h265</item>
    <item>h264plus</item>
    <item>h265plus</item>
    <item>mjpeg</item>
  </encodeTypeCaps>
  <encodeLevelCaps type="list" count="3">
    <itemType type="encodeLevel"/>
    <item>baseLine</item>
    <item>mainProfile</item>
    <item>highProfile</item>
  </encodeLevelCaps>
</item>
```

```
</encodeLevelCaps>
<encodeLevelCaps265 type="list" count="1">
  <itemType type="encodeLevelE"/>
  <item>mainProfile</item>
</encodeLevelCaps265>
<encodeLevelCaps type="list" count="3">
  <itemType type="encodeLevelE"/>
  <item>baseLine</item>
  <item>mainProfile</item>
  <item>highProfile</item>
</encodeLevelCaps>
<bitRateLists type="list" count="5">
  <itemType type="uint32"/>
  <item>8192</item>
  <item>6144</item>
  <item>3072</item>
  <item>256</item>
  <item>32</item>
</bitRateLists>
</item>
<item id="3">
  <streamName type="string">
    <![CDATA[profile3]]>
  </streamName>
  <resolutionCaps type="list" count="4">
    <itemType type="string"/>
    <item maxFrameRate="25"><![CDATA[1280x720]]></item>
    <item maxFrameRate="25"><![CDATA[704x576]]></item>
    <item maxFrameRate="25"><![CDATA[480x240]]></item>
    <item maxFrameRate="25"><![CDATA[352x288]]></item>
  </resolutionCaps>
  <encodeTypeCaps type="list" count="5">
```

```
<itemType type="encodeType"/>
<item>h264</item>
<item>h265</item>
<item>h264plus</item>
<item>h265plus</item>
<item>mjpeg</item>
</encodeTypeCaps>
<encodeLevelCaps type="list" count="3">
  <itemType type="encodeLevelE"/>
  <item>baseLine</item>
  <item>mainProfile</item>
  <item>highProfile</item>
</encodeLevelCaps>
<encodeLevelCaps265 type="list" count="1">
  <itemType type="encodeLevelE"/>
  <item>mainProfile</item>
</encodeLevelCaps265>
<encodeLevelCaps type="list" count="3">
  <itemType type="encodeLevelE"/>
  <item>baseLine</item>
  <item>mainProfile</item>
  <item>highProfile</item>
</encodeLevelCaps>
<bitRateLists type="list" count="5">
  <itemType type="uint32"/>
  <item>8192</item>
  <item>6144</item>
  <item>3072</item>
  <item>256</item>
  <item>32</item>
</bitRateLists>
</item>
```

<pre></streamList> </config></pre>
<p>[Tips]:</p> <p>The "count=3" means the channel supports 3 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 of NVR/DVR by the following URL.</p> <pre>rtsp://<host><:port>?chID=<channelId>&streamType=<streamType></pre> <p>"streamtype" can be main or sub</p> <p>The client application, can obtain the specific stream of IPC by the following URL.</p> <pre>rtsp://<host><:port>/<streamName></pre> <p>The "resolutionCaps" announces optional combinations for frame rate and resolution. The "encodeTypeCaps" announces optional compression types. The "encodeLevelCaps" optional compression levels.</p> <p>The "id" attribute for each item starts from "1".</p>

3.2 Image

3.2.1 GetImageConfig

GetImageConfig	
Description	To get the IP media device's image configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetImageConfig[/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	<config>
Applicable products	IPC,NVR
<p><config>XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <types></pre>	

```
<configFileType><!--Optional;define config file types.-->
    <enum>normal</enum>
    <enum>day</enum>
    <enum>night</enum>
</configFileType>
<frequency> <!--Optional;define video formats.NVR is not supported.-->
    <enum>60HZ</enum>
    <enum>50HZ</enum>
</frequency>
<imageRotaoMode><!--Optional;define corridor mode angle.-->
    <enum>0</enum>
    <enum>90</enum>
    <enum>180</enum>
    <enum>270</enum>
</imageRotaoMode>
<IRCutMode> <!--Optional;define IR-CUT mode.-->
    <enum>auto</enum>
    <enum>day</enum>
    <enum>night</enum>
    <enum>time</enum>
    <enum>alarmInLink</enum>
</IRCutMode>
<dayNightMode> <!--Optional;define IR-CUT day, night mode.-->
    <enum>day</enum>
    <enum>night</enum>
</dayNightMode>
<IRCutConvSen><!--Optional;define IR-CUT sensitivity.-->
    <enum>high</enum>
    <enum>mid</enum>
    <enum>low</enum>
</IRCutConvSen>
<whitebalanceMode> <!--Optional;define white balance mode.-->
```

```
<enum>auto</enum>
<enum>indoor</enum>
<enum>outdoor</enum>
<enum>manual</enum>
</whitebalanceMode>
<BLCMode><!--Optional;define back light compensation mode.-->
  <enum>OFF</enum>
  <enum>HWDR</enum>
  <enum>HLC</enum>
  <enum>BLC</enum>
</BLCMode>
<HWDRLevel><!--Optional;define HWDR level.-->
  <enum>low</enum>
  <enum>medium</enum>
  <enum>high</enum>
</HWDRLevel>
<infraredModeE> <!--Optional;define infrared light mode.-->
  <enum>on</enum>
  <enum>off</enum>
  <enum>auto</enum>
</infraredModeE>
<gainMode><!--Optional;define antiflicker mode.-->
  <enum>auto</enum>
  <enum>manual</enum>
</gainMode>
<smartIRMode><!--Optional;define smartIR mode.-->
  <enum>off</enum>
  <enum>manual</enum>
  <enum>auto</enum>
</smartIRMode>
<HWfogType><!--Optional;define Hardware fog mode.-->
  <enum>auto</enum>
```

```
<enum>manual</enum>
<enum>off</enum>
</HWfogType>
<autoExposureMode> <!--Optional;define exposure mode, valid for dome camera.-->
  <enum>auto</enum>
  <enum>manual</enum>
</autoExposureMode>
<autoExposureValue> <!--Optional;define exposure value enum, valid for dome camera.-->
  <enum>1/3</enum>
  <enum>1/6</enum>
  <enum>1/12</enum>
  <enum>1/20</enum>
  <enum>1/25</enum>
  <enum>1/50</enum>
  <enum>1/75</enum>
  <enum>1/100</enum>
  <enum>1/150</enum>
  <enum>1/200</enum>
  <enum>1/250</enum>
  <enum>1/300</enum>
  <enum>1/500</enum>
  <enum>1/750</enum>
  <enum>1/1000</enum>
  <enum>1/2000</enum>
  <enum>1/4000</enum>
  <enum>1/10000</enum>
  <enum>1/100000</enum>
</autoExposureValue>
<shutterValue> <!--Optional;define shutter value, valid for normal camera.-->
  <enum>1/3</enum>
  <enum>1/6</enum>
  <enum>1/12</enum>
```

```
<enum>1/20</enum>
<enum>1/25</enum>
<enum>1/50</enum>
<enum>1/75</enum>
<enum>1/100</enum>
<enum>1/150</enum>
<enum>1/200</enum>
<enum>1/250</enum>
<enum>1/300</enum>
<enum>1/500</enum>
<enum>1/750</enum>
<enum>1/1000</enum>
<enum>1/2000</enum>
<enum>1/4000</enum>
<enum>1/10000</enum>
<enum>1/100000</enum>
</shutterValue>
</types>
<cfgFile type="configFileType" default="">
  <!--Optional; IPC supports 3 sets of image configuration files, NVR not supported.-->
</cfgFile>
<image>
  <frequency type="frequency" default="">
    <!--Required; video formats. NVR is not supported.-->
  </frequency>
  <bright type="uint8" min="" max="" default=""><!--Required.--></bright>
  <saturation type="uint8" min="" max="" default=""><!--Required.--></saturation>
  <contrast type="uint8" min="" max="" default=""><!--Required.--></contrast>
  <hue type="uint8" min="" max="" default=""><!--Required.--></hue>
  <sharpen><!--Optional.-->
    <switch type="boolean" default="false"><!--Required.--></switch>
    <value type="uint8" min="" max="" default="">
```

```

        <!--Dependent; When switch is true, this field is valid.-->
    </value>
</sharpen>
<denoise><!--Optional.-->
    <switch type="boolean" default="false"><!--Required.--></switch>
    <value type="uint8" min="" max="" default="">
        <!--Dependent; When switch is true, this field is valid.-->
    </value>
</denoise>
<mirrorSwitch type="boolean" default="false"><!--Optional.--></mirrorSwitch>
<flipSwitch type="boolean" default="false"><!--Optional.--></flipSwitch>
<imageRotao type="imageRotaoMode" default="0"><!--Optional.--></imageRotao>
<irisSwitch type="boolean" default="false"><!--Optional.--></irisSwitch>
<IRCutMode type="IRCutMode" default="auto">
    <!--Optional; IR-CUT mode-->
</IRCutMode>
<IRCutDayTime type="string">
    <!--Dependent; string, when IRCutMode is set to 'time', this field is valid, time format: HH:
    MM, such as 07:00. -->
</IRCutDayTime>
<IRCutNightTime type="string">
    <!--Dependent; string, when IRCutMode is set to 'time', this field is valid, time format: HH:
    MM, such as 19:00. -->
</IRCutNightTime>
<IRCutDelayTime type="uint32" min="" max="" default="">
    <!--Dependent; when IRCutMode is set to 'auto', this field is valid. The delay time in
    seconds for switching between color and black and white in an image. -->
</IRCutDelayTime>
<IRCutAlarmInTrigger><!--Dependent; when IRCutMode is set to alarmInLink, this field must
    appear.-->
    <alarmInId type="uint32"><!--Required.--></alarmInId>
    <IRCutdayNightMode type="dayNightMode"><!--Required.--></IRCutdayNightMode>
</IRCutAlarmInTrigger>
<IRCutConvSen type="IRCutConvSen" default="mid">

```

<!--Dependent; when IRCutMode is set to 'auto', this field is valid. sensitivity for switching between color and black and white in an image. -->

</IRCutConvSen>

<IRLightBright type="uint8" min="" max="" default="">

<!--Optional. brightness of infrared light.-->

</IRLightBright>

<whiteBalance><!--Optional.-->

<mode type="whitebalanceMode" default="auto"><!--Required.--></mode>

<red type="uint32" min="" max="" default=""><!--Required.--></red>

<blue type="uint32" min="" max="" default=""><!--Required.--></blue>

</whiteBalance>

<backlightCompensation><!--Optional.-->

<mode type="BLCMode" default="OFF"><!--Required.--></mode>

<HWDRLevel type="HWDRLevel" default="medium">

<!--Dependent; when mode is set to 'HWDR', this field is valid. -->

</HWDRLevel>

<BLCvalue type="uint16" default="">

<!--Dependent; when mode is set to 'BLC', this field is valid. -->

</BLCvalue>

<HLCStartTime type="string" default="">

<!--Dependent; when mode is set to 'HLC', this field is valid.HH:MM.-->

</HLCStartTime>

<HLCEndTime type="string" default="">

<!--Dependent; when mode is set to 'HLC', this field is valid. HH:MM. -->

</HLCEndTime>

</backlightCompensation>

<antiShakeDsp><!--Optional.-->

<switch type="boolean" default="">

<!--Required.-->

</switch>

</antiShakeDsp>

<smartIR><!--Optional.-->

<switch type="boolean" default="false"><!--Required.--></switch>

```
<level type="uint8" min="" max="" default="">
  <!--Required. Level of Smart IR.-->
</level>
</smartIR>
<smartIRV2><!--Optional.For dome cameras.-->
  <mode type="smartIRMode"><!--Required.smart IR mode.--></mode>
  <lightLevel type="list" maxCount="" count=""><!--Dependent.Valid when mode set to
'manual'.-->
    <item>
      <id type="uint32"><!--Required.ID of the lamp group.--></id>
      <level type="uint8" min="" max="" default="">
        <!--Required. Level of Smart IR.-->
      </level>
    </item>
  </lightLevel>
</smartIRV2>
<infraredMode type="infraredModeE" default=""><!--Optional.--></infraredMode>
<autoExposureMode><!--Optional.for dome camera-->
  <mode type="autoExposureMode" default=""><!--Required.--></mode>
  <value type="autoExposureValue " default="">
    <!--Dependent;valid when the value of 'mode' is set to 'manual'.-->
  </value>
</autoExposureMode>
<fogReduction><!--Optional.-->
  <switch type="boolean" default=""><!--Required.--></switch>
  <value type="uint8" min="" max="" default="">
    <!--Dependent; When switch is true, this field is valid.-->
  </value>
</fogReduction>
<LDC><!--Optional. Lens Distortion Correction-->
  <switch type="boolean" default=""><!--Required.--></switch>
  <value type="uint8" min="" max="" default="">
    <!--Dependent; When switch is true, this field is valid.-->
```

```

        </value>
    </LDC>
    <shutter><!--Optional.-->
        <upLimit type="shutterValue" default="">
            <!--Required. upper limit of shutter value.-->
        </upLimit>
    </shutter>
    <gain><!--Optional.-->
        <mode type="gainMode" default=""><!--Required.--></mode>
        <value type="uint8" min="" max="" default="">
            <!--Dependent; Gain value, valid when the value of 'mode' is set to 'manual'. -->
        </value>
        <AGC type="uint8" min="" max="" default="">
            <!--Dependent; Gain upper limit, valid when the value of 'mode' is set to 'auto'. -->
        </AGC>
    </gain>
    <scanSpeed type="uint8" min="" max="" default=""><!--Optional.--></scanSpeed>
    <zoomDisplay><!--Optional.for dome carama.-->
        <switch type="boolean" default=""><!--Optional.display switch.--></switch>
        <value type="uint8" min="" max="" default="">
            <!--Optional. zoom speed.-->
        </value>
    </zoomDisplay>
</image>
</config>

```

Example of Successful Response(For IPC):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <configFileType>
            <enum>normal</enum>
            <enum>day</enum>

```

```
<enum>night</enum>
</configFileType>
<frequency>
  <enum>60HZ</enum>
  <enum>50HZ</enum>
</frequency>
<imageRotaoMode>
  <enum>0</enum>
  <enum>90</enum>
  <enum>180</enum>
  <enum>270</enum>
</imageRotaoMode>
<IRCutMode>
  <enum>auto</enum>
  <enum>day</enum>
  <enum>night</enum>
  <enum>time</enum>
  <enum>alarmInLink</enum>
</IRCutMode>
<IRCutConvSen>
  <enum>high</enum>
  <enum>mid</enum>
  <enum>low</enum>
</IRCutConvSen>
<whitebalanceMode>
  <enum>auto</enum>
  <enum>indoor</enum>
  <enum>outdoor</enum>
  <enum>manual</enum>
</whitebalanceMode>
<shutterValue>
  <enum>1/3</enum>
```

```
<enum>1/6</enum>
<enum>1/12</enum>
<enum>1/20</enum>
<enum>1/25</enum>
<enum>1/50</enum>
<enum>1/75</enum>
<enum>1/100</enum>
<enum>1/150</enum>
<enum>1/200</enum>
<enum>1/250</enum>
<enum>1/300</enum>
<enum>1/500</enum>
<enum>1/750</enum>
<enum>1/1000</enum>
<enum>1/2000</enum>
<enum>1/4000</enum>
<enum>1/10000</enum>
<enum>1/100000</enum>
```

```
</shutterValue>
```

```
<shutterMode>
```

```
  <enum>auto</enum>
```

```
</shutterMode>
```

```
<BLCMode>
```

```
  <enum>OFF</enum>
```

```
  <enum>HWDR</enum>
```

```
  <enum>HLC</enum>
```

```
  <enum>BLC</enum>
```

```
</BLCMode>
```

```
<HWDRLevel>
```

```
  <enum>low</enum>
```

```
  <enum>medium</enum>
```

```
  <enum>high</enum>
```

```
</HWDRLevel>
<infraredModeE>
  <enum>on</enum>
  <enum>off</enum>
  <enum>auto</enum>
</infraredModeE>
<gainMode>
  <enum>auto</enum>
</gainMode>
<HWfogType>
  <enum>auto</enum>
  <enum>manual</enum>
  <enum>off</enum>
</HWfogType>
</types>
<cfgFile type="configFileType" default="normal">normal</cfgFile>
<image>
  <frequency type="frequency" default="50HZ">50HZ</frequency>
  <bright type="uint8" min="0" max="100" default="50">50</bright>
  <saturation type="uint8" min="0" max="100" default="50">50</saturation>
  <contrast type="uint8" min="0" max="100" default="50">50</contrast>
  <hue type="uint8" min="0" max="100" default="50">50</hue>
  <sharpen>
    <switch type="boolean" default="false">false</switch>
    <value type="uint8" min="0" max="255" default="128">128</value>
  </sharpen>
  <denoise>
    <switch type="boolean" default="false">false</switch>
    <value type="uint8" min="0" max="255" default="128">128</value>
  </denoise>
  <mirrorSwitch type="boolean" default="false">false</mirrorSwitch>
  <flipSwitch type="boolean" default="false">true</flipSwitch>
```

```
<imageRotao type="imageRotaoMode" default="0">0</imageRotao>
<irisSwitch type="boolean" default="true">true</irisSwitch>
<IRCutMode type="IRCutMode" default="auto">alarmInLink</IRCutMode>
<IRCutDayTime type="string"><![CDATA[07:00]]></IRCutDayTime>
<IRCutNightTime type="string"><![CDATA[19:00]]></IRCutNightTime>
<IRCutDelayTime type="uint32" min="2" max="6000" default="2">2</IRCutDelayTime>
<IRCutAlarmInTrigger>
  <alarmInId type="uint32">1</alarmInId>
  <IRCutdayNightMode type="dayNightMode">night</IRCutdayNightMode>
</IRCutAlarmInTrigger>
<IRCutConvSen type="IRCutConvSen" default="mid">mid</IRCutConvSen>
<IRLightBright type="uint8" min="1" max="100" default="60">60</IRLightBright>
<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>
<backlightCompensation>
  <mode type="BLCMode" default="OFF">OFF</mode>
  <HWDRLevel type="HWDRLevel" default="medium">medium</HWDRLevel>
  <BLCvalue type="uint16" default="0">0</BLCvalue>
  <HLCStartTime type="string" default="00:00"><![CDATA[00:00]]></HLCStartTime>
  <HLCEndTime type="string" default="23:59"><![CDATA[23:59]]></HLCEndTime>
</backlightCompensation>
<antiShakeDsp>
  <switch type="boolean" default="false" >false</switch>
</antiShakeDsp>
<smartIR>
  <switch type="boolean" default="false">false</switch>
  <level type="uint8" min="0" max="2" default="1">1</level>
</smartIR>
<infraredMode type="infraredModeE" default="auto">on</infraredMode>
```

```
<fogReduction>
  <switch type="boolean" default="false">false</switch>
  <value type="uint8" min="0" max="255" default="128">128</value>
</fogReduction>
<gain>
  <mode type="gainMode" default="auto">auto</mode>
  <value type="uint8" min="0" max="100" default="50">50</value>
  <AGC type="uint8" min="0" max="100" default="50">50</AGC>
</gain>
<LDC>
  <switch type="boolean" default="false">false</switch>
  <value type="uint8" min="0" max="255" default="80">80</value>
</LDC>
<shutter>
  <upLimit type="shutterValue" default="1/25">1/25</upLimit>
</shutter>
</image>
</config>
```

Example of Successful Response(For NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <configFileType>
      <enum>normal</enum>
      <enum>day</enum>
      <enum>night</enum>
    </configFileType>
    <imageRotaoMode>
      <enum>0</enum>
      <enum>90</enum>
      <enum>180</enum>
```

```
<enum>270</enum>
</imageRotaoMode>
<IRCutMode>
  <enum>auto</enum>
  <enum>day</enum>
  <enum>night</enum>
  <enum>time</enum>
  <enum>alarmInLink</enum>
</IRCutMode>
<IRCutConvSen>
  <enum>high</enum>
  <enum>mid</enum>
  <enum>low</enum>
</IRCutConvSen>
<whitebalanceMode>
  <enum>auto</enum>
  <enum>indoor</enum>
  <enum>outdoor</enum>
  <enum>manual</enum>
</whitebalanceMode>
<BLCMode>
  <enum>OFF</enum>
  <enum>HWDR</enum>
  <enum>HLC</enum>
  <enum>BLC</enum>
</BLCMode>
<HWDRLevel>
  <enum>low</enum>
  <enum>medium</enum>
  <enum>high</enum>
</HWDRLevel>
<infraredModeE>
```

```
<enum>on</enum>
<enum>off</enum>
<enum>auto</enum>
</infraredModeE>
<gainMode>
  <enum>auto</enum>
  <enum>manual</enum>
</gainMode>
<smartIRMode>
  <enum>off</enum>
  <enum>manual</enum>
  <enum>auto</enum>
</smartIRMode>
<autoExposureMode>
  <enum>auto</enum>
  <enum>manual</enum>
</autoExposureMode>
<autoExposureValue>
  <enum>1/25</enum>
  <enum>1/50</enum>
  <enum>1/75</enum>
  <enum>1/100</enum>
  <enum>1/120</enum>
  <enum>1/150</enum>
  <enum>1/175</enum>
  <enum>1/200</enum>
  <enum>1/225</enum>
  <enum>1/250</enum>
  <enum>1/300</enum>
  <enum>1/425</enum>
  <enum>1/600</enum>
  <enum>1/1000</enum>
```

```
<enum>1/1250</enum>
<enum>1/1750</enum>
<enum>1/2500</enum>
<enum>1/3500</enum>
<enum>1/6000</enum>
<enum>1/10000</enum>
<enum>1/25000</enum>
```

```
</autoExposureValue>
```

```
</types>
```

```
<cfgFile type="configFileType" default="normal">normal</cfgFile>
```

```
<image>
```

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

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

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

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

```
<sharpen>
```

```
<switch type="boolean" default="false">true</switch>
```

```
<value type="uint8" min="0" max="100" default="50">180</value>
```

```
</sharpen>
```

```
<denoise>
```

```
<switch type="boolean" default="false">true</switch>
```

```
<value type="uint8" min="0" max="100" default="50">48</value>
```

```
</denoise>
```

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

```
<flipSwitch type="boolean" default="false">true</flipSwitch>
```

```
<IRCutMode type="IRCutMode" default="auto">time</IRCutMode>
```

```
<IRCutDayTime type="string">
```

```
<![CDATA[08:55]]>
```

```
</IRCutDayTime>
```

```
<IRCutNightTime type="string">
```

```
<![CDATA[19:00]]>
```

```
</IRCutNightTime>
```

```
<whiteBalance>
  <mode type="whitebalanceMode" default="auto">manual</mode>
  <red min="0" max="100" default="50">14</red>
  <blue min="0" max="100" default="50">5</blue>
</whiteBalance>
<backlightCompensation>
  <mode type="BLCMode" default="OFF">HLC</mode>
</backlightCompensation>
<smartIRV2>
  <mode type="smartIRMode">manual</mode>
  <lightLevel type="list" maxCount="1" count="1">
    <item>
      <id type="uint32">1</id>
      <level type="uint8" min="0" max="100" default="50">11</level>
    </item>
  </lightLevel>
</smartIRV2>
<infraredMode type="infraredModeE" default="auto">auto</infraredMode>
<autoExposureMode>
  <mode type="autoExposureMode" default="auto">manual</mode>
  <value type="autoExposureValue" default="1/30">1/25</value>
</autoExposureMode>
<fogReduction>
  <switch type="boolean" default="false">true</switch>
  <value type="uint8" min="0" max="100" default="50">50</value>
</fogReduction>
<gain>
  <mode type="gainMode" default="">manual</mode>
  <value type="uint8" min="0" max="100" default="50">29</value>
</gain>
</image>
</config>
```

[Tips]:

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.
Action name	None
Entity Data	<config>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<config> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <cfgFile> <!--Optional; IPC supports 3 sets of image configuration files.--> </cfgFile> <image> <frequency><!--Optional; video formats.--></frequency> <bright><!--Optional.--></bright> <saturation><!--Optional.--></saturation> <contrast><!--Optional.--></contrast> <hue><!--Optional.--></hue> <sharpen><!--Optional.--> <switch><!--Required.--></switch> <value><!--Dependent; When switch is true, this field is valid.--></value> </sharpen> <denoise><!--Optional.--> <switch><!--Required.--></switch></pre>	

```
<value><!--Dependent; When switch is true, this field is valid.--></value>

</denoise>

<mirrorSwitch><!--Optional.--></mirrorSwitch>

<flipSwitch><!--Optional.--></flipSwitch>

<imageRotao><!--Optional.--></imageRotao>

<irisSwitch><!--Optional.--></irisSwitch>

<IRCutMode><!--Optional; IR-CUT mode--></IRCutMode>

<IRCutDayTime>

    <!--Dependent; string, when IRCutMode is set to 'time', this field is valid, time format: HH:
    MM, such as 07:00. -->

</IRCutDayTime>

<IRCutNightTime>

    <!--Dependent; string, when IRCutMode is set to 'time', this field is valid, time format: HH:
    MM, such as 19:00. -->

</IRCutNightTime>

<IRCutDelayTime>

    <!--Dependent; when IRCutMode is set to 'auto', this field is valid. The delay time in
    seconds for switching between color and black and white in an image. -->

</IRCutDelayTime>

<IRCutAlarmInTrigger><!--Dependent; when IRCutMode is set to alarmInLink, this field must
appear.-->

    <alarmInId><!--Required.--></alarmInId>

    <IRCutdayNightMode><!--Required.--></IRCutdayNightMode>

</IRCutAlarmInTrigger>

<IRCutConvSen>

    <!--Dependent; when IRCutMode is set to 'auto', this field is valid. sensitivity for switching
    between color and black and white in an image. -->

</IRCutConvSen>

<IRLightBright><!--Optional. brightness of infrared light.--></IRLightBright>

<whiteBalance><!--Optional.-->

    <mode><!--Required.--></mode>

    <red><!--Required.--></red>

    <blue><!--Required.--></blue>

</whiteBalance>
```

```
<backlightCompensation><!--Optional.-->
  <mode><!--Required.--></mode>
  <HWDRLevel>
    <!--Dependent; when mode is set to 'HWDR', this field is valid. -->
  </HWDRLevel>
  <BLCvalue><!--Dependent; when mode is set to 'BLC', this field is valid. --></BLCvalue>
  <HLCStartTime>
    <!--Dependent; when mode is set to 'HLC', this field is valid. -->
  </HLCStartTime>
  <HLCEndTime>
    <!--Dependent; when mode is set to 'HLC', this field is valid. -->
  </HLCEndTime>
</backlightCompensation>
<antiShakeDsp><!--Optional.-->
  <switch><!--Required.--></switch>
</antiShakeDsp>
<smartIR><!--Optional.-->
  <switch><!--Required.--></switch>
  <level><!--Required.--></level>
</smartIR>
<smartIRV2><!--Optional.For dome cameras.-->
  <mode><!--Required.smart IR mode.--></mode>
  <lightLevel><!--Dependent.Valid when mode set to 'manual'.-->
    <item>
      <id><!--Required.ID of the lamp group.--></id>
      <level><!--Required. Level of Smart IR.--></level>
    </item>
  </lightLevel>
</smartIRV2>
<infraredMode><!--Optional.--></infraredMode>
<autoExposureMode><!--Optional.for dome camera-->
  <mode><!--Required.--></mode>
```

```
<value><!--Dependent;valid when the value of 'mode' is set to 'manual'.--></value>
</autoExposureMode>
<fogReduction><!--Optional.-->
  <switch><!--Required.--></switch>
  <value><!--Dependent; When switch is true, this field is valid.--></value>
</fogReduction>
<LDC><!--Optional. Lens Distortion Correction-->
  <switch><!--Required.--></switch>
  <value><!--Dependent; When switch is true, this field is valid.--></value>
</LDC>
<shutter><!--Optional.-->
  <upLimit><!--Required. upper limit of shutter value.--></upLimit>
</shutter>
<gain><!--Optional.-->
  <mode><!--Required.--></mode>
  <value>
    <!--Dependent; Gain value, valid when the value of 'mode' is set to 'manual'. -->
  </value>
  <AGC>
    <!--Dependent; Gain upper limit, valid when the value of 'mode' is set to 'auto'. -->
  </AGC>
</gain>
<scanSpeed><!--Optional.--></scanSpeed>
<zoomDisplay><!--Optional.for dome carama.-->
  <switch><!--Optional.display switch.--></switch>
  <value><!--Optional. zoom speed.--></value>
</zoomDisplay>
</image>
<rebootPrompt>
  <!--Optional. after modifying some parameters, the device needs to be restarted. If set to true,
  the system will fail with a prompt.detail see "Tips".-->
</rebootPrompt>
</config>
```

Example of Entity Data(Set bright to 60):

```
<xml version="1.0" encoding="utf-8" />
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <image>
    <bright>60</bright>
  </image>
  <cfgFile>normal</cfgFile>
  <rebootPrompt>true</rebootPrompt>
</config>
```

Example of Entity Data(Set contrast to 70):

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <image>
    <contrast>70</contrast>
  </image>
  <cfgFile>normal</cfgFile>
  <rebootPrompt>true</rebootPrompt>
</config>
```

Example of Entity Data(Set bright to 60 and set contrast to 70):

```
<xml version="1.0" encoding="utf-8" />
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <image>
    <bright>60</bright>
    <contrast>70</contrast>
  </image>
  <cfgFile>normal</cfgFile>
  <rebootPrompt>true</rebootPrompt>
</config>
```

[Tips]:

1. rebootPrompt: Modifying some parameters will cause the device to restart. If "rebootPrompt" is set to true, the system will first return a failure to remind the client. If the client confirms to continue modifying, it needs to set "rebootPrompt" to false and retry.
2. This API supports setting only partial parameters, and the parameters that are not carried remain unchanged.

3.2.3 GetImageConfigFileInfo

GetImageConfigFileInfo	
Description	To get the IP media device's image configuration file information for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetImageConfigFileInfo[/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	<imageConfigFileInfo>
Applicable products	IPC,NVR

<imageConfigFileInfo> XML Block:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <progType> <!--Required;define selection strategy for image configuration files, details refer to
    "Tips".-->
      <enum>normal</enum>
      <enum>time</enum>
      <enum>auto</enum>
    </progType>
  </types>
  <imageConfigFileInfo>
    <program type="progType"><!--Required; details refer to "Tips".--></program>
    <dayTime type="string">
      <!--Dependent; string,valid when 'program' is set to 'time', time format: HH: MM, such as
      07:00.-->
    </dayTime>
    <nightTime type="string">
      <!--Dependent; string, valid when 'program' is set to 'time', time format: HH: MM, such as
      18:00.-->
    </nightTime>
  </imageConfigFileInfo>
</config>
```

Example of Successful Response(For normal config file):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <progType>
      <enum>normal</enum>
      <enum>time</enum>
      <enum>auto</enum>
    </progType>
  </types>
  <imageConfigFileInfo>
    <program type="progType">normal</program>
  </imageConfigFileInfo>
</config>
```

Example of Successful Response(For time config file):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <progType>
      <enum>normal</enum>
      <enum>time</enum>
      <enum>auto</enum>
    </progType>
  </types>
  <imageConfigFileInfo>
    <program type="progType">time</program>
    <dayTime type="string"> <![CDATA[07:00]]></dayTime>
    <nightTime type="string"> <![CDATA[18:00]]> </nightTime>
  </imageConfigFileInfo>
</config>
```

[Tips]:

1. Regarding the explanation of "progType":

- normal: always use the 'normal' image configuration file.
- time: select the image configuration file based on the time configuration (select "day" configuration file or "night" configuration file).
- auto: automatically select image configuration file based on environmental brightness.

3.2.4 SetImageConfigFileInfo

SetImageConfigFileInfo	
Description	To set the IP media device's image configuration file information for specific channel.
Typical URL	POST http://<host>[:port]/SetImageConfigFileInfo[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<imageConfigFileInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><imageConfigFileInfo> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <imageConfigFileInfo> <program><!--Required; details refer to "Tips".--></program> <dayTime> <!--Dependent; string,valid when 'program' is set to 'time', time format: HH: MM, such as 07:00.--> </dayTime> <nightTime> <!--Dependent; string, valid when 'program' is set to 'time', time format: HH: MM, such as 18:00.--> </nightTime> </imageConfigFileInfo> </config> </pre>	

Example of Entity Data (For normal config file):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <imageConfigFileInfo>
    <program type="progType">normal</program>
  </imageConfigFileInfo>
</config>
```

Example of Successful Response(For time config file):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <imageConfigFileInfo>
    <program type="progType">time</program>
    <dayTime><![CDATA[07:00]]> </dayTime>
    <nightTime><![CDATA[18:00]]> </nightTime>
  </imageConfigFileInfo>
</config>
```

[Tips]:

1. Regarding the explanation of "progType":

- normal: always use the 'normal' image configuration file.
- time: select the image configuration file based on the time configuration(select "day" configuration file or "night" configuration file).
- auto: automatically select image configuration file based on environmental brightness.

3.2.5 GetSnapshot

GetSnapshot	
Description	To get a picture encoded by jpg for specific channel.
Typical URL	POST or GET http://<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.
Applicable products	IPC,NVR

3.2.6 GetSnapshotByTime

GetSnapshotByTime	
Description	To get a key frame for specific channel on specific time.
Typical URL	POST or GET http://<host>[:port]/GetSnapshotByTime[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<search>
Successful Response	<downloadOneImage>
Applicable products	IPC,NVR

<search> XML Block:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="" xmlns="http://www.ipc.com/ver10">
  <search> <!--Required; the time range for searching.-->
    <time> <!--Required; string, start time, format: YYYY-MM-DD HH:MM:SS, 24-hour.--></time>
    <length><!--Required; uint16, time span,unit: seconds, end time="time"+"length".--></length>
  </search>
</config>
```

<downloadOneImage> XML Block:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <downloadOneImage><!--Required.-->
    <fileName type="string"><!--Required.file name.--></fileName>
    <sourceBase64Length type="uint32">
      <!--Required.file length, the length of BASE64 encoding.-->
    </sourceBase64Length>
    <sourceBase64Data type="string">
      <!--Required. base64 encryption of source data.-->
    </sourceBase64Data>
  </downloadOneImage>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <search>
    <time><![CDATA[2024-08-23 15:07:28]]></time>
    <length>10</length>
  </search>
</config>
```

[Tips]:

- 1.It returns the data from "time" in "length" seconds.
- 2.The response a picture encoded by jpg.
3. If there are multiple images, return the first one. If there are no images, return 'sourceBase64Length' as 0.

3.3 Stream

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	<config>
Applicable products	IPC,NVR
<config> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <audioEncode><!--Required.--> <enum>G711A</enum> <enum>G711U</enum> </audioEncode> <audioInput><!--Required.--> <enum>MIC</enum> <enum>LIN</enum> </audioInput> </config>	

```

<audioOutput><!--Required.-->
  <enum>TALKBACK</enum>
  <enum>ALARM_AUDIO</enum>
  <enum>AUTO</enum>
</audioOutput>
</types>
<audioInSwitch type="boolean"><!--Optional.audio enable or not--></audioInSwitch>
<audioEncode type="audioEncode"><!-- Optional.audio encode type.--></audioEncode>
<audioInput type="audioInput"><!-- Optional.audio input type, MIC or Line.--></audioInput>
<volume><!--Optional.-->
  <linInVolume type="uint32" min="" max=""><!--Optional.--></linInVolume>
  <micInVolume type="uint32" min="" max=""><!--Optional.--></micInVolume>
  <audioOutVolume type="uint32" min="" max=""><!--Optional.--></audioOutVolume>
</volume>
<audioOutput type="audioOutput">
  <!--Optional. selection mode for alarm and talkback.-->
</audioOutput>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <audioEncodeE>
      <enum>G711A</enum>
      <enum>G711U</enum>
    </audioEncodeE>
    <audioInputE>
      <enum>MIC</enum>
      <enum>LIN</enum>
    </audioInputE>
    <audioOutputE>
      <enum>TALKBACK</enum>

```

```

        <enum>ALARM_AUDIO</enum>
        <enum>AUTO</enum>
    </audioOutputE>
</types>
<audioInSwitch type="boolean">true</audioInSwitch>
<audioEncode type="audioEncodeE">G711U</audioEncode>
<audioInput type="audioInputE">MIC</audioInput>
<volume>
    <linInVolume type="uint32" min="0" max="100">75</linInVolume>
    <micInVolume type="uint32" min="0" max="100">75</micInVolume>
    <audioOutVolume type="uint32" min="0" max="100">75</audioOutVolume>
</volume>
<audioOutput type="audioOutputE">ALARM_AUDIO</audioOutput>
</config>

```

[Tips]:

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.
Action name	None
Entity Data	<config>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<config> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <audioInSwitch><!--Optional.boolean, audio enable or not.--></audioInSwitch>
  <audioEncode><!-- Optional.audioEncodeE, audio encode type.--></audioEncode>
  <audioInput><!-- Optional.audioInputE, audio input type, MIC or Line.--></audioInput>
  <volume><!--Optional.-->
    <linInVolume><!--Optional.uint32.--></linInVolume>
    <micInVolume><!--Optional.uint32.--></micInVolume>
    <audioOutVolume><!--Optional.uint32.--></audioOutVolume>
  </volume>
  <audioOutput><!--Optional. audioOutputE,selection mode for alarm and talkback.--></audioOutput>
</config>
```

Example of Entity Data(Turn on audio):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <audioInSwitch>true</audioInSwitch>
  <audioEncode>G711A</audioEncode>
  <audioInput>MIC</audioInput>
  <volume>
    <micInVolume>75</micInVolume>
    <audioOutVolume>75</audioOutVolume>
  </volume>
  <audioOutput>ALARM_AUDIO</audioOutput>
</config>
```

Example of Entity Data(Turn off audio):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <audioInSwitch>>false</audioInSwitch>
</config>
```

[Tips]:

3.3.3 GetVideoStreamConfig

GetVideoStreamConfig	
Description	To get the IP media device's video stream configuration for specific channel.
Typical URL	POST or GET http://<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	<config>
Applicable products	IPC,NVR
<p><config> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <bitRateTypeE><!--Required;define bit rate type.--> <enum>VBR</enum> <enum>CBR</enum> </bitRateTypeE> <quality><!--Required;define video quality level.--> <enum>lowest</enum> <enum>lower</enum> <enum>medium</enum> <enum>higher</enum> <enum>highest</enum> </quality> <encodeTypeE><!--Required;define video encode type.--> <enum>h264</enum> <enum>h265</enum> <enum>h264plus</enum> <enum>h265plus</enum> </pre>	

```

        <enum>h264smart</enum>
        <enum>h265smart</enum>
        <enum>mjpeg</enum>
    </encodeTypeE>
    <encodeLevelE><!--Required;define h.264 profile.-->
        <enum>baseLine</enum>
        <enum>mainProfile</enum>
        <enum>highProfile</enum>
    </encodeLevelE>
</types>
<streams type="list" count=""><!--Required; The video parameters of each stream usually include
the main stream, sub stream, and third stream.-->
    <item id=""><!--Required; id represents the sequence number of the stream, 1 represents the
main stream, 2 represents the sub stream, etc.-->
        <name type="string" maxLen="32">
            <!--Optional;string,the name of stream.-->
        </name>
        <resolution type="string"><!--Required; video width*height.--></resolution>
        <encodeType type="encodeTypeE">
            <!--Required; the encoding type currently in use.-->
        </encodeType>
        <encodeLevel type="encodeLevelE">
            <!--Required; the profile currently in use.-->
        </encodeLevel>
        <frameRate type="uint32" min="" max=""><!--Required.--></frameRate>
        <bitRateType type="bitRateType"><!--Required--></bitRateType>
        <maxBitRate type="uint32" min="" max="">
            <!--Required; when the value of "bitRateType" is VBR, it represents the upper limit of
the bit rate; when the value of 'bitRateType' is CBR, it represents the bit rate.-->
        </maxBitRate>
        <quality type="quality"><!--Required; video quality.--></quality>
        <GOP type="uint32" min="" max="" default="">
            <!--Required; I-frame interval.-->

```

```
</GOP>
</item>
</streams>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <bitRateTypeE>
      <enum>VBR</enum>
      <enum>CBR</enum>
    </bitRateTypeE>
    <quality>
      <enum>lowest</enum>
      <enum>lower</enum>
      <enum>medium</enum>
      <enum>higher</enum>
      <enum>highest</enum>
    </quality>
    <encodeTypeE>
      <enum>h264</enum>
      <enum>h265</enum>
      <enum>h264plus</enum>
      <enum>h265plus</enum>
      <enum>h264smart</enum>
      <enum>h265smart</enum>
      <enum>mjpeg</enum>
    </encodeTypeE>
    <encodeLevelE>
      <enum>baseLine</enum>
      <enum>mainProfile</enum>
      <enum>highProfile</enum>
```

```
</encodeLevelE>
</types>
<streams type="list" count="3">
  <item id="1">
    <name type="string" maxLen="32">
      <![CDATA[profile1]]>
    </name>
    <resolution type="string"><![CDATA[2688x1520]]></resolution>
    <encodeType type="encodeTypeE">h265</encodeType>
    <encodeLevel type="encodeLevelE">highProfile</encodeLevel>
    <frameRate type="uint32" min="1" max="25">25</frameRate>
    <bitRateType type="bitRateTypeE">VBR</bitRateType>
    <maxBitRate type="uint32" min="32" max="8192">5120</maxBitRate>
    <quality type="quality">higher</quality>
    <GOP type="uint32" min="1" max="1500" default="100">50</GOP>
  </item>
  <item id="2">
    <name type="string" maxLen="32">
      <![CDATA[profile2]]>
    </name>
    <resolution type="string"><![CDATA[352x288]]></resolution>
    <encodeType type="encodeTypeE">h265</encodeType>
    <encodeLevel type="encodeLevelE">highProfile</encodeLevel>
    <frameRate type="uint32" min="1" max="25">25</frameRate>
    <bitRateType type="bitRateTypeE">VBR</bitRateType>
    <maxBitRate type="uint32" min="32" max="6144">512</maxBitRate>
    <quality type="quality">higher</quality>
    <GOP type="uint32" min="1" max="1500" default="100">50</GOP>
  </item>
  <item id="3">
    <name type="string" maxLen="32">
      <![CDATA[profile3]]>
    </name>
```

```

</name>
<resolution><![CDATA[704x576]]></resolution>
<encodeType type="encodeTypeE">h265</encodeType>
<encodeLevel type="encodeLevelE">highProfile</encodeLevel>
<frameRate type="uint32" min="1" max="25">25</frameRate>
<bitRateType type="bitRateTypeE">CBR</bitRateType>
<maxBitRate type="uint32" min="32" max="4096">512</maxBitRate>
<quality type="quality">medium</quality>
<GOP type="uint32" min="1" max="1500" default="100">50</GOP>
</item>
</streams>
</config>

```

[Tips]:

- 1.The "count=3" means the channel supports 3 streams at the same time. Each stream's current video configuration is announced in the "item" sub element.
- 2.The "id" attribute for each item starts from "1".1:main stream,2:sub stream,3:third stream, and so on.

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	<config>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<config> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <streams><!--Required; The video parameters of each stream usually include the main stream, sub
  stream, and third stream.-->
    <item id=""><!--Required; id represents the sequence number of the stream, 1: main stream,
    2 :sub stream,3:third stream. etc.-->
      <resolution><!--Optional; string,video width*height.--></resolution>
      <encodeType><!--Optional; encodeTypeE,set encoding type.--></encodeType>
      <encodeLevel><!--Optional; encodeLevel,set the profile.--></encodeLevel>
      <frameRate><!--Optional;--></frameRate>
      <bitRateType><!--Optional; bitRateTypeE.--></bitRateType>
      <maxBitRate>
        <!--Optional; when the value of "bitRateType" is VBR, it represents the upper limit of
        the bit rate; when the value of 'bitRateType' is CBR, it represents the bit rate.-->
      </maxBitRate>
      <quality><!--Optional; video quality.--></quality>
      <GOP><!--Optional; I-frame interval.--></GOP>
    </item>
  </streams>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <streams>
    <item id="1">
      <resolution><![CDATA[2688x1520]]></resolution>
      <encodeType>h265</encodeType>
      <encodeLevel>highProfile</encodeLevel>
      <frameRate>25</frameRate>
      <bitRateType>VBR</bitRateType>
      <maxBitRate>5120</maxBitRate>
      <quality>higher</quality>
      <GOP>50</GOP>
    </item>
  </streams>
</config>
```

```

</item>
<item id="2">
  <resolution><![CDATA[352x288]]></resolution>
  <encodeType>h265</encodeType>
  <encodeLevel>highProfile</encodeLevel>
  <frameRate>25</frameRate>
  <bitRateType>VBR</bitRateType>
  <maxBitRate>512</maxBitRate>
  <quality>higher</quality>
  <GOP>50</GOP>
</item>
<item id="3">
  <resolution><![CDATA[704x576]]></resolution>
  <encodeType>h265</encodeType>
  <encodeLevel>highProfile</encodeLevel>
  <frameRate>25</frameRate>
  <bitRateType>CBR</bitRateType>
  <maxBitRate>512</maxBitRate>
  <quality>medium</quality>
  <GOP>50</GOP>
</item>
</streams>
</config>

```

[Tips]:

1.The "id" attribute for each item starts from "1".1:main stream,2:sub stream,3:third stream, and so on.

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 GET http://<host>[:port]/RequestKeyFrame [/channelId]

RequestKeyFrame	
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.
Applicable products	IPC

3.4 OSD

3.4.1 GetImageOsdConfig

GetImageOsdConfig	
Description	To get the IP media device's image OSD(on-screen display) element 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	<imageOsd>
Applicable products	IPC,NVR
<p><imageOsd> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <types> <timeFormatModeType><!--Required.--> <enum>12h</enum> <enum>24h</enum> </timeFormatModeType></pre>	

```
<dateFormatE><!--Required.-->
```

```
  <enum>year-month-day</enum>
```

```
  <enum>month-day-year</enum>
```

```
  <enum>day-month-year</enum>
```

```
  <enum>year/month/day</enum>
```

```
  <enum>month/day/year</enum>
```

```
  <enum>day/month/year</enum>
```

```
</dateFormatE>
```

```
<osdFontSize><!--Required;OSD font size.-->
```

```
  <enum>16x16</enum>
```

```
  <enum>32x32</enum>
```

```
  <enum>48x48</enum>
```

```
  <enum>64x64</enum>
```

```
  <enum>auto</enum>
```

```
</osdFontSize>
```

```
<fontColorModeE><!--Optional;osd color mode.-->
```

```
  <enum>outline</enum><!--Required; System default color.-->
```

```
  <enum>customize</enum><!--Required; Custom color.-->
```

```
</fontColorModeE>
```

```
<osdOverlayTypeE><!--Required;text OSD or image OSD.-->
```

```
  <enum>TEXT</enum>
```

```
  <enum>IMAGE</enum><!--Optional; NVR is not supported. -->
```

```
  <enum>TEMPERINFO</enum><!-- Optional; osd of temperature bar. -->
```

```
</osdOverlayTypeE>
```

```
<supportCalendarTypeE><!--Optional. valid when supporting Persian calendar.-->
```

```
  <enum>Gregorian</enum>
```

```
  <enum>Persian</enum>
```

```
</supportCalendarTypeE>
```

```
</types>
```

```
<imageOsd><!--Required.-->
```

```
  <time><!--Required; OSD display "date and time".-->
```

```
    <switch type="boolean"><!--Required.--></switch>
```

```
<X type="uint32" min="0" max="10000">
    <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when switch is
    'true'.-->
</X>
<Y type="uint32" min="0" max="10000">
    <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when switch is
    'true'. -->
</Y>
<timeFormatMode type="timeFormatModeType">
    <!--Dependent; Valid when switch is 'true'.-->
</timeFormatMode>
<dateFormat type="dateFormatE">
    <!--Dependent; Valid when switch is 'true'.-->
</dateFormat>
<supportCalendarType type="supportCalendarTypeE">
    <!--Optional. valid when supporting Persian calendar.-->
</supportCalendarType >
</time>
<channelName><!--Required; OSD display 'channel name'.-->
    <switch type="boolean"><!--Required.--></switch>
    <X type="uint32" min="" max="">
        <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when 'switch' is
        True.-->
    </X>
    <Y type="uint32" min="" max="">
        <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when switch is
        'true'. -->
    </Y>
    <name type="string" maxCharNum="">
        <!--Dependent; channel name.valid when switch is 'true'.-->
    </name>
</channelName>
<textOverLay type="list" maxCount="" count=""><!--Optional; customize OSD information.
    The current NVR is not supported.-->
```

```
<item>
  <switch type="boolean"><!--Required.--></switch>
  <X type="uint32" min="" max="">
    <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when
    switch is 'true'.-->
  </X>
  <Y type="uint32" min="" max="">
    <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when
    switch is 'true'. -->
  </Y>
  <osdOverlayType type="osdOverlayTypeE">
    <!--Required. type of OSD.-->
  </osdOverlayType>
  <value type="list" maxCount="2" count="">
    <!--Dependent; customize the content of OSD. valid when the value of
    osdOverlayType is 'TEXT'.-->
    <itemType type="string" maxCharNum=""/>
    <item>
      <![CDATA[2222222222]]>
    </item>
  </value>
</item>
</textOverLay>
<imgDataLen type="uint32">
  <!--Dependent; The value of 'osdOverlayType' is valid when set to 'IMAGE', indicating the
  length of the image osd. Only valid for first item in textOverLay.NVR is not supported.-->
</imgDataLen>
<imgBase64Data type="string" maxLen="204800">
  <!--Dependent; The value of 'osdOverlayType' is valid when set to 'IMAGE', indicating the
  content of the image osd. Only valid for first item in textOverLay. NVR is not
  supported.-->
</imgBase64Data>
<devStatus><!--Optional. device status, only for dome camera.-->
  <X type="uint32" min="" max="">
    <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when
```

```

        switch is 'true'.-->
    </X>
    <Y type="uint32" min="" max="">
        <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when
        switch is 'true'. -->
    </Y>
    <runStatus>
        <switch type="boolean"><!--Required; display run status or not.--></switch>
    </runStatus>
    <domDirect>
        <switch type="boolean"><!--Required; display PTZ direction or not.--></switch>
    </domDirect>
</devStatus>
<osdAttr><!--Optional. OSD font properties. NVR is not supported.-->
    <transparent type="uint32" min="" max="" default="">
        <!--Required. Font Transparency. -->
    </transparent>
    <fontSize type="osdFontSize"><!--Required;font size.--></fontSize>
    <fontColorMode type="fontColorModeE">
        <!--Required;font color mode.-->
    </fontColorMode>
    <fontColor type="string">
        <!--Dependent;font color, Valid when fontColorMode set to 'customize'.-->
    </fontColor >
</osdAttr>
</imageOsd>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <timeFormatModeType>

```

```
<enum>12h</enum>
<enum>24h</enum>
</timeFormatModeType>
<dateFormatE>
  <enum>year-month-day</enum>
  <enum>month-day-year</enum>
  <enum>day-month-year</enum>
  <enum>year/month/day</enum>
  <enum>month/day/year</enum>
  <enum>day/month/year</enum>
</dateFormatE>
<osdFontSize>
  <enum>16x16</enum>
  <enum>32x32</enum>
  <enum>48x48</enum>
  <enum>64x64</enum>
  <enum>auto</enum>
</osdFontSize>
<osdOverlayTypeE>
  <enum>TEXT</enum>
  <enum>IMAGE</enum>
</osdOverlayTypeE>
</types>
<imageOsd>
  <time>
    <switch type="boolean">true</switch>
    <X type="uint32" min="0" max="10000">363</X>
    <Y type="uint32" min="0" max="10000">3000</Y>
    <timeFormatMode type="timeFormatModeType">24h</timeFormatMode>
    <dateFormat type="dateFormatE">year-month-day</dateFormat>
  </time>
  <channelName>
```

```
<switch type="boolean">true</switch>
<X type="uint32" min="0" max="10000">1000</X>
<Y type="uint32" min="0" max="10000">3000</Y>
<name type="string" maxCharNum="18">
  <![CDATA[IPC]]>
</name>
</channelName>
<textOverLay type="list" maxCount="4" count="4">
  <item>
    <switch type="boolean">false</switch>
    <X type="uint32" min="0" max="10000">0</X>
    <Y type="uint32" min="0" max="10000">181</Y>
    <osdOverlayType type="osdOverlayTypeE">TEXT</osdOverlayType>
    <value type="list" maxCount="2" count="1">
      <itemType type="string" maxCharNum="15"/>
      <item>
        <![CDATA[IPC]]>
      </item>
    </value>
  </item>
  <item>
    <switch type="boolean">false</switch>
    <X type="uint32">613</X>
    <Y type="uint32">2151</Y>
    <osdOverlayType type="osdOverlayTypeE">TEXT</osdOverlayType>
    <value type="list" maxCount="2" count="2">
      <itemType type="string" maxLen="15"/>
      <item>
        <![CDATA[222222222]]>
      </item>
      <item>
        <![CDATA[333333333]]>
      </item>
    </value>
  </item>
</textOverLay>
</channelName>
```

```
</item>
  </value>
</item>
<item>
  <switch type="boolean">false</switch>
  <X type="uint32">727</X>
  <Y type="uint32">2151</Y>
  <value type="list" maxCount="2" count="1">
    <itemType type="string" maxLen="15"/>
    <item>
      <![CDATA[333333333]]>
    </item>
  </value>
  <osdOverlayType type="osdOverlayTypeE">TEXT</osdOverlayType>
</item>
<item>
  <switch type="boolean">false</switch>
  <X type="uint32">613</X>
  <Y type="uint32">2393</Y>
  <value type="list" maxCount="2" count="1">
    <itemType type="string" maxLen="15"/>
    <item>
      <![CDATA[444444444]]>
    </item>
  </value>
  <osdOverlayType type="osdOverlayTypeE">IMAGE</osdOverlayType>
</item>
</textOverLay>
<imgDataLen type="uint32">1024</imgDataLen>
<imgBase64Data type="string" maxLen="204800">
  <![CDATA[/9j/4AAQSkZJRgABA..... //2Q==]]>
</imgBase64Data>
```

```

    <osdAttr>
      <transparent type="uint32" min="0" max="15" default="0">0</transparent>
      <fontSize type="osdFontSize">auto</fontSize>
      <fontColorMode type="fontColorModeE">customize</fontColorMode>
      <fontColor type="string">0099bb</fontColor>
    </osdAttr>
  </imageOsd>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution.
2. The content of imgBase64Data is too long and has been omitted from the examples in the document.

3.4.2 SetImageOsdConfig

SetImageOsdConfig	
Description	To set the IP media device's image OSD element 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	<imageOsd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<imageOsd> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <imageOsd><!--Required.-->
    <time><!--Optional; OSD display "date and time".-->
      <switch><!--Required.--></switch>
      <X>
        <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when switch is

```

```

        'true'.-->
    </X>
    <Y>
        <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when switch is
        'true'. -->
    </Y>
    <timeFormatMode><!--Dependent; Valid when switch is 'true'.--></timeFormatMode>
    <dateFormat><!--Dependent; Valid when switch is 'true'.--></dateFormat>
    <supportCalendarType>
        <!--Optional. valid when supporting Persian calendar.-->
    </supportCalendarType >
</time>
<channelName><!--Optional; OSD display 'channel name'.-->
    <switch><!--Required.--></switch>
    <X>
        <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when switch is
        'true'.-->
    </X>
    <Y>
        <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when switch is
        'true'. -->
    </Y>
    <name><!--Dependent; channel name.valid when switch is 'true'.--> </name>
</channelName>
<textOverLay><!--Optional; customize OSD information.-->
    <item><!--Optional.-->
        <switch><!--Required.--></switch>
        <X>
            <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when
            switch is 'true'.-->
        </X>
        <Y>
            <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when
            switch is 'true'. -->

```

```
</Y>
<osdOverlayType><!--Required. type of OSD.--></osdOverlayType>
<value>
  <!--Dependent; customize the content of OSD. valid when the value of
  osdOverlayType is 'TEXT'.-->
  <item>
    <![CDATA[2222222222]]>
  </item>
</value>
</item>
</textOverLay>
<imgDataLen>
  <!--Dependent; unit32, The length of ImgBase64Data.-->
</imgDataLen>
<imgBase64Data>
  <!--Dependent; type of OSD. The value of 'osdOverlayType' is valid when set to 'IMAGE',
  indicating the content of the image osd. base64 encryption value of image content.-->
</imgBase64Data>
<devStatus><!--Optional. device status, only for dome camera.-->
  <X>
    <!--Dependent; The X-coordinate in the upper left corner of OSD, Valid when
    switch is 'true'.-->
  </X>
  <Y>
    <!--Dependent; The Y-coordinate in the upper left corner of OSD, Valid when
    switch is 'true'. -->
  </Y>
  <runStatus>
    <switch><!--Required; run status display switch.--></switch>
  </runStatus>
  <domDirect>
    <switch><!--Required; PTZ direction display switch.--></switch>
  </domDirect>
</devStatus>
```

```

<osdAttr><!--Optional. OSD font properties. NVR is not supported.-->
  <transparent><!--Optional. Font transparency.--></transparent>
  <fontSize><!--Optional;font size.--></fontSize>
  <fontColorMode>
    <!--Required;fontColorModeE;font color mode.-->
  </fontColorMode>
  <fontColor><!--Dependent;string;font color, Valid when fontColorMode set to
'customize'.-->
  </fontColor >
</osdAttr>
</imageOsd>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <imageOsd>
    <time>
      <switch>true</switch>
      <X>363</X>
      <Y>3000</Y>
      <timeFormatMode>24h</timeFormatMode>
      <dateFormat>year-month-day</dateFormat>
    </time>
    <channelName>
      <switch>true</switch>
      <X>1000</X>
      <Y>3000</Y>
      <name><![CDATA[IPC]]></name>
    </channelName>
    <textOverLay>
      <item>
        <switch type="boolean">false</switch>

```

<X>0</X>

<Y>181</Y>

<osdOverlayType>TEXT</osdOverlayType>

<value>

<item>

<![CDATA[IPC]]>

</item>

</value>

</item>

<item>

<switch type="boolean">>false</switch>

<X>613</X>

<Y>2151</Y>

<osdOverlayType>TEXT</osdOverlayType>

<value>

<item>

<![CDATA[222222222]]>

</item>

<item>

<![CDATA[333333333]]>

</item>

</value>

</item>

<item>

<switch>>false</switch>

<X>727</X>

<Y>2151</Y>

<value>

<item>

<![CDATA[333333333]]>

</item>

</value>

```

        <osdOverlayType>TEXT</osdOverlayType>
    </item>
    <item>
        <switch>>false</switch>
        <X>613</X>
        <Y>2393</Y>
        <value>
            <item>
                <![CDATA[4444444444]]>
            </item>
        </value>
        <osdOverlayType>IMAGE</osdOverlayType>
    </item>
</textOverLay>
<imgDataLen>1024</imgDataLen>
<imgBase64Data>
    <![CDATA[/9j/4AAQSkZJRgABA..... //2Q==]]><!--The content is too long and
    has been omitted.-->
</imgBase64Data>
<osdAttr>
    <transparent>0</transparent>
    <fontSize>auto</fontSize>
    <fontColorMode>customize</fontColorMode>
    <fontColor>0099bb</fontColor>
</osdAttr>
</imageOsd>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution.
2. The content of imgBase64Data is too long and has been omitted from the examples in the document.

3.5 Privacy Mask

3.5.1 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	<privacyMask>
Applicable products	IPC,NVR
<privacyMask> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <privacyMask type="list" maxCount="" count=""><!--Optional;list of privacy mask.--> <item id="1"><!--Required; id represents the identification of privacy masking region.--> <switch type="boolean"><!--Required;privacy mask switch.--></switch> <rectangle><!--Required; location of privacy mask region.--> <X type="uint32"><!-- Required; X coordinate in the upper left corner. --></X> <Y type="uint32"><!-- Required; Y coordinate in the upper left corner.--></Y> <width type="uint32"><!-- Required; region width. --></width> <height type="uint32"><!-- Required; region height. --></height> </rectangle> </item> </privacyMask> </config></pre>	
Example of Successful Response: <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"></pre>	

```
<privacyMask type="list" count="4">
  <item id="1">
    <switch type="boolean">true</switch>
    <rectangle>
      <X type="uint32">102</X>
      <Y type="uint32">136</Y>
      <width type="uint32">116</width>
      <height type="uint32">48</height>
    </rectangle>
  </item>
  <item id="2">
    <switch type="boolean">true</switch>
    <rectangle>
      <X type="uint32">345</X>
      <Y type="uint32">297</Y>
      <width type="uint32">97</width>
      <height type="uint32">55</height>
    </rectangle>
  </item>
  <item id="3">
    <switch type="boolean">true</switch>
    <rectangle>
      <X type="uint32">0</X>
      <Y type="uint32">0</Y>
      <width type="uint32">0</width>
      <height type="uint32">0</height>
    </rectangle>
  </item>
  <item id="4">
    <switch type="boolean">true</switch>
    <rectangle>
      <X type="uint32">0</X>
```

```

        <Y type="uint32">0</Y>
        <width type="uint32">0</width>
        <height type="uint32">0</height>
    </rectangle>
</item>
</privacyMask>
</config>

```

[Tips]:

1.The "X" and "Y" element announce the horizontal and vertical position based in the 640*480 resolution.

3.5.2 AddPrivacyMaskConfig

AddPrivacyMaskConfig	
Description	To add the IP media device's privacy mask element for specific channel.
Typical URL	POST http://<host>[:port]/AddPrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<privacyMask>
Successful Response	<privacyMaskReply>
Applicable products	IPC,NVR

<privacyMask> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
    <privacyMask><!--Required;privacy mask information.-->
        <switch><!--Required;boolean,privacy mask switch.--></switch>
        <rectangle><!--Dependent; location of privacy mask area. valid when switch is 'true'. -->
            <X><!-- Required; uint32,X coordinate in the upper left corner. --></X>
            <Y><!-- Required; uint32,Y coordinate in the upper left corner.--></Y>
            <width><!-- Required; uint32,region width. --></width>
            <height><!-- Required; uint32,region height. --></height>

```

```
</rectangle>
</privacyMask>
</config>

<privacyMaskReply> XML Block:
<config version="" xmlns="http://www.ipc.com/ver10">
  <privacyMaskReply><!--Required.-->
    <id type="uint32"><!--Required;privacy mask region id.--></id>
  </privacyMaskReply>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <privacyMask>
    <switch>true</switch>
    <rectangle>
      <X>102</X>
      <Y>136</Y>
      <width>116</width>
      <height>48</height>
    </rectangle>
  </privacyMask>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <privacyMaskReply>
    <id type="uint32">1</id>
  </privacyMaskReply>
</config>
```

[Tips]:

1.The "X" and "Y" element announce the horizontal and vertical position based in the 640*480 resolution.

3.5.3 ModifyPrivacyMaskConfig

ModifyPrivacyMaskConfig	
Description	To modify the IP media device's privacy mask element for specific channel.
Typical URL	POST http://<host>[:port]/AddPrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<privacyMask>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><privacyMask> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <privacyMask><!--Required;privacy mask information.--> <id><!--Required;uint32,privacy mask region id.--></id> <switch><!--Optional;boolean,privacy mask switch.--></switch> <rectangle><!--Optional; location of privacy mask region. valid when switch is 'true'. --> <X><!-- Required; uint32,X coordinate in the upper left corner. --></X> <Y><!-- Required; uint32,Y coordinate in the upper left corner.--></Y> <width><!-- Required; uint32,region width. --></width> <height><!-- Required; uint32,region height. --></height> </rectangle> </privacyMask> </config></pre>	
<p>Example of Entity Data:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"></pre>	

```

<privacyMask>
  <id>1</id>
  <switch>true</switch>
  <rectangle>
    <X>102</X>
    <Y>136</Y>
    <width>116</width>
    <height>48</height>
  </rectangle>
</privacyMask>
</config>

```

[Tips]:

1.The "X" and "Y" element announce the horizontal and vertical position based in the 640*480 resolution.

3.5.4 DeletePrivacyMaskConfig

DeletePrivacyMaskConfig	
Description	To delete the IP media device's privacy mask element for specific channel.
Typical URL	POST http://<host>[:port]/DeletePrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<privacyMask>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<privacyMask> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <privacyMask><!--Required;privacy mask information.-->
    <id>
      <!--Required:uint32, privacy mask region id,get it through "GetPrivaceMaskConfig" or

```

<pre> "AddPrivaceMaskConfig".--> </id> </privacyMask> </config> </pre>
<p>Example of Entity Data:</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <privacyMask> <id>1</id> </privacyMask> </config> </pre>
<p>[Tips]: The value of id must not exceed the maxCount property in the privacyMask element of the return value of "GetPrivaceMaskConfig".</p>

3.5.5 GotoPrivacyMaskLocation

GotoPrivacyMaskLocation	
Description	To run PTZ to a specific privacy mask location. Only valid for dome camera.
Typical URL	POST http://<host>[:port]/DeletePrivacyMaskConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<privacyMask>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><config> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <privacyMask> <id><!-- Required; uint32,id of privacy mask, get it through "GetPrivaceMaskConfig". --></id> </privacyMask> </pre>	

```
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <privacyMask>
    <id>1</id>
  </privacyMask>
</config>
```

[Tips]: The value of id must not exceed the maxCount property in the privityMask element of the return value of "GetPrivaceMaskConfig".

4 PTZ commands

4.1 Protocol

4.1.1 PtzGetCaps

PtzGetCaps	
Description	To get the IP media device's PTZ capabilities 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	<caps>
Applicable products	IPC,NVR
<p><caps> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <caps> <controlMinSpeed type="uint32"> <!--Required; The minimum speed for the PTZ control command.--> </controlMinSpeed> <controlMaxSpeed type="uint32"> <!--Required; The maximum speed for the PTZ control command.--> </controlMaxSpeed> <presetMaxCount type="uint32"> <!--Required; The maximum number of preset points supported by the system.--> </presetMaxCount> <cruiseMaxCount type="uint32"> <!--Required; The maximum number of patrol routes supported by the system.--> </cruiseMaxCount> <traceMaxCount type="uint32"> <!--Optional; The maximum number of traces supported by the system.NVR is not supported.--> </traceMaxCount> <traceMaxHoldTime type="uint32"> <!--Optional; The maximum trace duration supported by the system. NVR is not supported.--> </traceMaxHoldTime> <cruisePresetMinSpeed type="uint32"> <!--Required; The 'preset point' minimum speed in the cruise line.--> </cruisePresetMinSpeed> <cruisePresetMaxSpeed type="uint32"> <!--Required; The 'preset point' maximum speed in the cruise line.--> </cruisePresetMaxSpeed> <cruisePresetMinHoldTime type="uint32"> </pre>	

```

        <!--Required; The minimum duration of each preset point in the cruise line.-->
    </cruisePresetMinHoldTime>
    <cruisePresetDefaultHoldTime type="uint32">
        <!--Required; The default duration of each preset point in the cruise line.-->
    </cruisePresetDefaultHoldTime>
    <cruisePresetMaxHoldTime type="uint32">
        <!--Required; The maximum duration of each preset point in the cruise line.-->
    </cruisePresetMaxHoldTime>
    <cruisePresetMaxCount type="uint32">
        <!--Required; The maximum number of preset points supported by the cruise line.-->
    </cruisePresetMaxCount>
</caps>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <caps>
        <controlMinSpeed type="uint32">1</controlMinSpeed>
        <controlMaxSpeed type="uint32">32</controlMaxSpeed>
        <presetMaxCount type="uint32">512</presetMaxCount>
        <cruiseMaxCount type="uint32">8</cruiseMaxCount>
        <traceMaxCount type="uint32">8</traceMaxCount>
        <traceMaxHoldTime type="uint32">600</traceMaxHoldTime>
        <cruisePresetMinSpeed type="uint32">1</cruisePresetMinSpeed>
        <cruisePresetMaxSpeed type="uint32">8</cruisePresetMaxSpeed>
        <cruisePresetMinHoldTime type="uint32">5</cruisePresetMinHoldTime>
        <cruisePresetDefaultHoldTime type="uint32">30</cruisePresetDefaultHoldTime>
        <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.

4.1.2 GetPtzConfig

GetPtzConfig	
Description	To get the IP media device's PTZ protocol configuration for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetPtzConfig[/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	<ptzSettings>
Applicable products	IPC
<p><ptzSettings> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <types> <languageE><!--Optional; define OSD of PTZ language.--> <enum>cn</enum><!--Optional; Chinese.--> <enum>en</enum><!--Optional; English.--> </languageE> <autoExitTimeE><!--Optional; OSD menu hold time.--> <enum>off</enum> <enum>15sec</enum> <enum>30sec</enum> <enum>60sec</enum> <enum>90sec</enum> <enum>120sec</enum> </autoExitTimeE> <protocolE><!--Optional; PTZ protocol type for external RS485.--></pre>	

```

        <enum>PELCOP</enum>
        <enum>PELCOD</enum>
    </protocolE>
    <baudRateE><!--Optional; Baud rate of external RS485 PTZ protocol.-->
        <enum>1200</enum>
        <enum>2400</enum>
        <enum>4800</enum>
        <enum>9600</enum>
    </baudRateE>
</types>
<ptzSettings><!--Required; PTZ config information.-->
    <autoPtzFlip type="boolean">
        <!--Optional; when the PTZ rotates vertically to the maximum angle, it triggers flipping.-->
    </autoPtzFlip>
    <language type="languageE">
        <!--Optional; Language of PTZ OSD.-->
    </language>
    <autoExitTime type="autoExitTimeE">
        <!--Optional; OSD menu hold time.-->
    </autoExitTime>
    <rs485><!--Optional; RS485 transmission parameter configuration.-->
        <demoId type="uint8"><!--Required; Device ID.--></demoId>
        <protocol type="protocolE"><!--Required; protocol type.--></protocol>
        <baudRate type="baudRateE"><!--Required; baud rate.--></baudRate>
    </rs485>
    <savePtzPosition><!--Optional; Save PTZ position before power off.-->
        <switch type="boolean"><!--Required;Function switch.--></switch>
        <time type="uint32" min="" max="" default="">
            <!--Dependent;Valid when switch is 'true', If the duration of the PTZ stop state reaches
            the time threshold, record the PTZ position.-->
        </time>
    </savePtzPosition>
</ptzSettings>

```

```
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <languageE>
      <enum>cn</enum>
    </languageE>
    <autoExitTimeE>
      <enum>off</enum>
      <enum>15sec</enum>
      <enum>30sec</enum>
      <enum>60sec</enum>
      <enum>90sec</enum>
      <enum>120sec</enum>
    </autoExitTimeE>
  </types>
  <ptzSettings>
    <autoPtzFlip type="boolean">true</autoPtzFlip>
    <language type="languageE">cn</language>
    <autoExitTime type="autoExitTimeE">off</autoExitTime>
    <savePtzPosition>
      <switch type="boolean">false</switch>
      <time type="uint32" min="10" max="600" default="30">30</time>
    </savePtzPosition>
  </ptzSettings>
</config>
```

[Tips]:

4.1.3 SetPtzConfig

SetPtzConfig

Description	To set the IP media device's PTZ protocol configuration for specific channel.
Typical URL	POST http://<host>[:port]/SetPtzConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<ptzSettings>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC

<ptzSettings> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <ptzSettings><!--Required; PTZ config information.-->
    <autoPtzFlip>
      <!--Optional; boolean,when the PTZ rotates vertically to the maximum angle, it triggers
      flipping.-->
    </autoPtzFlip>
    <language>
      <!--Optional; languageE,Language of PTZ OSD.-->
    </language>
    <autoExitTime>
      <!--Optional; autoExitTimeE,OSD menu hold time.-->
    </autoExitTime>
    <rs485><!--Optional; RS485 transmission parameter configuration.-->
      <demoId><!--Required; uint8,Device ID.--></demoId>
      <protocol><!--Required; protocolE, protocol type.--></protocol>
      <baudRate><!--Required; baudRateE,baud rate.--></baudRate>
    </rs485>
    <savePtzPosition><!--Optional; Save PTZ position before power off.-->
      <switch><!--Required;boolean,Function switch.--></switch>
      <time>
        <!--Dependent; uint32,Valid when switch is 'true', If the duration of the PTZ stop state
        reaches the time threshold, record the PTZ position.-->

```

```

        </time>
    </savePtzPosition>
</ptzSettings>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <ptzSettings>
        <autoPtzFlip>true</autoPtzFlip>
        <language>cn</language>
        <autoExitTime>off</autoExitTime>
        <savePtzPosition>
            <switch>>false</switch>
            <time>30</time>
        </savePtzPosition>
    </ptzSettings>
</config>

```

[Tips]:

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.

Action name	<p>Up: to move up</p> <p>Down: to move down</p> <p>Left: to move left</p> <p>Right: to move right</p> <p>LeftUp: to move left and up</p> <p>LeftDown: to move left and down</p> <p>RightUp: to move right and up</p> <p>RightDown: to move right and down</p> <p>Near: to focus near</p> <p>Far: to focus far</p> <p>ZoomIn: to zoom in</p> <p>ZoomOut: to zoom out</p> <p>IrisOpen: to open the iris</p> <p>IrisClose: to close the iris</p> <p>WiperOn: to open the wiper.</p> <p>WiperOff: to close the wiper.</p> <p>ScanOn: to open Auto scan.</p> <p>ScanOff: to close Auto scan.</p> <p>LightOn: to open the light.</p> <p>LightOff: to close the light.</p> <p>LightAuto: to set auto light mode.</p> <p>RandomScanOn: to open random scan.</p> <p>RandomScanOff: to close random scan.</p> <p>GroupRun: to run the curises of group.</p> <p>GroupStop: to run the curises of group.</p> <p>Stop: to stop current action</p>
Entity Data	<actionInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<actionInfo> XML Block:

```
<actionInfo version="" xmlns="http://www.ipc.com/ver10">
  <speed><!--Optional;uint32,PTZ movement speed during control.--></speed>
</actionInfo>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8" ?>
<actionInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <speed>5</speed>
</actionInfo>
```

[Tips]:

1.The value of "speed" should be in the scope of the corresponding capability announced in the "PtzGetCaps" successful respond message.

4.2.2 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	<presetInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<presetInfo> XML Block:	
<pre><presetInfo version="" xmlns="http://www.ipc.com/ver10"> <id><!--Required;uint32,Id of preset.--></id> </presetInfo></pre>	
Example of Entity Data:	

```
<?xml version="1.0" encoding="utf-8" ?>
<presetInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>50</id>
</presetInfo>
```

[Tips]:

1. The maximum value of id is the field 'presetMaxCount' in the successful response of calling 'PtzGetCaps'.

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	<cruiseInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<cruiseInfo> XML Block: <pre><cruiseInfo version="" xmlns="http://www.ipc.com/ver10"> <id><!--Required;uint32,Id of patrol.--></id> </cruiseInfo></pre>	
Example of Entity Data: <pre><?xml version="1.0" encoding="utf-8" ?> <cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> </cruiseInfo></pre>	

[Tips]:

1. The maximum value of id is the field 'cruiseMaxCount' in the successful response of calling 'PtzGetCaps'.

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.
Applicable products	IPC,NVR

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	<presetInfo>
Applicable products	IPC,NVR

<presetInfo> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <presetInfo type="list" maxCount="" count=""><!--Required;List of preset Info.-->
    <itemType type="string" maxLen=""></itemType>
    <item id=""><!--The attribute 'id' represents the preset point ID.-->
      <!--Required; string, preset name.-->
    </item>
  </presetInfo>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <presetInfo type="list" maxCount="512" count="2">
    <itemType type="string" maxLen="10"></itemType>
    <item id="1"><![CDATA[preset1]]></item>
    <item id="2"><![CDATA[preset3]]></item>
  </presetInfo>
</config>

```

[Tips]:

1.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	<presetInfo>

Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><presetInfo> XML Block:</p> <pre><presetInfo version="" xmlns="http://www.ipc.com/ver10"> <id><!--Required; uint32,preset id.--></id> <name><!--Required; string,preset name.--></name> </presetInfo></pre>	
<p>Example of Entity Data:</p> <pre><?xml version="1.0" encoding="utf-8" ?> <presetInfo version="2.0.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> <name><![CDATA[preset1]]</name> </presetInfo></pre>	
<p>[Tips]:</p>	

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	<presetInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<presetInfo> XML Block:

```
<presetInfo version="" xmlns="http://www.ipc.com/ver10">
  <id><!--Required; uint32,preset id.--></id>
  <name><!--Required; string,preset name.--></name>
</presetInfo>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8" ?>
<presetInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>1</id>
  <name><![CDATA[preset1]]></name>
</presetInfo>
```

[Tips]:

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	<presetInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<presetInfo> XML Block: <pre><presetInfo version="" xmlns="http://www.ipc.com/ver10"> <id><!--Required; uint32,preset id.--></id> </presetInfo></pre>	

PtzDeletePreset

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8" ?>
<presetInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>1</id>
</presetInfo>
```

[Tips]:

- 1.The PTZ preset's ID that needs to be deleted will be included in the entity of the request message.

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	<presetInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<presetInfo> XML Block:

```
<presetInfo version="" xmlns="http://www.ipc.com/ver10">
  <id><!--Required; uint32,preset id.--></id>
</presetInfo>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8" ?>
<presetInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>1</id>
</presetInfo>
```

[Tips]:

4.4 Cruise

4.4.1 PtzGetCruises

PtzGetCruises	
Description	To get the IP media device's PTZ cruises list for specific channel.
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	<cruiseInfo>
Applicable products	IPC,NVR
<cruiseInfo> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <cruiseInfo type="list" maxCount="" count=""><!--Required.List of cruise information.--> <itemType type="string" maxLen=""></itemType> <item id="1"><!--The attribute 'id' represents the cruise ID.--> <!--Required; string, Name of patrol route.--> </item> </cruiseInfo> </config></pre>	

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <cruiseInfo type="list" count="2">
    <itemType type="string" maxLen=""></itemType>
    <item id="1"><![CDATA[Patrol1]]></item>
    <item id="2"><![CDATA[Patrol2]]></item>
  </cruiseInfo>
</config>
```

[Tips]:

1.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]/PtzGetCruise[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<cruiseInfo> of Entity Data
Successful Response	<cruiseInfo> of Successful Response
Applicable products	IPC,NVR

<cruiseInfo> of Entity Data XML Block:

```
<cruiseInfo version="" xmlns="http://www.ipc.com/ver10">
  <id><!--Required; uint32, cruise ID.--></id>
</cruiseInfo>
```

<cruiseInfo> of Successful Response XML Block:

```
<?xml version="1.0" encoding="UTF-8"?>
<cruiseInfo version="" xmlns="http://www.ipc.com/ver10">
  <id type="uint32"><!--Required; uint32, cruise ID.--></id>
  <name type="string" maxLen="">
    <!--Required; string, Name of patrol route.-->
  </name>
  <presetInfo type="list" maxCount="" count=""><!--Required; List of preset information.-->
    <item id="1"><!--Required. The attribute 'id' represents the preset ID.-->
      <name type="string" maxLen=""><!--Required. Name of preset point.--></name>
      <speed type="uint32" min="" max="">
        <!--Required. Preset point movement speed in patrol route.-->
      </speed>
      <holdTime type="uint32" min="" max="">
        <!--Required. Duration of each preset point in patrol route.-->
      </holdTime>
    </item>
  </presetInfo>
</cruiseInfo>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8" ?>
<cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>1</id>
</cruiseInfo>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id type="uint32">1</id>
  <name type="string" maxLen="31">
    <![CDATA[Patrol1]]>
  </name>
  <presetInfo type="list" maxCount="16" count="2">
    <item id="1">
      <name type="string" maxLen="11"><![CDATA[Preset1]]></name>
      <speed type="uint32" min="1" max="8">5</speed>
      <holdTime type="uint32" min="5" max="240">5</holdTime>
    </item>
    <item id="2">
      <name type="string" maxLen="11"><![CDATA[Preset2]]></name>
      <speed type="uint32" min="1" max="8">5</speed>
      <holdTime type="uint32" min="5" max="240">10</holdTime>
    </item>
  </presetInfo>
</cruiseInfo>
```

[Tips]:

1.The "id" attribute for each item starts from "1".

4.4.3 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	<cruiseInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><cruiseInfo> XML Block:</p> <pre> <cruiseInfo version="" xmlns="http://www.ipc.com/ver10"> <name><!--Required; string, Name of patrol route.--></name> <presetInfo><!--Required; List of preset information.--> <item id=""><!--Required. The attribute 'id' represents the preset ID.--> <speed><!--Required. uint32,Preset point movement speed in patrol route.--></speed> <holdTime> <!--Required. uint32,Duration of each preset point in patrol route.--> </holdTime> </item> </presetInfo> </cruiseInfo> </pre>	

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <name type="string" maxLen="31">
    <![CDATA[Patrol1]]>
  </name>
  <presetInfo count="2">
    <item id="1">
      <speed>5</speed>
      <holdTime>5</holdTime>
    </item>
    <item id="2">
      <speed>5</speed>
      <holdTime>10</holdTime>
    </item>
  </presetInfo>
</cruiseInfo>
```

[Tips]:

4.4.4 PtzModifyCruise

PtzModifyCruise	
Description	To modify one cruise information of the IP media device's specific channel.
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	<cruiseInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<cruiseInfo> XML Block:

```
<cruiseInfo version="" xmlns="http://www.ipc.com/ver10">
  <id><!--Required; uint32, cruise ID.--></id>
  <name><!--Required; string, Name of patrol route.--></name>
  <presetInfo><!--Required; List of preset information.-->
    <item id=""><!--Required. The attribute 'id' represents the preset ID.-->
      <speed><!--Optional. uint32,Preset point movement speed in patrol route.--></speed>
      <holdTime>
        <!--Optional. uint32,Duration of each preset point in patrol route.-->
      </holdTime>
    </item>
  </presetInfo>
</cruiseInfo>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <id>1</id>
  <name><![CDATA[Patrol1]]> </name>
  <presetInfo count="2">
    <item id="1">
      <speed>5</speed>
      <holdTime>20</holdTime>
    </item>
    <item id="2">
      <speed>5</speed>
      <holdTime>30</holdTime>
    </item>
  </presetInfo>
</cruiseInfo>
```

[Tips]:

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	<cruiseInfo>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<cruiseInfo> XML Block: <pre><cruiseInfo version="" xmlns="http://www.ipc.com/ver10"> <id><!--Required; uint32, cruise ID that needs to be deleted.--></id> </cruiseInfo></pre>	
Example of Entity Data: <pre><?xml version="1.0" encoding="utf-8" ?> <cruiseInfo version="2.0.0" xmlns="http://www.ipc.com/ver10"> <id>1</id> </cruiseInfo></pre>	
[Tips]:	

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	<motion>
Applicable products	IPC,NVR
<p><motion> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <motion><!--Required;motion configuration information.--> <switch type="boolean"><!--Required;motion function enabled or not.--></switch> <alarmHoldTime type="uint32" min="" max="" default=""> <!--Required; The duration of the alarm after the event ends.--> </alarmHoldTime> <objectFilter><!--Optional; Motion is triggered only when a specified type of target is detected.--> <car><!--Optional; Detecting Motion events of car targets.--> <switch type="boolean"><!--Required;Detection or not.--></switch></pre>	

```

    </car>
    <person><!--Required; Detecting Motion events of person targets.-->
        <switch type="boolean"><!--Required;Detection or not.--></switch>
    </person>
</objectFilter>
<area type="list" maxCount="" count=""><!--Required; Grid of the area to be detected.-->
    <itemType type="string" minLen="" maxLen=""/>
    <item>
        <!--Required; Mark whether each grid is detected, 1: Need to detect, 0: No need to
        detect, each character represents a grid.refer to the description in "Tips" for details. eg:
        <![CDATA[111111111100000011111]]>.-->
    </item>
</area>
<sensitivities type="list" maxCount="" count=""><!--Optional; Sensitivity of each grid.-->
    <itemType type="string" minLen="" maxLen=""/>
    <item>
        <!--Required; Mark whether each grid is detected, 1: Need to detect, 0: No need to
        detect, each character represents a grid.refer to the description in "Tips" for details. eg:
        <![CDATA[4400000000000000000000]]>.-->
    </item>
</sensitivities>
</motion>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <motion>
        <switch type="boolean">true</switch>
        <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
    <objectFilter>
        <car>
            <switch type="boolean">>false</switch>
        </car>
    </objectFilter>

```

```
<![CDATA[444444444444444444444444444444444444]]>
</item>
<item>
  <![CDATA[444444444444444444444444444444444444]]>
</item>
<item>
  <![CDATA[444444111114444488888884]]>
</item>
<item>
  <![CDATA[444444111115555588888884]]>
</item>
<item>
  <![CDATA[444444111115555588888884]]>
</item>
<item>
  <![CDATA[444444444455555588888888]]>
</item>
<item>
  <![CDATA[444444444455555588888888]]>
</item>
<item>
  <![CDATA[444444444455555588888888]]>
</item>
<item>
  <![CDATA[4466666666666658888888]]>
</item>
<item>
  <![CDATA[4466666666666658888888]]>
</item>
<item>
  <![CDATA[4466666666666658888888]]>
</item>
<item>
  <![CDATA[4466666666666658888888]]>
</item>
<item>
  <![CDATA[44666666666666777788888888]]>
</item>
```

```
<item>
  <![CDATA[44666666666677778888888]]>
</item>
<item>
  <![CDATA[4466666666667777744444]]>
</item>
<item>
  <![CDATA[4444444444447777744444]]>
</item>
<item>
  <![CDATA[4444444444447777744444]]>
</item>
<item>
  <![CDATA[4444444444444444440000]]>
</item>
<item>
  <![CDATA[4400000000000000000000]]>
</item>
<item>
  <![CDATA[4400000000000000000000]]>
</item>
</sensitivities>
</motion>
</config>
```

[Tips]:

1. 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 definition of sensitivities is similar, indicating the sensitivity of the detection area.
2. Please refer to "GetAlarmTriggerConfig/SetAlarmTriggerConfig" for alarm linkage and "GetScheduleConfig/SetScheduleConfig" for schedule configuration.

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]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<motion>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<motion> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <motion><!--Required;motion configuration information.-->
    <switch><!--Optional;motion function enabled or not.--></switch>
    <alarmHoldTime>
      <!--Optional; Alarm duration after event triggering ends.-->
    </alarmHoldTime>
    <objectFilter><!--Optional; Motion is triggered only when a specified type of target is
detected-->
      <car><!--Required; Detecting Motion events of car targets.-->
        <switch><!--Required;Detection or not.--></switch>
      </car>
      <person><!--Required; Detecting Motion events of person targets.-->
        <switch><!--Required;Detection or not.--></switch>
      </person>
    </objectFilter>
    <area>
      <!--Optional; Enable or disable Motion detection in each grid area.-->
      <item>
        <!--Required; Mark whether each grid is detected, 1: Need to detect, 0: No need to
detect, each character represents a grid.refer to the description in "Tips" for details. eg:
<![CDATA[111111111100000011111]]>.-->
      </item>

```

```

</area>

<sensitivities><!--Optional; Sensitivity of each grid.-->
  <item>
    <!--Required; Mark whether each grid is detected, 1: Need to detect, 0: No need to
    detect, each character represents a grid.refer to the description in "Tips" for details. eg:
    <![CDATA[44000000000000000000]]>.-->

  </item>
</sensitivities>

</motion>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <motion>
    <switch>true</switch>
    <alarmHoldTime>20</alarmHoldTime>
    <objectFilter>
      <car>
        <switch>>false</switch>
      </car>
      <person>
        <switch>>false</switch>
      </person>
    </objectFilter>
    <area>
      <item>
        <![CDATA[11111111111111111111]]>
      </item>
      <item>
        <![CDATA[11111111111111111111]]>
      </item>
      <item>
        <![CDATA[11111111111111111111]]>

```

```
<![CDATA[11111111111111111111]]>
</item>
<item>
  <![CDATA[11111111111111111111]]>
</item>
<item>
  <![CDATA[111111111111111110000]]>
</item>
<item>
  <![CDATA[11000000000000000000]]>
</item>
<item>
  <![CDATA[11000000000000000000]]>
</item>
</area>
<sensitivities>
  <item>
    <![CDATA[44444444444444444444]]>
  </item>
  <item>
    <![CDATA[44444444444444444444]]>
  </item>
  <item>
    <![CDATA[4444411111444448888884]]>
  </item>
  <item>
    <![CDATA[444441111155558888884]]>
  </item>
  <item>
    <![CDATA[444441111155558888884]]>
  </item>
  <item>
```

```
<![CDATA[44444444455555888888]]>
</item>
<item>
  <![CDATA[44444444455555888888]]>
</item>
<item>
  <![CDATA[44666666666658888888]]>
</item>
<item>
  <![CDATA[44666666666658888888]]>
</item>
<item>
  <![CDATA[44666666666658888888]]>
</item>
<item>
  <![CDATA[44666666666658888888]]>
</item>
<item>
  <![CDATA[4466666666667777888888]]>
</item>
<item>
  <![CDATA[4466666666667777888888]]>
</item>
<item>
  <![CDATA[4466666666667777744444]]>
</item>
<item>
  <![CDATA[44444444447777744444]]>
</item>
<item>
  <![CDATA[44444444447777744444]]>
</item>
<item>
  <![CDATA[44444444447777744444]]>
</item>
<item>
  <![CDATA[44444444444444440000]]>
</item>
```

```

        <item>
            <![CDATA[4400000000000000000000]]>
        </item>
        <item>
            <![CDATA[4400000000000000000000]]>
        </item>
    </sensitivities>
</motion>
</config>

```

[Tips]:

1. 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 definition of sensitivities is similar, indicating the sensitivity of the detection area.
2. Please refer to "**GetAlarmTriggerConfig/SetAlarmTriggerConfig**" for alarm linkage and "**GetScheduleConfig/SetScheduleConfig**" for schedule configuration.

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[id]
Sensor ID	Optional. If none id included in the URL, the default Sensor ID is 1
Action name	None
Entity Data	None
Successful Response	<sensor>
Applicable products	IPC,NVR
<sensor> XML Block:	

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <alarmInVoltage><!--Required.-->
      <enum>NO</enum>
      <enum>NC</enum>
    </alarmInVoltage>
  </types>
  <sensor><!--Required;Alarm in configuration.-->
    <id type="uint32"><!--Required;Alarm in ID.--></id>
    <sensorName type="string" maxCharNum="">
      <!--Optional;Name of alarm in.-->
    </sensorName>
    <switch type="boolean"><!--Required;Alarm in enabled or not.--></switch>
    <voltage type="alarmInVoltage"><!--Required;Voltage type of alarm in.--></voltage>
    <alarmHoldTime type="uint32" min="" max="" default="">
      <!--Required; The duration after the end of the Alarm in trigger.-->
    </alarmHoldTime>
  </sensor>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <alarmInVoltage>
      <enum>NO</enum>
      <enum>NC</enum>
    </alarmInVoltage>
  </types>
  <sensor>
    <sensorName type="string" maxLen="12">
      <![CDATA[111]]>
    </sensorName>
  </sensor>
</config>

```

```

    <switch type="boolean">true</switch>
    <voltage type="alarmInVoltage">NO</voltage>
    <alarmHoldTime type="uint32" min="1" max="120" default="20">10</alarmHoldTime>
  </sensor>
</config>

```

[Tips]:

1. Please refer to "**GetAlarmTriggerConfig/SetAlarmTriggerConfig**" for alarm linkage and "**GetScheduleConfig/SetScheduleConfig**" for schedule configuration.
2. The 'id' in the URL, refer to the successful response of "**GetAlarmInInfo**".

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[/id]
Sensor ID	Optional. If none id included in the URL, the default sensor ID is 1.
Action name	None
Entity Data	< sensor >
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<sensor> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <sensor><!--Required;Alarm in configuration.-->
    <sensorName>
      <!--Optional;string,Name of alarm in.-->
    </sensorName>
    <switch><!--Required;boolean,Alarm in enabled or not.--></switch>
    <voltage><!--Required;alarmInVoltage,Voltage type of alarm in.--></voltage>
    <alarmHoldTime>
      <!--Required;uint32,The duration after the end of the Alarm in trigger.-->
    </alarmHoldTime>
  </sensor>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <sensor>
    <sensorName><![CDATA[111]]></sensorName>
    <switch>true</switch>
    <voltage>NO</voltage>
    <alarmHoldTime>10</alarmHoldTime>
  </sensor>
</config>
```

[Tips]:

1.Please refer to "**GetAlarmTriggerConfig/SetAlarmTriggerConfig**" for alarm linkage and "**GetScheduleConfig/SetScheduleConfig**" for schedule configuration.

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[/id]
Sensor ID	Optional. If none id included in the URL, the default sensor ID is 1.
Action name	None
Entity Data	<action>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><action> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <action><!--Required.--> <status><!--Required;boolean, alarm output on or off, true: on, false:off.--></status> </action> </config></pre>	
<p>Example of Entity Data:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <action> <status>>true</status> </action> </config></pre>	
<p>[Tips]:</p> <p>1. The duration of manually triggering the alarm output is "alarmHoldTime" in the successful response of calling "GetAlarmOutConfig".</p>	

5.2.4 GetAlarmOutConfig

GetAlarmOutConfig	
Description	To get the IP media device's alarm output configuration for specific alarm output channel.
Typical URL	POST or GET http://<host>[:port]/GetAlarmOutConfig[/id]

Sensor ID	Optional. If none id included in the URL, the default sensor ID is 1.
Action name	None
Entity Data	None
Successful Response	<alarmOut>
Applicable products	IPC,NVR

<alarmOut> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <alarmOutMode><!--Required.-->
      <enum>alarmLinkage</enum><!--Required. Alarm output triggered by other events.-->
      <enum>manualAlarm</enum><!--Optional. Alarm output triggered manually, NVR is not supported.-->
      <enum>dayNightChange</enum><!--Optional. The alarm output is triggered when switching between image color mode and "black and white mode".NVR is not supported.-->
      <enum>timerAlarm</enum><!--Optional. Alarm output triggered by specified time period, NVR is not supported.-->
    </alarmOutMode>
    <outputVoltageType><!--Optional.alarm out voltage type.NVR is not supported.-->
      <enum>NO</enum><!--Normally Open.-->
      <enum>NC</enum><!--Normally Close.-->
    </outputVoltageType>
    <dayNightMode><!--Optional.NVR is not supported.-->
      <enum>off</enum><!--none. compatible with historical versions-->
      <enum>day</enum><!--image color mode.-->
      <enum>night</enum><!--black and white mode.-->
      <enum>dayAndNight</enum><!--both day and night. Compatible with historical versions.-->
    </dayNightMode >
  </types>
  <alarmOut>
    <id type="uint32"><!--Required.alarm out ID.--></id>

```

```

<typeAlarmOut type="alarmOutMode"><!--Required.alarm out mode.--></typeAlarmOut>
<dayPeroid type="list" maxCount="" count=""><!--Dependent; Valid when
typeAlarmOut='timerAlarm', used to specify the time period for alarm output.-->
  <item>
    <startTime type="string"><!--Required; start time, Format HH:MM.--></startTime>
    <endTime type="string"><!--Required; end time, Format HH:MM.--></endTime>
  </item>
</dayPeroid>
<dayNightLinkage><!--Dependent. alid when typeAlarmOut= "dayNightChange".-->
  <channelId type="uint32"><!--Optional;image channel ID.default: 1.--><channelId/>
  <imageMode type="dayNightMode">
    <!--Required; When the image of the specified channel switches to this mode, an
alarm output is triggered.-->
  </imageMode/>
</dayNightLinkage>
  <alarmOutName type="string" maxCharNum="">
    <!--Optional.the name of alarm out.-->
  </alarmOutName>
  <alarmHoldTime type="uint32" min="1" max="120" default="20">
    <!--Required. The duration after the alarm output is no longer triggered.-->
  </alarmHoldTime>
  <outputVoltage type="outputVoltageType">
    <!--Optional. alarm out voltage type, NO or NC.-->
  </outputVoltage>
</alarmOut>
</config>

```

Example of Successful Response(IPC "alarmLinkage"):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <alarmOutMode>
      <enum>alarmLinkage</enum>
      <enum>manualAlarm</enum>

```

```

        <enum>dayNightChange</enum>
        <enum>timerAlarm</enum>
    </alarmOutMode>
    <outputVoltageType>
        <enum>NO</enum>
        <enum>NC</enum>
    </outputVoltageType>
    <dayNightMode>
        <enum>off</enum>
        <enum>day</enum>
        <enum>night</enum>
        <enum>dayAndNight</enum>
    </dayNightMode>
</types>
<alarmOut>
    <typeAlarmOut type="alarmOutMode">alarmLinkage</typeAlarmOut>
    <alarmOutName type="string" maxCharNum="12">
        <![CDATA[alarmOut1]]>
    </alarmOutName>
    <alarmHoldTime type="uint32" min="1" max="120" default="20">5</alarmHoldTime>
    <outputVoltage type="outputVoltageType">NC</outputVoltage>
</alarmOut>
</config>

```

Example of Successful Response(IPC "timeAlarm"):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <alarmOutMode>
            <enum>alarmLinkage</enum>
            <enum>manualAlarm</enum>
            <enum>dayNightChange</enum>

```

```

        <enum>timerAlarm</enum>
    </alarmOutMode>
    <outputVoltageType>
        <enum>NO</enum>
        <enum>NC</enum>
    </outputVoltageType>
    <dayNightMode>
        <enum>off</enum>
        <enum>day</enum>
        <enum>night</enum>
        <enum>dayAndNight</enum>
    </dayNightMode>
</types>
<alarmOut>
    <typeAlarmOut type="alarmOutMode">timerAlarm</typeAlarmOut>
    <dayPeroid type="list" maxCount="100" count="1">
        <item>
            <startTime type="string"><![CDATA[08:00]]></startTime>
            <endTime type="string"><![CDATA[18:00]]></endTime>
        </item>
    </dayPeroid>
    <alarmOutName type="string" maxCharNum="12">
        <![CDATA[alarmOut1]]>
    </alarmOutName>
    <alarmHoldTime type="uint32" min="1" max="120" default="20">5</alarmHoldTime>
    <outputVoltage type="outputVoltageType">NC</outputVoltage>
</alarmOut>
</config>

```

Example of Successful Response(NVR):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">

```

```

<types>
  <alarmOutMode>
    <enum>alarmLinkage</enum>
  </alarmOutMode>
</types>
<alarmOut>
  <typeAlarmOut type="alarmOutMode">alarmLinkage</typeAlarmOut>
  <alarmOutName type="string" maxCharNum="12">
    <![CDATA[alarmOut1]]>
  </alarmOutName>
  <alarmHoldTime type="uint32" min="1" max="120" default="20">5</alarmHoldTime>
</alarmOut>
</config>

```

[Tips]:

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[/id]
Sensor ID	Optional. If none id included in the URL, the default sensor ID is 1.
Action name	None
Entity Data	<alarmOut>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<alarmOut> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <alarmOut>	

```

<typeAlarmOut><!--Required. alarmOutMode,alarm out mode.--></typeAlarmOut>
<dayPeroid><!--Dependent; Valid when typeAlarmOut='timerAlarm', used to specify the time
period for alarm output.-->
  <item>
    <startTime><!--Required; string,start time, Format HH:MM.--></startTime>
    <endTime><!--Required; string,end time, Format HH:MM.--></endTime>
  </item>
</dayPeroid>
<dayNightLinkage><!--Dependent. alid when typeAlarmOut= "dayNightChange".-->
  <channelId><!--Optional;uint32,image channel ID, default:channel 1.--><channelId/>
  <imageMode>
    <!--Required;dayNightMode, When the image of the specified channel switches to this
mode, an alarm output is triggered.-->
  <imageMode/>
</dayNightLinkage>
  <alarmOutName>
    <!--Optional.string, Limit the number of characters,the name of alarm out.-->
  </alarmOutName>
  <alarmHoldTime>
    <!--Required. Uint32,The duration after the alarm output is no longer triggered.-->
  </alarmHoldTime>
  <outputVoltage>
    <!--Optional. outputVoltageType, alarm out voltage type, NO or NC.-->
  </outputVoltage>
</alarmOut>
</config>

```

Example of Entity Data (IPC "alarmLinkage"):

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmOut>
    <typeAlarmOut>alarmLinkage</typeAlarmOut>
    <alarmOutName><![CDATA[alarmOut1]]></alarmOutName>
    <alarmHoldTime>5</alarmHoldTime>
  </alarmOut>
</config>

```

```
<outputVoltage>NC</outputVoltage>
</alarmOut>
</config>
```

Example of Entity Data(IPC "timeAlarm"):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmOut>
    <typeAlarmOut>timerAlarm</typeAlarmOut>
    <dayPeroid>
      <item>
        <startTime><![CDATA[08:00]]></startTime>
        <endTime><![CDATA[18:00]]></endTime>
      </item>
    </dayPeroid>
    <alarmOutName><![CDATA[alarmOut1]]> </alarmOutName>
    <alarmHoldTime>5</alarmHoldTime>
    <outputVoltage>NC</outputVoltage>
  </alarmOut>
</config>
```

Example of Entity Data(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmOut>
    <typeAlarmOut>alarmLinkage</typeAlarmOut>
    <alarmOutName><![CDATA[alarmOut1]]> </alarmOutName>
    <alarmHoldTime>5</alarmHoldTime>
  </alarmOut>
</config>
```

[Tips]:

5.2.6 AlarmOutputControl

AlarmOutputControl	
Description	To manually set the IP media device's alarm output status(Including light alarm, sound alarm, alarm output and auto track) for specific alarm output channel. " AlarmOutputControl " can control light alarms, sound alarms, alarm outputs and auto track(for dome camera). " ManualAlarmOut " can only control alarm output.
Typical URL	POST http://<host>[:port]/AlarmOutputControl
Sensor ID	None
Action name	None
Entity Data	<config>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><config> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <switch> <!--Required;boolean, alarm status that need to be set, on or off, true: on, false:off. --> </switch> <alarmOutputType> <!--Required;string, specify the type of alarm that needs to be controlled.Value in {"io","flashingLight","audioAlarm"}.--> </alarmOutputType> <id> <!--Dependent:uint32,valid when alarmOutputType set to 'io'. Alarm out ID.--> </id> </config></pre>	

Example of Entity Data(io):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <switch>true</switch>
  <alarmOutputType>io</alarmOutputType>
  <id>1</id>
</config>
```

Example of Entity Data(audioAlarm):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <switch>true</switch>
  <alarmOutputType>audioAlarm</alarmOutputType>
</config>
```

[Tips]:

- 1.A new alarm output control interface is added to control the general alarm output items.
- 2.The "io" node represents control relay output;
- 3.The "flashingLigh"t node indicates the control of flashing alarm;
- 4."audioAlarm" node indicates control sound alarm;
- 5.NVR is not supported.

5.3 AlarmStatus

5.3.1 GetAlarmStatusInfo

GetAlarmStatusInfo	
Description	To get the IP media device's alarm trigger status.
Typical URL	POST or GET http://<host>[:port]/GetAlarmStatusInfo
Channel ID	None
Action name	None

Entity Data	None
Successful Response	<config>
Applicable products	IPC
<p><config>XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <alarmStatusInfo><!--Optional.the status of the alarm source.--> <deviceAlarmStatus><!--Optional.--> <diskFull type="boolean"> <!--Optional. Disk full alarm status.--> </diskFull> <noDisk type="boolean"> <!--Optional. No disk alarm status.Only valid for NVR.--> </noDisk> <hddPullOut type="boolean"> <!--Optional. Hard drive pulled out alarm status.Only valid for NVR.--> </hddPullOut> <diskRWEror type="list" maxCount="" count=""><!--Optional.--> <item> <sn type="string"> <!--Optional. SN of the hard drive.Only valid for hard drive. SD card is not required.--> </sn> <status type="boolean"> <!--Required. Status.--> <status> </item> </diskRWEror> <illegalAccess type="boolean"> <!--Optional. Illegal login alarm.--> </illegalAccess> </pre>	

```
</deviceAlarmStatus>

<sensorAlarmIn type="list" maxCount="" count=""><!--Optional.the status of alarm
in.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of alarm in. the attribute 'id' represents Alarm in
ID(uint32), refer to the successful response of GetAlarmInInfo.-->
  </item>
</sensorAlarmIn>

<motionAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of motion alarm. the attribute 'id' represents channel
ID.-->
  </item>
</motionAlarm>

<perimeterAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of perimeter alarm. the attribute 'id' represents channel
ID.-->
  </item>
</perimeterAlarm>

<tripwireAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of tripwire alarm. the attribute 'id' represents channel
ID.-->
  </item>
</tripwireAlarm>

<oscAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
```

```
<item id="1">
    <!--Required.the status of OSC(Object Status Change alarm. the attribute 'id'
    represents channel ID.-->
</item>
</oscAlarm>
<sceneChange type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the scene change alarm. the attribute 'id' represents
        channel ID.-->
    </item>
</sceneChange>
<clarityAbnormal type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the clarity abnormal alarm. the attribute 'id' represents
        channel ID.-->
    </item>
</clarityAbnormal>
<colorAbnormal type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the color abnormal alarm. the attribute 'id' represents
        channel ID.-->
    </item>
</colorAbnormal>
<vfdAlarm type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the video face detection alarm. the attribute 'id'
        represents channel ID.-->
    </item>
</vfdAlarm>
<vehicleAlarm type="list" maxCount="" count=""><!--Optional.-->
```

```
<itemType type="boolean"/>
<item id="1">
    <!--Required.the status of the vehicle detection alarm. the attribute 'id'
    represents channel ID.-->
</item>
</vehicleAlarm>
<plateMatchAlarm type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the license plate match alarm. the attribute 'id'
        represents channel ID.-->
    </item>
</plateMatchAlarm>
<aoiEntryAlarm type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the area of interest entry alarm. the attribute 'id'
        represents channel ID.-->
    </item>
</aoiEntryAlarm>
<aoiLeaveAlarm type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the area of interest leave alarm. the attribute 'id'
        represents channel ID.-->
    </item>
</aoiLeaveAlarm>
<passlineAlarm type="list" maxCount="" count=""><!--Optional.-->
    <itemType type="boolean"/>
    <item id="1">
        <!--Required.the status of the target counting by line alarm. the attribute 'id'
        represents channel ID.-->
    </item>
</passlineAlarm>
```

```
<trafficAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of the target counting by area alarm. the attribute 'id'
    represents channel ID.-->
  </item>
</trafficAlarm>
<pvdAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of illegal parking alarm. the attribute 'id' represents
    channel ID.-->
  </item>
</pvdAlarm>
<loiteringAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required. The status of loitering alarm. the attribute 'id' represents channel
    ID.-->
  </item>
</loiteringAlarm>
<crowdGatheringAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required. The status of the crowd gathering detection alarm. the attribute
    'id' represents channel ID.-->
  </item>
</crowdGatheringAlarm>
<fireAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required. The status of the fire detection alarm. the attribute 'id' represents
    channel ID.-->
  </item>
```

```
</fireAlarm>
<temperatureAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required. The status of the temperature detection alarm. the attribute 'id'
    represents channel ID.-->
  </item>
</temperatureAlarm>
<soundAbruptUpAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of the audio abnormal alarm. the attribute 'id' represents
    channel ID, If the attribute exists, it indicates an alarm for the channel;
    otherwise, it indicates an alarm for the device.-->
  </item>
</soundAbruptUpAlarm>
<soundAbruptDownAlarm type="list" maxCount="" count=""><!--Optional.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required.the status of the audio abnormal alarm. the attribute 'id' represents
    channel ID, If the attribute exists, it indicates an alarm for the channel;
    otherwise, it indicates an alarm for the device.-->
  </item>
</soundAbruptDownAlarm>
<chlOfflineAlarm type="list" maxCount="" count=""><!--Optional.Only valid for
NVR.-->
  <itemType type="boolean"/>
  <item id="1">
    <!--Required. Channel offline alarm status. the attribute 'id' represents channel
    ID.-->
  </item>
</chlOfflineAlarm>
<chlVideoLoss type="list" maxCount="" count=""><!--Optional.Only valid for NVR.-->
  <itemType type="boolean"/>
  <item id="1">
```

```

        <!--Required. Channel video loss status. the attribute 'id' represents channel
        ID.-->
        </item>
    </chlVideoLoss>
</alarmStatusInfo>
<alarmOutStatusInfo><!--Optional.which alarms are currently being activated.-->
    <audioAlarm type="list" maxCount="" count=""><!--Optional.-->
        <itemType type="boolean"/>
        <item id="1">
            <!--Required. The system is playing alarm voice or not. the attribute 'id'
            represents channel ID(For IPC, it can be omitted, but it is effective for
            NVR).-->
        </item>
    </audioAlarm>
    <lightAlarm type="list" maxCount="" count=""><!--Optional.-->
        <itemType type="boolean"/>
        <item id="1">
            <!--Required. The system is lighting alarm or not. the attribute 'id' represents
            channel ID(For IPC, it can be omitted, but it is effective for NVR).-->
        </item>
    </lightAlarm>
    <sensorAlarmOut type="list" maxCount="" count=""><!--Optional.the status of alarm
    out.-->
        <itemType type="boolean"/>
        <item id="1">
            <!--Required.the status of alarm out. the attribute 'id' represents Alarm out
            ID(uint32), refer to the successful response of GetAlarmOutInfo.-->
        </item>
    </sensorAlarmOut>
</alarmOutStatusInfo>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">

```

```

<alarmStatusInfo>
  <sensorAlarmIn type="list" maxCount="2" count="1">
    <itemType type="boolean"/>
    <item id="1">true</item>
  </sensorAlarmIn>
  <motionAlarm type="list" maxCount="1" count="1">
    <itemType type="boolean"/>
    <item id="1">true</item>
  </motionAlarm>
</alarmStatusInfo>
<alarmOutStatusInfo>
  <audioAlarm type="list" maxCount="1" count="1">
    <itemType type="boolean"/>
    <item>true</item>
  </audioAlarm>
</alarmOutStatusInfo>
</config>

```

[Tips]:

1. The "id" attribute for each item starts from "1".
2. Only report the status information of alarms that are currently occurring.

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

Successful Response	<alarmServer>
Applicable products	IPC,NVR
<p><alarmServer> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <alarmServer><!--Required. Alarm server's configuration.--> <switch type="boolean"><!--Required. boolean,The alarm server is enabled or not.--></switch> <serverAddr type="string" maxLen=""> <!--Required. string,The address of alarm server.--> </serverAddr> <serverPort type="uint16" min="1" max="65535"> <!--Required. uint16,The port of alarm server.--> </serverPort> <url type="string" maxLen=""> <!--Optional. string, URL for sending alarm status.IPC is not supported.--> </url> <enableHeartbeat type="boolean"> <!--Required. boolean, whether to enable heartbeat between the device and the alarm server.--> </enableHeartbeat> <heartbeatInterval type="uint16" min="" max=""> <!--Dependent. uint16, he interval time for sending heartbeats. Unit: seconds. Valid when enableHeartbeat is 'true'.--> </heartbeatInterval> </alarmServer> </config> </pre>	

Example of Successful Response(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmServer>
    <switch type="boolean">false</switch>
    <serverAddr type="string" maxLen="15"><![CDATA[10.10.10.10]]></serverAddr>
    <serverPort type="uint16" min="1" max="65535">8010</serverPort>
    <enableHeartbeat type="boolean">true</enableHeartbeat>
    <heartbeatInterval type="uint16" min="5" max="65535">30</heartbeatInterval>
  </alarmServer>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmServer>
    <switch type="boolean">false</switch>
    <serverAddr type="string" maxLen="15"><![CDATA[10.10.10.10]]></serverAddr>
    <serverPort type="uint16" min="1" max="65535">8010</serverPort>
    <url type="string" maxLen="60"><![CDATA[AlarmStatus]]></url>
    <enableHeartbeat type="boolean">true</enableHeartbeat>
    <heartbeatInterval type="uint16" min="5" max="65535">30</heartbeatInterval>
  </alarmServer>
</config>
```

[Tips]:

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	<alarmServer>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><alarmServer> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <alarmServer><!--Required. Alarm server's configuration.--> <switch><!--Required. boolean,The alarm server is enabled or not.--></switch> <serverAddr><!--Required. string,The address of alarm server.--></serverAddr> <serverPort><!--Required. uint16,The port of alarm server.--></serverPort> <url> <!--Optional. string, URL for sending alarm status.IPC is not supported.--> </url> <enableHeartbeat> <!--Required. boolean, whether to enable heartbeat between the device and the alarm server.--> </enableHeartbeat> <heartbeatInterval> <!--Dependent. uint16, he interval time for sending heartbeats. Unit: seconds. Valid when enableHeartbeat is 'true'.--> </heartbeatInterval> </alarmServer> </config> </pre>	

Example of Entity Data(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmServer>
    <switch>true</switch>
    <serverAddr><![CDATA[10.10.10.10]]> </serverAddr>
    <serverPort>8010</serverPort>
    <enableHeartbeat>true</enableHeartbeat>
    <heartbeatInterval>30</heartbeatInterval>
  </alarmServer>
</config>
```

Example of Entity Data(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <alarmServer>
    <switch>true</switch>
    <serverAddr><![CDATA[10.10.10.10]]> </serverAddr>
    <serverPort>8010</serverPort>
    <url><![CDATA[AlarmStatus]]></url>
    <enableHeartbeat>true</enableHeartbeat>
    <heartbeatInterval>30</heartbeatInterval>
  </alarmServer>
</config>
```

[Tips]:

5.4 AlarmTrigger

5.4.1 GetAlarmTriggerConfig

GetAlarmTriggerConfig

Description	To get the IP media device's trigger configuration of alarms or events.
Typical URL	POST or GET http://<host>[:port]/GetAlarmTriggerConfig[/channelId]</action_name>
Channel ID	Optional. If none channelId included in the URL, the default Channel ID is 1. When "action name" equals "alarmIn", channelId represents the alarmIn ID. In other cases, channelId represents channel ID.
Action name	<p>The action names are defined as follows:</p> <p>alarmIn: alarm linkage of alarmIn.</p> <p>motion: alarm linkage of motion.</p> <p>tripwire: alarm linkage of Tripwire Detection.</p> <p>osc: alarm linkage of Object Status Change.</p> <p>perimeter: alarm linkage of Perimeter Environment Assurance.</p> <p>vfd: alarm linkage of Video Face Detection.</p> <p>vehicle: alarm linkage of Video vehicle Detection.</p> <p>aoientry: alarm linkage of Aoi Entry Detection.</p> <p>aoileave: alarm linkage of Aoi Leave Detection.</p> <p>passlinecount: alarm linkage of Target Counting by Line Detection.</p> <p>traffic: alarm linkage of Target Counting by Area Detection.</p> <p>heatMap: alarm linkage of Heat Map Detection.</p> <p>vsd: alarm linkage of Video Metadata Detection.</p> <p>asd: alarm linkage of Audio Abnormal Detection.</p> <p>avd: alarm linkage of Abnormal Video Detection.</p> <p>pvd: alarm linkage of Illegal Parking Detection.</p> <p>loitering: alarm linkage of Loitering Detection.</p> <p>crowdGathering: alarm linkage of Crowd Gathering Detection.</p> <p>temperatureAlarm: alarm linkage of Thermal imaging temperature measurement.</p> <p>fire:alarm linkage of fire.</p>
Entity Data	None
Successful Response	< triggerConfig >
Applicable products	IPC,NVR

<triggerConfig> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <triggerConfig><!--Required;alarm trigger configuration.-->
    <snap type="list" maxCount="" count=""><!--Optional;snap trigger configuration.-->
      <item>
        <channelId type="uint32">
          <!--Optional; For NVR:capture a picture for the specified channelId;IPC is not supported.-->
        </channelId>
        <switch type="boolean"><!--Required;snap enabled or not.--></switch>
      </item>
    </snap>
    <record type="list" maxCount="" count=""><!--Optional;record trigger configuration.-->
      <item>
        <channelId type="uint32">
          <!--Optional; For NVR:record for the specified channelId; IPC is not supported.-->
        </channelId>
        <switch type="boolean"><!--Required;record enabled or not.--></switch>
      </item>
    </record>
    <alarmOut type="list" maxCount="" count=""><!--Optional;alarm out trigger configuration.-->
      <item>
        <alarmOutId type="uint32"><!--Required;alarm out ID.--></alarmOutId>
        <switch type="boolean"><!--Required;alarm out enabled or not.--></switch>
      </item>
    </alarmOut>
    <audio type="list" maxCount="" count=""><!--Optional;audio alarm configuration.NVR is not supported.-->
      <item>
        <switch type="boolean"><!--Required;enabled or not.--></switch>
        <audioType type="audioAlarmType">
          <!--Optional;The ID of the alarm voice file,refer to the element <audioAlarmType> in the successful response of "GetAudioAlarmOutConfig". If this node does not exist, it means using the default voice (the voice file configured in the successful response of
```

```

        GetAudioAlarmOutConfig).-->
        </audioType>
    </item>
</audio>
<light type="list" maxCount="" count=""><!--Optional;light alarm configuration. NVR is not
supported.-->
    <item>
        <switch type="boolean"><!--Required;light alarm enabled or not.--></switch>
    </item>
</light>
<autoTrack type="list" maxCount="" count=""><!--Optional;auto track configuration.NVR is not
supported.-->
    <item>
        <switch type="boolean"><!--Required;light alarm enabled or not.--></switch>
    </item>
</autoTrack>
</triggerConfig>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <triggerConfig>
        <snap type="list" maxCount="1" count="1">
            <item>
                <channelId type="uint32">1</channelId>
                <switch type="boolean">true</switch>
            </item>
        </snap>
        <record type="list" maxCount="1" count="1">
            <item>
                <channelId type="uint32">1</channelId>
                <switch type="boolean">true</switch>
            </item>

```

```

</record>
<alarmOut type="list" maxCount="1" count="1">
  <item>
    <alarmOutId type="uint32">0</alarmOutId>
    <switch type="boolean">true</switch>
  </item>
</alarmOut>
<audio type="list" maxCount="1" count="1">
  <item>
    <switch type="boolean">true</switch>
    <audioType type="audioAlarmType">10</audioType>
  </item>
</audio>
<light type="list" maxCount="1" count="1">
  <item>
    <switch type="boolean">false</switch>
  </item>
</light>
</triggerConfig>
</config>

```

[Tips]:

1. For "action name", NVR only supports "alarmIn".

5.4.2 SetAlarmTriggerConfig

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

Action name	The same as "GetAlarmTriggerConfig".
Entity Data	<triggerConfig>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<triggerConfig> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <triggerConfig><!--Required;alarm trigger configuration.-->
    <snap><!--Optional;snap trigger configuration.-->
      <item>
        <channelId>
          <!--Optional; uint32,For NVR:capture a picture for the specified channelId;IPC is
            not supported.-->
        </channelId>
        <switch><!--Required; boolean, snap enabled or not.--></switch>
      </item>
    </snap>
    <record><!--Optional;record trigger configuration.-->
      <item>
        <channelId>
          <!--Optional; uint32,For NVR:record for the specified channelId; IPC is not
            supported.-->
        </channelId>
        <switch><!--Required;boolean,record enabled or not.--></switch>
      </item>
    </record>
    <alarmOut><!--Optional;alarm out trigger configuration.-->
      <item>
        <alarmOutId><!--Required;uint32,alarm out ID.--></alarmOutId>
        <switch><!--Required;boolean,alarm out enabled or not.--></switch>
      </item>
    </alarmOut>

```

```

<audio><!--Optional;audio alarm configuration.NVR is not supported.-->
  <item>
    <switch><!--Required;boolean, enabled or not.--></switch>
    <audioType>
      <!--Optional;audioAlarmType,The ID of the alarm voice file,refer to the element
      <audioAlarmType> in the successful response of "GetAudioAlarmOutConfig". If
      this node does not exist, it means using the default voice (the voice file
      configured in the successful response of GetAudioAlarmOutConfig).-->
    </audioType>
  </item>
</audio>
<light><!--Optional;light alarm configuration. NVR is not supported.-->
  <item>
    <switch><!--Required;boolean,light alarm enabled or not.--></switch>
  </item>
</light>
<autoTrack><!--Optional;auto track configuration. NVR is not supported.s.-->
  <item>
    <switch><!--Required;boolean,light alarm enabled or not.--></switch>
  </item>
</autoTrack>
</triggerConfig>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <triggerConfig>
    <snap>
      <item>
        <channelId>1</channelId>
        <switch>true</switch>
      </item>
    </snap>

```

```
<record>
  <item>
    <channelId>1</channelId>
    <switch>true</switch>
  </item>
</record>
<alarmOut>
  <item>
    <alarmOutId>0</alarmOutId>
    <switch>true</switch>
  </item>
</alarmOut>
<audio>
  <item>
    <switch>true</switch>
    <audioType>10</audioType>
  </item>
</audio>
<light>
  <item>
    <switch>>false</switch>
  </item>
</light>
</triggerConfig>
</config>
```

[Tips]:

5.5 Sound-Light Alarm

5.5.1 GetAudioAlarmOutConfig

GetAudioAlarmOutConfig	
Description	To get the IP media device's audio alarm configuration.
Typical URL	POST or GET http://<host>[:port]/GetAudioAlarmOutConfig[/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	<audioAlarmOut>
Applicable products	IPC
<p><audioAlarmOut> XML Block:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <audioAlarmType><!--Optional; List of alarm voice.--> <enum language="en-us" value="Warning area, leave as soon as possible">1</enum> <!--Optional; The attribute 'language' represents the language of the voice file(Refer to the definition of <audioLanguageType>.), the attribute 'value' represents the content of the voice file, and the value represents the identifier of the voice file. The system may support multiple languages and files, and there may be differences between products. Please refer to the actual content carried in the successful response.--> </audioAlarmType> <audioLanguageType><!--Optional; Language list of voice files.--> <enum value="English">en-us</enum><!--Optional; The attribute 'value' represents the description of language, and the value represents the identifier of the language. There may be multiple languages, Please refer to the actual content carried in the successful response. Specifically, "customize" means to customize a voice file without any attributes. --> </audioLanguageType> </types> <audioAlarmOut><!--Required;audio alarm configurations.--></pre>	

```

<switch type="boolean"><!--Required;audio alarm enabled or not.--></switch>
<languageType type="audioLanguageType">
  <!--Optional; the language of voice file.-->
</languageType>
<audioType type="audioAlarmType"><!--Required;voice file ID.--></audioType>
<alarmTimes type="uint32" min="" max="" default="">
  <!--Required; uint32,the number of times to listen to the voice.-->
</alarmTimes>
<audioVolume type="uint32" min="" max="" default="">
  <!--Required; uint32, volume of voice playback.-->
</audioVolume>
</audioAlarmOut>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <audioAlarmType>
      <enum language="en-us" value="Warning area, leave as soon as possible">1</enum>
      <enum language="en-us" value="Dangerous area, please do not approach">2</enum>
      <enum language="en-us" value="No parking in this area">3</enum>
      <enum language="en-us" value="You have entered the real-time monitoring
        area">4</enum>
      <enum language="en-us" value="Hello, welcome">5</enum>
      <enum language="en-us" value="Do not touch valuables">6</enum>
      <enum language="en-us" value="Private area, no entry">7</enum>
      <enum language="en-us" value="Danger of water depth, pay attention to
        safety">8</enum>
      <enum language="en-us" value="High altitude, don't climb">9</enum>
      <enum language="en-us" value="Howling alarm sound">10</enum>
      <enum language="customize" value="My customize voice">100</enum>
    </audioAlarmType>
  </types>
  <audioLanguageType>

```

```

        <enum value="English">en-us</enum>
        <enum>customize</enum>
    </audioLanguageType>
</types>
<audioAlarmOut>
    <switch type="boolean">>true</switch>
    <languageType type="audioLanguageType">zh-cn</languageType>
    <audioType type="audioAlarmType">10</audioType>
    <alarmTimes type="uint32" min="1" max="50" default="5">5</alarmTimes>
    <audioVolume type="uint32" min="0" max="100" default="100">75</audioVolume>
</audioAlarmOut>
</config>

```

[Tips]:

1. Requirements/limitations for custom voice files: The maximum number of custom voice files is 10; The file format requirement is WAV format; The sampling rate is 8000HZ; The sound channel is monophonic, 16 bits, and the size of a single file does not exceed 200KB.

5.5.2 SetAudioAlarmOutConfig

SetAudioAlarmOutConfig	
Description	To set the IP media device's audio alarm configuration.
Typical URL	POST http://<host>[:port]/SetAudioAlarmOutConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<audioAlarmOut>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><audioAlarmOut> XML Block:</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="" xmlns="http://www.ipc.com/ver10"> </pre>	

<pre> <audioAlarmOut><!--Required;audio alarm configurations.--> <switch><!--Optional;boolean,audio alarm enabled or not.--></switch> <audioType><!--Optional; audioAlarmType,voice file ID.--></audioType> <alarmTimes> <!--Optional; uint32,the number of times to listen to the voice.--> </alarmTimes> <audioVolume> <!--Optional; uint32, volume of voice playback.--> </audioVolume> </audioAlarmOut> </config> </pre>
<p>Example of Entity Data:</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <audioAlarmOut> <switch>true</switch> <audioType>10</audioType> <alarmTimes>5</alarmTimes> <audioVolume>75</audioVolume> </audioAlarmOut> </config> </pre>
<p>[Tips]:</p>

5.5.3 AddCustomizeAudioAlarm

AddCustomizeAudioAlarm	
Description	Add custom alarm audio.
Typical URL	POST http://<host>[:port]/AddCustomizeAudioAlarm[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1

Action name	None
Entity Data	<addAudioAlarm>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC

<addAudioAlarm> of Entity Data XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <addAudioAlarm><!--Required;Information about the alarm voice file.-->
    <audioName>
      <!--Required;string, name of the alarm voice file.-->
    </audioName>
    <audioFileSize>
      <!--Required;uint32, the length of audioFileData (after base64 encoding).-->
    </audioFileSize>
    <audioFileData>
      <!--Required;string, base64 encoded alarm voice file content.-->
    </audioFileData>
  </addAudioAlarm>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <addAudioAlarm>
    <audioName><![CDATA[welcome]]> </audioName>
    <audioFileSize>123</audioFileSize>
    <audioFileData><![CDATA[...base64encodeData...]]> </audioFileData>
  </addAudioAlarm>
</config>
```

[Tips]:

5.5.4 DeleteCustomizeAudioAlarm

DeleteCustomizeAudioAlarm	
Description	Delete custom alarm audio
Typical URL	POST http://<host>[:port]/DeleteCustomizeAudioAlarm[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1
Action name	None
Entity Data	<deleteAudioAlarm>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><deleteAudioAlarm>XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <deleteAudioAlarm><!--Required;Voice file information that needs to be deleted.--> <id><!--Required; audioAlarmType, Customize voice file ID. Value range: The enumeration value of <audioAlarmType> defined in the successful response of GetAudioAlarmOutConfig(Select the voice ID with language attribute="customize").--> </id> </deleteAudioAlarm> </config></pre>	
<p>Example of Entity Data:</p> <pre><?xml version="1.0" encoding="utf-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <deleteAudioAlarm> <id>101</id> </deleteAudioAlarm> </config></pre>	
<p>[Tips]:</p> <p>1. The 'id' is the id in the element <customize> returned by "AddCustomizeAudioAlarm", or the element <customize> returned by "GetAudioAlarmOutConfig".</p>	

5.5.5 AuditionAudioAlarm

AuditionAudioAlarm	
Description	Request the system to play the specified voice file.
Typical URL	POST http://<host>[:port]/AuditionAudioAlarm[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1
Action name	None
Entity Data	<auditionAudioAlarm>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><auditionAudioAlarm> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <auditionAudioAlarm><!--Required;Voice file information that needs to be played.--> <audioType> <!--Required; audioAlarmType, system voice file ID. --> </audioType> </auditionAudioAlarm> </config></pre>	
<p>Example of Entity Data(Listen to system voice files):</p> <pre><?xml version="1.0" encoding="utf-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <auditionAudioAlarm> <audioType>2</audioType> </auditionAudioAlarm> </config></pre>	
<p>Example of Entity Data(Listen to customize voice files):</p>	

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <auditionAudioAlarm>
    <audioType>100</audioType>
  </auditionAudioAlarm>
</config>

```

[Tips]:

5.5.6 GetLightAlarmOutConfig

GetLightAlarmOutConfig	
Description	To get the configuration of light alarm
Typical URL	POST or GET http://<host>[:port]/GetLightAlarmOutConfig[/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	<lightAlarmOut>
Applicable products	IPC
<p><lightAlarmOut>XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <lightFrequency><!--Required; The flicker frequency of the light.--> <enum>low</enum> <enum>medium</enum> <enum>high</enum> </lightFrequency> </types> <lightAlarmOut> <switch type="boolean"><!--Required; Light alarm enabled or not.--></switch> </pre>	

```

<durationTime type="uint32" min="" max="" default="">
    <!--Dependent; The duration of flashing lights. Valid when switch is 'true'.-->
</durationTime>
<frequency type="lightFrequency">
    <!--Dependent; The flicker frequency of the light. Valid when switch is 'true'. -->
</frequency>
</lightAlarmOut>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <lightFrequency>
            <enum>low</enum>
            <enum>medium</enum>
            <enum>high</enum>
        </lightFrequency>
    </types>
    <lightAlarmOut>
        <switch type="boolean">false</switch>
        <durationTime type="uint32" min="1" max="60" default="20">20</durationTime>
        <frequency type="lightFrequency">low</frequency>
    </lightAlarmOut>
</config>

```

[Tips]:

5.5.7 SetLightAlarmOutConfig

SetLightAlarmOutConfig	
Description	To set the configuration of light alarm

Typical URL	POST http://<host>[:port]/SetLightAlarmOutConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1
Action name	None
Entity Data	<lightAlarmOut>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><lightAlarmOut>XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <lightAlarmOut> <switch><!--Required; Light alarm enabled or not.--></switch> <durationTime> <!--Dependent; The duration of flashing lights. Valid when switch is 'true'.--> </durationTime> <frequency> <!--Dependent; The flicker frequency of the light. Valid when switch is 'true'. --> </frequency> </lightAlarmOut> </config> </pre>	
<p>Example of Entity Data(Enable light alarm):</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <lightAlarmOut> <switch>true</switch> <durationTime>20</durationTime> <frequency>low</frequency> </lightAlarmOut> </config> </pre>	
<p>Example of Entity Data(Disable light alarm):</p>	

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <lightAlarmOut>
    <switch>>false</switch>
  </lightAlarmOut>
</config>

```

[Tips]:

5.6 Disarming

5.6.1 RequestDisarming

RequestDisarming	
Description	To execute system disarming. After the system is disarmed, the related linkage alarm items will no longer be effective.
Typical URL	POST http://<host>[:port]/RequestDisarming
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
[Tips]:	

5.6.2 CancelDisarming

CancelDisarming	
Description	To cancel system disarming.

Typical URL	POST http://<host>[:port]/CancelDisarming
Channel ID	None
Action name	None
Entity Data	None
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
[Tips]:	

5.6.3 GetDisarmingStatus

GetDisarmingStatus	
Description	To get system disarming status.
Typical URL	POST or GET http://<host>[:port]/GetDisarmingStatus
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<disarmingStatus>
Applicable products	IPC,NVR
<p><disarmingStatus> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <disarmingStatus><!--Required;the status of disarming.--> <status type="boolean"><!--Required; Disarming is enabled or not.--></status> </disarmingStatus> </config> </pre>	
Example of Successful Response:	

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <disarmingStatus>
    <status type="boolean">true</status>
  </disarmingStatus>
</config>

```

[Tips]:

6 Playback

6.1 Record Search

6.1.1 GetRecordType

GetRecordType	
Description	To get the video recording types supported by the device.
Typical URL	POST or GET http://<host>[:port]/GetRecordType
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<recTypeCaps>
Applicable products	IPC,NVR

<recTypeCaps> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <recType><!--Required;Define the type of record.-->
      <enum>manual</enum><!--Optional; Manual recording.-->
      <enum>schedule</enum><!--Optional; Schedule recording.-->
      <enum>motion</enum><!--Optional; Motion recording.-->
      <enum>sensor</enum><!--Optional; Alarm in linkage recording.-->
      <enum>intelligentDetection</enum><!--Optional; AI recording.-->
      <enum>networkBroken</enum><!--Optional; Network abnormality recording.Only
      valid for IPC.-->
    </recType>
  </types>
  <recTypeCaps type="list" maxCount="" count=""><!--Required;List of record type.-->
    <itemType type="recType" maxLen=""/>
    <item><!--Required;video recording type item.--></item>
  </recTypeCaps>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <recType>
      <enum>manual</enum>
      <enum>schedule</enum>
      <enum>motion</enum>
      <enum>sensor</enum>
      <enum>intelligentDetection</enum>
    </recType>
  </types>
  <recTypeCaps type="list" maxCount="32" count="5">
    <itemType type="recType" maxLen="20"/>
  </recTypeCaps>
</config>
```

```
<item>manual</item>
<item>schedule</item>
<item>motion</item>
<item>sensor</item>
<item>intelligentDetection</item>
</recTypeCaps>
</config>
```

Example of Successful Response(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <recType>
      <enum>manual</enum>
      <enum>schedule</enum>
      <enum>motion</enum>
      <enum>sensor</enum>
      <enum>intelligentDetection</enum>
      <enum>networkBroken</enum>
    </recType>
  </types>
  <recTypeCaps type="list" maxCount="32" count="6">
    <itemType type="recType" maxLen="20"/>
    <item>manual</item>
    <item>schedule</item>
    <item>motion</item>
    <item>sensor</item>
    <item>intelligentDetection</item>
    <item>networkBroken</item>
  </recTypeCaps>
</config>
```

[Tips]:

6.1.2 SearchRecordDate

SearchRecordDate	
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
Successful Response	<dateList>
Applicable products	IPC,NVR
<dateList> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <dateList type="list" maxCount="" count=""><!--Required;List of recorded dates.--> <itemType type="string"/> <item><!--Required;Date,format:YYYY-MM-DD.--></item> </dateList> </config>	

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <dateList type="list" maxCount="500" count="6">
    <itemType type="string"/>
    <item><![CDATA[2014-01-09]]</item>
    <item><![CDATA[2014-02-09]]</item>
    <item><![CDATA[2014-03-08]]</item>
    <item><![CDATA[2014-04-02]]</item>
    <item><![CDATA[2014-04-03]]</item>
    <item><![CDATA[2014-04-04]]</item>
  </dateList>
</config>
```

[Tips]:

6.1.3 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	<search>
Successful Response	<timesectionList>
Applicable products	IPC,NVR
<search> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <search><!--Required; search criteria.--> <recTypes><!--Required; List of video types that need to be searched.--></pre>	

```

        <item><!--Required;video recording type item, the definition refers to the successful
            response of calling "GetRecordType".-->

        </item>
    </recTypes>
    <starttime>
        <!--Required;string, the start time of the recording.format:YYYY-MM-DD HH:MM:SS.-->
    </starttime>
    <endtime>
        <!--Required;string, the end time of the recording.format:YYYY-MM-DD HH:MM:SS.-->
    </endtime>
</search>
</config>

```

<timesectionList> XML Block:

```

<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <timesectionList type="list" maxCount="" count=""><!--Required; List of recorded time slots.-->
        <item>
            <starttime seconds="" recType="">
                <!--Required;string, the start time of the recording, with the attribute "seconds"
                    indicating the duration of the recording (in seconds) and the attribute "recType"
                    indicating the type of recording,the definition refers to the successful response of
                    calling "GetRecordType".format:YYYY-MM-DD HH:MM:SS.-->
            </starttime>
            <endtime>
                <!--Required;string, the end time of the recording.format:YYYY-MM-DD
                    HH:MM:SS.-->
            </endtime>
        </item>
    </timesectionList>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <search>

```

```
<recTypes>
  <item>manual</item>
  <item>schedule</item>
  <item>motion</item>
  <item>sensor</item>
  <item>intelligentDetection</item>
</recTypes>
<starttime><![CDATA[2024-09-30 00:00:00]]></starttime>
<endtime><![CDATA[2024-10-30 23:59:59]]></endtime>
</search>
```

```
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <timesectionList maxCount="500" count="2">
    <item>
      <starttime type="string" seconds="827" recType="schedule">
        <![CDATA[2024-09-30 07:39:36]]>
      </starttime>
      <endtime type="string">
        <![CDATA[2024-09-30 07:53:36]]>
      </endtime>
    </item>
    <item>
      <starttime type="string" seconds="533" recType="motion">
        <![CDATA[2024-09-30 07:54:03]]>
      </starttime>
      <endtime type="string">
        <![CDATA[2024-09-30 07:53:36]]>
      </endtime>
    </item>
```

```

    </timesectionList>
</config>

```

[Tips]:

6.2 RecordStatus

6.2.1 GetRecordStatusInfo

GetRecordStatusInfo	
Description	To get the record status.
Typical URL	POST or GET http://<host>[:port]/GetRecordStatusInfo
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<recordStatusList>
Applicable products	IPC,NVR
<p><recordStatusList> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <recordStatusType><!--Required; Define the status of record types.--> <enum>no recording</enum> <enum>recording</enum> <enum>exception</enum> </recordStatusType> </types> <recordStatusList type="list" maxCount="" count=""><!--Required; List of recording status for each channel.--> <itemType type="recordStatusType"/> <item id="" streamType="" resolution="" frameRate="" bitrateType="" imageQuality="" </pre>	

```
maxBitrate="" recordTypes="">

<!--Required; The attribute ID represents the channel ID, and the value represents the
recording status of the channel. Other attributes are optional, please refer to "Tips" for
details.-->

</item>
</recordStatusList>
</config>
```

Example of Successful Response(IPC):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <recordStatusType>
      <enum>no recording</enum>
      <enum>recording</enum>
      <enum>exception</enum>
    </recordStatusType>
  </types>
  <recordStatusList type="list" maxCount="1" count="1">
    <itemType type="recordStatusType"/>
    <item id="1">recording</item>
  </recordStatusList>
</config>
```

Example of Successful Response(NVR):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <recordStatusType>
      <enum>no recording</enum>
      <enum>recording</enum>
      <enum>exception</enum>
    </recordStatusType>
  </types>
```

```
<recordStatusList type="list" maxCount="256" count="1">
  <itemType type="recordStatusType"/>
  <item id="1" streamType="main" resolution="2592x1520" frameRate="30" bitrateType="VBR"
    imageQuality="higher" maxBitrate="3072" recordTypes="motion">recording</item>
</recordStatusList>
</config>
```

[Tips]:

Attribute Description:

1. "streamType": string, value in {main, sub}. main represents the main stream, and sub represents the sub stream.
2. "encodeType": string, video encoding type. Value in {H.264, 264Smart, H.264Plus, H.265, H.265Smart, H.265Plus}.
3. "bitrateType": string, value in {VBR, CBR}. The type of bit rate.
4. "audioSwitch": string, value in {on, off}, whether the audio is recorded.
5. "imageQuality": string, video quality, value in {lowest, lower, medium, higher, highest}.
6. "maxBitrate": uint32, maximum bit rate,.
7. "recordTypes": Recording types. Refer to the element<recType>in the successful response of "GetRecordType".

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	<tcpIp>
Applicable products	IPC,NVR
<tcpIp> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <ipSettingMode><!--Required.the type of ip.--> <enum>staticIp</enum> <enum>dhcp</enum> </ipSettingMode> </types> <tcpIp><!--Required.IP information of device.--> <ipSettingMode type="ipSettingMode"><!--Required.DHCP or static IP.--> </ipSettingMode>	

```

<staticIp type="string" minLength="" maxLength="">
    <!--Dependent;ip address, valid when ipSettingMode='staticIp'.-->
</staticIp>
<staticIpRoute type="string" minLength="" maxLength="">
    <!--Dependent;ip gateway address, valid when ipSettingMode='staticIp'.-->
</staticIpRoute>
<staticIpMask type="string" minLength="" maxLength="">
    <!--Dependent;ip address mask, valid when ipSettingMode='staticIp'.-->
</staticIpMask>
<dhcpIp type="string" minLength="" maxLength="">
    <!--Dependent;ip address, valid when ipSettingMode='dhcp'.-->
</dhcpIp>
<dhcpIpRoute type="string" minLength="" maxLength="">
    <!--Dependent;ip gateway address, valid when ipSettingMode='dhcp'.-->
</dhcpIpRoute>
<dhcpIpMask type="string" minLength="" maxLength="">
    <!--Dependent;ip address mask, valid when ipSettingMode='dhcp'.-->
</dhcpIpMask>
<dnsFromDhcpSwitch type="boolean">
    <!--Dependent; Valid when ipSettingMode='dhcp'.automatically obtain DNS or not.-->
</dnsFromDhcpSwitch>
<dnsServer1 type="string" minLength="" maxLength="">
    <!--Dependent; Valid when ipSettingMode='staticIp' or ipSettingMode='dhcp'&&
    dnsFromDhcpSwitch='false',preferred DNS server address.-->
</dnsServer1>
<dnsServer2 type="string" minLength="" maxLength="">
    <!--Dependent; Valid when ipSettingMode='staticIp' or ipSettingMode='dhcp'&&
    dnsFromDhcpSwitch='false',preferred DNS server address.-->
</dnsServer2>
</tcpIp>
</config>

```

Example of Successful Response:

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

```
<config version="2.0.0" 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" minLength="7" maxLength="15">
      <![CDATA[10.10.10.123]]>
    </staticIp>
    <staticIpRoute type="string" minLength="7" maxLength="15">
      <![CDATA[10.10.10.253]]>
    </staticIpRoute>
    <staticIpMask type="string" minLength="7" maxLength="15">
      <![CDATA[255.255.255.0]]>
    </staticIpMask>
    <dnsServer1 type="string" minLength="7" maxLength="15">
      <![CDATA[192.168.226.1]]>
    </dnsServer1>
    <dnsServer2 type="string" minLength="7" maxLength="15">
      <![CDATA[8.8.8.8]]>
    </dnsServer2>
  </tcpIp>
</config>
```

[Tips]:

1. For NVR, only the first network port is supported.

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	<tcpIp>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<tcpIp> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <tcpIp><!--Required.IP information of device.-->
    <ipSettingMode><!--Required.DHCP or static IP.--> </ipSettingMode>
    <staticIp>
      <!--Dependent;ip address, valid when ipSettingMode='staticIp'.-->
    </staticIp>
    <staticIpRoute>
      <!--Dependent;ip gateway address, valid when ipSettingMode='staticIp',Optional.-->
    </staticIpRoute>
    <staticIpMask>
      <!--Dependent;ip address mask, valid when ipSettingMode='staticIp'.-->
    </staticIpMask>
    <dnsFromDhcpSwitch>
      <!--Dependent; Valid when ipSettingMode='dhcp'.automatically obtain DNS or not.-->
    </dnsFromDhcpSwitch>
    <dnsServer1>
      <!--Dependent; Valid when ipSettingMode='staticIp' or ipSettingMode='dhcp'&&
      dnsFromDhcpSwitch='false',preferred DNS server address.-->
    </dnsServer1>
    <dnsServer2>
      <!--Dependent; Valid when ipSettingMode='staticIp' or ipSettingMode='dhcp'&&
      dnsFromDhcpSwitch='false',backup DNS server address.-->
  </tcpIp>
</config>

```

```
</dnsServer2>
</tcpIp>
</config>
```

Example of Entity Data(static ip):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <tcpIp>
    <ipSettingMode>staticIp</ipSettingMode>
    <staticIp>
      <![CDATA[10.10.10.123]]>
    </staticIp>
    <staticIpRoute>
      <![CDATA[10.10.10.253]]>
    </staticIpRoute>
    <staticIpMask>
      <![CDATA[255.255.255.0]]>
    </staticIpMask>
    <dnsServer1 type="string" minLength="7" maxLength="15">
      <![CDATA[192.168.226.1]]>
    </dnsServer1>
    <dnsServer2 type="string" minLength="7" maxLength="15">
      <![CDATA[8.8.8.8]]>
    </dnsServer2>
  </tcpIp>
</config>
```

Example of Entity Data(dhcp):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <tcpIp>
    <ipSettingMode>dhcp</ipSettingMode>
    <dnsFromDhcpSwitch>true</dnsFromDhcpSwitch>
```

```
</tcpIp>
</config>
```

[Tips]:

1. For NVR, only the first network port is supported.

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	<pppoe>
Applicable products	IPC,NVR
<pppoe> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <pppoe><!--Required.Information of pppoe.--> <switch type="boolean"><!--Required. PPPOE function is enabled or not.--></switch> <userName type="string" maxLen=""> <!--Required. PPPOE user name.--> </userName> </pppoe> </config>	

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <pppoe>
    <switch type="boolean">false</switch>
    <userName type="string" maxLen="">
      <![CDATA[test]]>
    </userName>
  </pppoe>
</config>
```

[Tips]:

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	<pppoe>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<pppoe> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <pppoe><!--Required.Information of pppoe.-->
    <switch><!--Optional;boolean,PPPOE function is enabled or not.--></switch>
    <userName>
      <!--Optional. string, PPPOE user name.-->
    </userName>
    <password>
      <!--Optional. string,PPPOE password, Base64 encoding of password.-->
    </password>
  </pppoe>
</config>
```

Example of Entity Data(static ip):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <pppoe>
    <switch>>false</switch>
    <userName><![CDATA[test]]></userName>
    <password><![CDATA[YWFh]]></password>
  </pppoe>
</config>
```

[Tips]:

1. The "password" elements are all "string" type with maxlen"16". They should be encoded by base64, the "YWFh" is the encoded result for "aaa".

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	<port>
Applicable products	IPC,NVR
<p><port> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <port><!--Required;Port information of the device.--> <httpPort type="uint16" min="" max=""><!--Required; Http service port.--></httpPort> <httpsPort type="uint16" min="" max=""><!--Required;Https service port.--></httpsPort> <netPort type="uint16" min="" max=""><!--Required; Private binary service port.--></netPort> <rtspPort type="uint16" min="" max=""><!--Required; Rtsp service port.--></rtspPort> <rtspOverTLSPort type="uint16" min="" max=""> <!--Optional;Rtsp over TLS service port.NVR is not supported.--> </rtspOverTLSPort> </port> </config></pre>	
<p>Example of Successful Response:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <port> <httpPort type="uint16" min="2" max="65535">80</httpPort> <httpsPort type="uint16" min="2" max="65535">443</httpsPort> <netPort type="uint16" min="1" max="65535">9008</netPort> <rtspPort type="uint16" min="1" max="65535">554</rtspPort> <rtspOverTLSPort type="uint16" min="1" max="65535">332</rtspOverTLSPort> </port> </config></pre>	

[Tips]:

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	<port>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<port> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <port><!--Required;Port information of the device.--> <httpPort><!--Required; Http service port.--></httpPort> <httpsPort><!--Required;Https service port.--></httpsPort> <netPort><!--Required; Private binary service port.--></netPort> <rtspPort><!--Required; Rtsp service port.--></rtspPort> <rtspOverTLSPort> <!--Optional;Rtsp over TLS service port. NVR is not supported.--> </rtspOverTLSPort> </port> </config>	

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <port>
    <httpPort>80</httpPort>
    <netPort>9008</netPort>
    <rtspPort>554</rtspPort>
    <httpsPort>443</httpsPort>
    <rtspOverTLSPort>332</rtspOverTLSPort>
  </port>
</config>
```

[Tips]:

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	<ddns>
Applicable products	IPC,NVR
<ddns> XML Block:	
<config version="" xmlns="http://www.ipc.com/ver10">	

```

<types>
  <ddnsServerType>
    <!--Required;Define DDNS type list. The attribute "requestParameters" indicate which
    fields are valid in the<ddns> block. The supported DDNS types may vary depending on the
    product or software version, please refer to the successful response returned by
    'GetDdnsConfig'. The attribute 'domainSuffix' represents the suffix of<domainName>.
    When calling GetDdnsConfig or SetDdnsConfig,<domainName> does not have the content
    with this suffix.-->
    <enum requireParameters="userName,password,domainName,heartbeatTime,serverName "
    domainSuffix=".autoddns.com">xxx</enum>
  </types>
<ddns>
  <switch type="boolean"><!--Required;DDNS service enable or not.--></switch>
  <servertype type="ddnsServerType">
    <!--Dependent;DDNS service type. Valid when switch is 'true'.-->
  </servertype>
  <userName type="string" maxLen="">
    <!--Optional;string,user name.The 'requireParameters' defined in ddnsServerType specifies
    whether this field is valid.-->
  </userName>
  <domainName type="string" maxLen="">
    <!--Optional;string,domain name. The 'requireParameters' defined in ddnsServerType
    specifies whether this field is valid.-->
  </domainName>
  <serverName type="string" maxLen="">
    <!--Optional;string,server name of DDNS. The 'requireParameters' defined in
    ddnsServerType specifies whether this field is valid.-->
  </serverName>
  <heartbeatTime type="uint32">
    <!--Optional;uint32,Range[5,3600],heartbeat time of DDNS. The 'requireParameters'
    defined in ddnsServerType specifies whether this field is valid.-->
  </heartbeatTime>
</ddns>
</config>

```

Example of Successful Response:

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

```

<config version="2.0.0" 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">dvrddns</enum>
      <enum requireParameters="domainName" domainSuffix=".autoddns.com">
        www.autoddns.com</enum>
    </ddnsServerType>
  </types>
  <ddns>
    <switch type="boolean">true</switch>
    <servertype type="ddnsServerType">www.autoddns.com</servertype>
    <domainName type="string" maxLen="63"><![CDATA[11]]></domainName>
  </ddns>
</config>

```

[Tips]:

7.4.2 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	<ddns>
Successful Response	The standard successful result response that described in 1.3.5.

Applicable products

IPC,NVR

<ddns> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <ddns>
    <switch><!--Required;DDNS service enable or not.--></switch>
    <servertype>
      <!--Dependent; ddnsServerType, DDNS service type. Valid when switch is 'true'.-->
    </servertype>
    <userName>
      <!--Optional;string,user name.The 'requireParameters' defined in ddnsServerType specifies whether this field is valid.-->
    </userName>
    <password><!--Optional. string, Base64 encoding of password.--></password>
    <domainName>
      <!--Optional;string,domain name. The 'requireParameters' defined in ddnsServerType specifies whether this field is valid.-->
    </domainName>
    <serverName>
      <!--Optional;string,server name of DDNS. The 'requireParameters' defined in ddnsServerType specifies whether this field is valid.-->
    </serverName>
    <heartbeatTime>
      <!--Optional;uint32,Range[5,3600],heartbeat time of DDNS. The 'requireParameters' defined in ddnsServerType specifies whether this field is valid.-->
    </heartbeatTime>
  </ddns>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <ddns>
    <switch>true</switch>
    <servertype>www.autoddns.com</servertype>
```

```
<domainName><![CDATA[11]]></domainName>
</ddns>
</config>
```

[Tips]:

1. The "password" elements are all "string" type with maxLen"16". They should be encoded by base64, the "YWFh" is the encoded result for "aaa".

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
Entity Data	<userPassword>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<userPassword>XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <userPassword><!--Required.-->
    <oldPassword><!--Required;string, Base64 encoding of old password.--></oldPassword>
    <password><!--Required;string, Base64 encoding of new password.--></password>
  </userPassword>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" 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"16". They should be encoded by base64, the "YWFh" and "YmJi" are the encoded result for "aaa" and "bbb".

8.2 Reboot

8.2.1 Reboot

Reboot	
Description	To reboot the IP media device.
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.
Applicable products	IPC,NVR

9 Talkback commands

9.1 Talkback

9.1.1 profile_talk

profile_talk	
Description	Get the url that can used to send and receive the two-way audio data after the intercom opened.
Typical URL	POST or GET http://<host>[:port]/profile_talk
Channel ID	None
Action name	None
Entity Data	None
Successful Response	<URL>
Applicable products	IPC,NVR

<URL> XML Block:

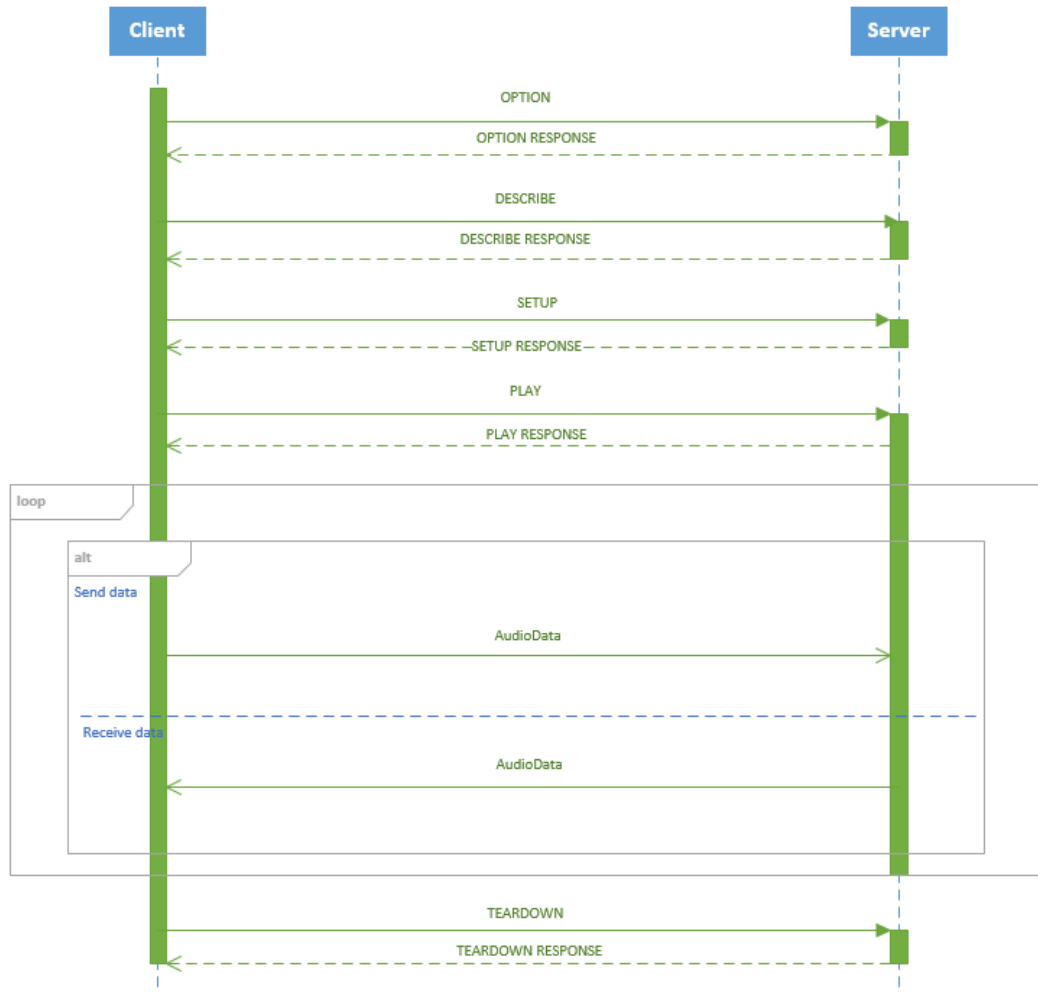
```
<config version="" xmlns="http://www.ipc.com/ver10">  
  <URL type="string"><!--Required.URL for audio intercom.--></URL>  
</config>
```

Example of Successful Response:

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">  
  <URL type="string"><![CDATA[rtsp://192.168.0.9:554/profile_talk]]></URL>  
</config>
```

[Tips]:

1. When the URL invoked by the client application, the two-way audio data stream can be passed through the RTSP protocol as below:



2. The RTSP error code is defined as below:

600	The device is busy
601	Audio open failed
602	No permission

3. Get the format of the tow-way data from rtp payload. And send the same format to device. It supports only single channel. The sampling rate is 8000HZ. The RTP size is a multiple of 320 bytes. Maximum of 320*5.

9.1.2 channel_talk

channel_talk

Description	Get the URL used to send and receive two-way audio data after the interphone is turned on, which is only used for intercom with the channel.
Typical URL	POST or GET http://<host>[:port]/channel_talk[/channelId]
Channel ID	Optional. If the URL does not contain a channel ID, the default channel ID is 1.
Action name	None
Entity Data	None
Successful Response	<URL>
Applicable products	NVR
<p><URL> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <URL type="string"><!--Required.URL for audio intercom.--></URL> </config></pre>	
<p>Example of Successful Response:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <URL type="string"><![CDATA[rtsp://192.168.0.9:554/intercom/1]]> </URL> </config></pre>	
<p>[Tips]:</p> <ol style="list-style-type: none"> 1. Refer to the "Tips" of the "Talkback" API. 	

10 Smart commands

10.1 Face Detect

10.1.1 GetSmartVfdConfig

GetSmartVfdConfig	
Description	To get the IP media device's Video Face Detection configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartVfdConfig[/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	<vfd>
Applicable products	IPC,NVR
<vfd> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <detectAlarmModeType><!--Required.--> <enum>allAlarm</enum><!--Required. If a face is detected, an alarm will be triggered.--> <enum>noWearmask</enum><!--Required. The alarm is triggered only when a face without a mask is detected.--> </detectAlarmModeType> </types> <vfd><!--Required.--> <switch type="boolean">	

```

        <!--Required.Face detection/Face comparison is enabled or not.--></switch>
    <detectAlarmMode type="detectAlarmModeType">
        <!--Required. The conditions for triggering an alarm.-->
    </detectAlarmMode>
    <alarmHoldTime type="uint32" min="" max="">
        <!--Required; The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <saveFacePicture type="boolean">
        <!--Optional; boolean,save the face picture to local storage or not.-->
    </saveFacePicture>
    <saveSourcePicture type="boolean">
        <!--Optional; boolean,save the source picture to local storage or not.-->
    </saveSourcePicture>
    <regionInfo type="list" maxCount="" count=""><!--Optional; Specify the area to be detected.-->
        <item>
            <X1 type="uint32"><!--Required; X coordinate in the upper left corner.--></X1>
            <Y1 type="uint32"><!--Required; Y coordinate in the upper left corner.--></Y1>
            <X2 type="uint32"><!--Required; X coordinate in the bottom right corner.--></X2>
            <Y2 type="uint32"><!--Required; Y coordinate in the bottom right corner.--></Y2>
        </item>
    </regionInfo>
    <maxFaceFrame type="uint16">
        <!--Required; The maximum size of a face, The width or height of a face must not exceed
        this value.-->
    </maxFaceFrame>
    <minFaceFrame type="uint16">
        <!--Required; The minimum size of a face, The width or height of a face must not be less
        than this value.-->
    </minFaceFrame>
</vfd>
</config>

```

Example of Successful Response:

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

```

<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <detectAlarmModeType>
      <enum>allAlarm</enum>
      <enum>noWearmask</enum>
    </detectAlarmModeType>
  </types>
  <vfd>
    <switch type="boolean">true</switch>
    <detectAlarmMode type="detectAlarmModeType">allAlarm</detectAlarmMode>
    <alarmHoldTime type="uint32">20</alarmHoldTime>
    <saveFacePicture type="boolean">>false</saveFacePicture>
    <saveSourcePicture type="boolean">>false</saveSourcePicture>
    <regionInfo type="list" maxCount="1" count="1">
      <item>
        <X1 type="uint32">750</X1>
        <Y1 type="uint32">1000</Y1>
        <X2 type="uint32">8111</X2>
        <Y2 type="uint32">8666</Y2>
      </item>
    </regionInfo>
    <maxFaceFrame type="uint16">5000</maxFaceFrame>
    <minFaceFrame type="uint16">500</minFaceFrame>
  </vfd>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.1.2 SetSmartVfdConfig

SetSmartVfdConfig

Description	To set the IP media device's Video Face Detection configuration.
Typical URL	POST http://<host>[:port]/SetSmartVfdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<vfd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<vfd> XML Block:

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

```
  <vfd><!--Required.-->
```

```
    <switch><!--Required:boolean,Face detection/Face comparison is enabled or not.--></switch>
```

```
    <detectAlarmMode>
```

```
      <!--Required. detectAlarmModeType,The conditions for triggering an alarm.-->
```

```
    </detectAlarmMode>
```

```
    <alarmHoldTime>
```

```
      <!--Required; uint32,The duration of the alarm after the event ends, Unit: seconds.-->
```

```
    </alarmHoldTime>
```

```
    <saveFacePicture>
```

```
      <!--Optional; boolean,save the face picture to local storage or not.-->
```

```
    </saveFacePicture>
```

```
    <saveSourcePicture>
```

```
      <!--Optional; boolean,save the source picture to local storage or not.-->
```

```
    </saveSourcePicture>
```

```
    <regionInfo><!--Optional; Specify the area to be detected.-->
```

```
      <item>
```

```
        <X1><!--Required; uint32,X coordinate in the upper left corner.--></X1>
```

```
        <Y1><!--Required; uint32,Y coordinate in the upper left corner.--></Y1>
```

```
        <X2><!--Required; uint32,X coordinate in the bottom right corner.--></X2>
```

```

        <Y2><!--Required; uint32,Y coordinate in the bottom right corner.--></Y2>
    </item>
</regionInfo>
<maxFaceFrame >
    <!--Required; uint16,The maximum size of a face, The width or height of a face must not
    exceed this value.-->
</maxFaceFrame>
<minFaceFrame>
    <!--Required; uint16,The minimum size of a face, The width or height of a face must not be
    less than this value.-->
</minFaceFrame>
</vfd>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <vfd>
        <switch>true</switch>
        <detectAlarmMode>allAlarm</detectAlarmMode>
        <alarmHoldTime>20</alarmHoldTime>
        <saveFacePicture>>false</saveFacePicture>
        <saveSourcePicture>>false</saveSourcePicture>
        <regionInfo>
            <item>
                <X1>750</X1>
                <Y1>1000</Y1>
                <X2>8111</X2>
                <Y2>8666</Y2>
            </item>
        </regionInfo>
        <maxFaceFrame>5000</maxFaceFrame>
        <minFaceFrame>500</minFaceFrame>
    </vfd>

```

</config>

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.1.3 SearchSnapFaceByTime

SearchSnapFaceByTime	
Description	Search for captured pictures of faces by time.
Typical URL	POST http://<host>[:port]/SearchSnapFaceByTime[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<search>
Successful Response	<captureFaceList>
Applicable products	IPC
<p><search> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <search><!--Required;search criteria.--> <starttime> <!--Required;string, start time.format:YYYY-MM-DD HH:MM:SS.--> </starttime> <endtime> <!--Required;string, end time. The end time needs to be greater than or equal to the start time.format:YYYY-MM-DD HH:MM:SS.--> </endtime> </search> </config></pre>	

<captureFaceList> XML Block:

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
  <captureFaceList type="list" maxCount="" count=""><!--Required;List of capture face.-->
    <item>
      <snapTime type="uint64"><!--Required;time stamp.--></snapTime>
      <faceID type="uint32"><!--Required;face ID.--></faceID>
    </item>
  </captureFaceList>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
  <search>
    <starttime>
      <![CDATA[2024-10-30 00:00:00]]>
    </starttime>
    <endtime>
      <![CDATA[2024-10-30 23:59:59]]>
    </endtime>
  </search>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
  <captureFaceList type="list" maxCount="1000" count="3">
    <item>
      <snapTime type="uint64">6234564566</snapTime>
      <faceID type="uint32">66</faceID>
    </item>
    <item>
      <snapTime type="uint64">6234780985</snapTime>
```

```

        <faceID type="uint32">195</faceID>
    </item>
    <item>
        <snapTime type="uint64">7645456908</snapTime>
        <faceID type="uint32">10320</faceID>
    </item>
</captureFaceList>
</config>

```

[Tips]:

1. Maximum return of 1000 valid result information.

10.1.4 SearchSnapFaceByKey

SearchSnapFaceByKey	
Description	Search for captured pictures of faces by face id.
Typical URL	POST http://<host>[:port]/SearchSnapFaceByKey[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<search>
Successful Response	<snapFace>
Applicable products	IPC
<p><search> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <search><!--Required;search criteria.--> <snapTime><!--Required; uint64, time stamp.--></snapTime> <faceID><!--Required;uint32,face ID.--></faceID> <requestPanoramicPic> <!--Optional;boolean, the search results carry the original picture or not.--> </pre>	

```

    </requestPanoramicPic>
    <requestPersonPic>
        <!--Optional;boolean, the search results carry the person picture or not.-->
    </requestPersonPic>
</search>
</config>

```

<snapFace> XML Block:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
    <types>
        <listType><!--Optional; Define the classification of face management.-->
            <enum>strangerList</enum><!--Required.-->
            <enum>whiteList</enum><!--Required.-->
            <enum>blackList</enum><!--Required.-->
        </listType>
        <sexType><!--Optional; Define the value of gender.-->
            <enum>male</enum>
            <enum>female</enum>
        </sexType>
        <formatType><!--Optional; Define supported image formats.-->
            <enum>jpg</enum>
        </formatType>
    </types>
    <snapFace><!--Required; picture information that meets the search criteria.-->
        <snapInfo><!--Optional;snap picture infomation.-->
            <time type="string">
                <!--Required;string, snap time.format:YYYY-MM-DD HH:MM:SS.-->
            </time>
            <pictureData type="string">
                <!--Required;string, picture data(Base64 encoded data).-->
            </pictureData>

```

```

<width type="uint32"><!--Required; uint32, the width of the picture.--></width>
<height type="uint32"><!-- Required; uint32, the height of the picture.--></height>
<format type="formatType">
    <!--Required; formatType,picture format, Currently only supports the "jpg" format.-->
</format>
<size type="uint32">
    <!--Required; uint32, the size of the picture. size before base64 encryption.-->
</size>
</snapInfo>
<panoramicInfo><!--Optional;source picture information.-->
    <pictureData type="string">
        <!--Required;string, picture data(Base64 encoded data).-->
    </pictureData>
    <width type="uint32"><!--Required; uint32, the width of the picture.--></width>
    <height type="uint32"><!-- Required; uint32, the height of the picture.--></height>
    <format type="formatType">
        <!--Required; formatType,picture format, Currently only supports the "jpg" format.-->
    </format>
    <size type="uint32">
        <!--Required; uint32, the size of the picture. size before base64 encryption.-->
    </size>
</panoramicInfo>
</snapFace>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
    <search>
        <snapTime>6234564566</snapTime>
        <faceID>66</faceID>
        <requestPanoramicPic>true</requestPanoramicPic>
        <requestPersonPic>true</requestPersonPic>
    </search>
</config>

```

```
</search>
```

```
</config>
```

Example of Successful Response:

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

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
```

```
  <types>
```

```
    <listType>
```

```
      <enum>strangerList</enum>
```

```
      <enum>whiteList</enum>
```

```
      <enum>blackList</enum>
```

```
    </listType>
```

```
    <sexType>
```

```
      <enum>male</enum>
```

```
      <enum>female</enum>
```

```
    </sexType>
```

```
    <formatType>
```

```
      <enum>jpg</enum>
```

```
    </formatType>
```

```
  </types>
```

```
  <snapFace>
```

```
    <snapInfo>
```

```
      <time type="string">
```

```
        <![CDATA[2017-06-30 00:00:00]]>
```

```
      </time>
```

```
      <pictureData type="string" maxLen="95576">
```

```
        <![CDATA[Base64 Picture Data]]><!--The content is too long and has been omitted.-->
```

```
      </pictureData>
```

```
      <width type="uint32">100</width>
```

```
      <height type="uint32">80</height>
```

```
      <format type="formatType">jpg</format>
```

```
      <size type="uint32">50000</size>
```

```

</snapInfo>
<panoramicInfo>
  <pictureData type="string" maxLen="95576">
    <![CDATA[Base64 Picture Data]]><!--The content is too long and has been omitted.-->
  </pictureData>
  <width type="uint32">100</width>
  <height type="uint32">80</height>
  <format type="formatType">jpg</format>
  <size type="uint32">50000</size>
</panoramicInfo>
</snapFace>
</config>

```

[Tips]:

1.If there is no comparison, there will be no matchInfo element. If the comparison fails, there will be no personInfo node. If requestPanoramicPic is false, there will be no panoramicInfo node. If requestPersonPic is false, there will be no picInfo node.

10.2 Intrusion

10.2.1 GetSmartPerimeterConfig

GetSmartPerimeterConfig	
Description	To get the IP media device's Perimeter configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartPerimeterConfig[/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	<perimeter>
Applicable products	IPC,NVR

<perimeter> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <perimeter><!--Required; perimeter configurations.-->
    <switch type="boolean"><!--Optional; boolean,perimeter is enabled or not.--></switch>
    <alarmHoldTime type="uint32" min="" max="">
      <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
      <objectFilter> definition in the <boundary> element for details. It is reserved for
      compatibility with older versions in history. If you are using it for the first time, please
      ignore this element.-->
    </objectFilter>
    <saveTargetPicture type="boolean">
      <!--Optional; boolean,save the target picture to Local storage or not.-->
    </saveTargetPicture>
    <saveSourcePicture type="boolean">
      <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <boundary type="list" maxCount="" count=""><!--Optional; Detection parameters for each
      detection region.-->
      <item>
        <objectFilter><!--Optional; Filter parameters for detecting targets.-->
        <person><!--Optional; person filter parameters.-->
          <switch type="boolean"><!--Required;Enabled or not.--></switch>
          <sensitivity type="uint32" max="" min="" default="">
            <!--Required;Detection sensitivity.-->
          </sensitivity>
          <minDetectTarget><!--Optional; The minimum target size that needs to be
            detected.-->
            <width type="uint32" max="" min="" default="">
              <!--Required;Minimum target width.-->
            </width>
            <height type="uint32" max="" min="" default="">
              <!--Required;Minimum target height.-->
            </height>
          </minDetectTarget>
        </person>
      </item>
    </boundary>
  </perimeter>
</config>
```

```
</height>
</minDetectTarget>
<maxDetectTarget><!--Optional; The maximum target size that needs to be
detected.-->
  <width type="uint32" max="" min="" default="">
    <!--Required;Minimum target width.-->
  </width>
  <height type="uint32" max="" min="" default="">
    <!--Required;Minimum target height.-->
  </height>
</maxDetectTarget>
</person>
<car><!--Optional; car filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" max="" min="" default=""></sensitivity>
  <minDetectTarget>
    <width type="uint32" max="" min="" default=""></width>
    <height type="uint32" max="" min="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="" min="" default=""></width>
    <height type="uint32" max="" min="" default=""></height>
  </maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" max="" min="" default=""></sensitivity>
  <minDetectTarget>
    <width type="uint32" max="" min="" default=""></width>
    <height type="uint32" max="" min="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="" min="" default=""></width>
```

```

        <height type="uint32" max="" min="" default=""></height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="" maxCount="" count=""><!--Required;
Coordinates of the detection area. Specifically, when count is 0, it indicates that
no detection area has been set.-->
    <item>
        <X type="uint32"><!--Required;X-coordinate--></X>
        <Y type="uint32"><!--Required;Y-coordinate--></Y>
    </item>
</pointGroup>
</item>
</boundary>
</perimeter>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <perimeter>
        <switch type="boolean">true</switch>
        <alarmHoldTime type="uint32" min="1" max="120">1</alarmHoldTime>
        <objectFilter>
            <person>
                <switch type="boolean">true</switch>
                <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
                <minDetectTarget>
                    <width type="uint32" max="10000" min="100" default="300">100</width>
                    <height type="uint32" max="10000" min="100" default="600">100</height>
                </minDetectTarget>
                <maxDetectTarget>
                    <width type="uint32" max="10000" min="100" default="9000">9000</width>
                    <height type="uint32" max="10000" min="100" default="9000">9000</height>
                </maxDetectTarget>
            </person>
        </objectFilter>
    </perimeter>
</config>

```

```
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100" default="300">100</width>
    <height type="uint32" max="10000" min="100" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100" default="9000">9000</width>
    <height type="uint32" max="10000" min="100" default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100" default="300">100</width>
    <height type="uint32" max="10000" min="100" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100" default="9000">9000</width>
    <height type="uint32" max="10000" min="100" default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture type="boolean">true</saveTargetPicture>
<saveSourcePicture type="boolean">true</saveSourcePicture>
<boundary type="list" maxCount="4" count="4">
  <item>
    <objectFilter>
```

```
<person>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
```

```
        default="50">50</sensitivity>
    <minDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="300">100</width>
        <height type="uint32" max="10000" min="100"
            default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
            default="9000">9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="8" count="4">
    <item>
        <X type="uint32">2175</X>
        <Y type="uint32">3000</Y>
    </item>
    <item>
        <X type="uint32">4825</X>
        <Y type="uint32">7000</Y>
    </item>
    <item>
        <X type="uint32">5850</X>
        <Y type="uint32">3600</Y>
    </item>
    <item>
        <X type="uint32">4575</X>
        <Y type="uint32">1700</Y>
    </item>
</pointGroup>
</item>
```

```
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">60</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">60</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </car>
  </objectFilter>
</item>
```

```
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">60</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="8" count="4">
  <item>
    <X type="uint32">2175</X>
    <Y type="uint32">3000</Y>
  </item>
  <item>
    <X type="uint32">5825</X>
    <Y type="uint32">7000</Y>
  </item>
  <item>
    <X type="uint32">6850</X>
    <Y type="uint32">3600</Y>
  </item>
  <item>
    <X type="uint32">7575</X>
    <Y type="uint32">2700</Y>
  </item>
</pointGroup>
```

```
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </car>
  </objectFilter>
</item>
```

```
        default="9000">9000</height>
    </maxDetectTarget>
</car>
<motor>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
    <minDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="300">100</width>
        <height type="uint32" max="10000" min="100"
            default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
            default="9000">9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="8" count="4">
    <item>
        <X type="uint32">8175</X>
        <Y type="uint32">7000</Y>
    </item>
    <item>
        <X type="uint32">5825</X>
        <Y type="uint32">5000</Y>
    </item>
    <item>
        <X type="uint32">1850</X>
        <Y type="uint32">2600</Y>
    </item>
</pointGroup>
```

```
<item>
  <X type="uint32">3575</X>
  <Y type="uint32">4700</Y>
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
    </car>
  </objectFilter>
</item>
```

```
<maxDetectTarget>
  <width type="uint32" max="10000" min="100"
    default="9000">9000</width>
  <height type="uint32" max="10000" min="100"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="8" count="4">
  <item>
    <X type="uint32">2175</X>
    <Y type="uint32">2000</Y>
  </item>
  <item>
    <X type="uint32">7825</X>
    <Y type="uint32">6000</Y>
  </item>
  <item>
```

```

        <X type="uint32">5850</X>
        <Y type="uint32">4600</Y>
    </item>
    <item>
        <X type="uint32">3575</X>
        <Y type="uint32">2700</Y>
    </item>
</pointGroup>
</item>
</boundary>
</perimeter>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.2.2 SetSmartPerimeterConfig

SetSmartPerimeterConfig	
Description	To set the IP media device's Perimeter configuration.
Typical URL	POST http://<host>[:port]/SetSmartPerimeterConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<perimeter>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><perimeter> XML Block:</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="" xmlns="http://www.ipc.com/ver10"> </pre>	

```

<perimeter><!--Required; perimeter configurations.-->
  <switch><!--Optional; boolean,perimeter is enabled or not.--></switch>
  <alarmHoldTime>
    <!--Optional; uint32,The duration of the alarm after the event ends, Unit: seconds.-->
  </alarmHoldTime>
  <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
  <objectFilter> definition in the <boundary> element for details. It is reserved for
  compatibility with older versions in history. If you are using it for the first time, please
  ignore this element.-->
  </objectFilter>
  <saveTargetPicture>
    <!--Optional; boolean,save the target picture to Local storage or not.-->
  </saveTargetPicture>
  <saveSourcePicture>
    <!--Optional; boolean,save the source picture to Local storage or not.-->
  </saveSourcePicture>
  <boundary><!--Optional; Detection parameters for each detection region.-->
    <item>
      <objectFilter><!--Optional; Filter parameters for detecting targets.-->
      <person><!--Optional; person filter parameters.-->
        <switch"><!--Required;boolean,Enabled or not.--></switch>
        <sensitivity>
          <!--Required;uint32,Detection sensitivity.-->
        </sensitivity>
        <minDetectTarget><!--Optional; The minimum target size that needs to be
        detected.-->
          <width>
            <!--Required;uint32,minimum target width.-->
          </width>
          <height>
            <!--Required;uint32, minimum target height.-->
          </height>
        </minDetectTarget>
        <maxDetectTarget><!--Optional; The maximum target size that needs to be

```

```
        detected.-->
        <width>
            <!--Required;uint32,Minimum target width.-->
        </width>
        <height>
            <!--Required;uint32, minimum target height.-->
        </height>
    </maxDetectTarget>
</person>
<car><!--Optional; car filter parameters, refer to element <person>.-->
    <switch></switch>
    <sensitivity></sensitivity>
    <minDetectTarget>
        <width></width>
        <height></height>
    </minDetectTarget>
    <maxDetectTarget>
        <width></width>
        <height></height>
    </maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
    <switch></switch>
    <sensitivity></sensitivity>
    <minDetectTarget>
        <width></width>
        <height></height>
    </minDetectTarget>
    <maxDetectTarget>
        <width></width>
        <height></height>
    </maxDetectTarget>
```

```
        </motor>
    </objectFilter>
    <pointGroup><!--Required; Coordinates of the detection area. Specifically, when the
        count of points is 0, it means clearing the detection area.-->
        <item><!--Optional.-->
            <X><!--Required:uint32,X-coordinate--></X>
            <Y><!--Required:uint32,Y-coordinate--></Y>
        </item>
    </pointGroup>
</item>
</boundary>
</perimeter>
</config>
```

Example of Entity Data:

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <perimeter>
        <switch>true</switch>
        <alarmHoldTime>1</alarmHoldTime>
        <saveTargetPicture>true</saveTargetPicture>
        <saveSourcePicture>true</saveSourcePicture>
    </boundary>
    <item>
        <objectFilter>
            <person>
                <switch>true</switch>
                <sensitivity>50</sensitivity>
                <minDetectTarget>
                    <width>100</width>
                    <height>100</height>
                </minDetectTarget>
                <maxDetectTarget>
                    <width>9000</width>
```

```
<height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>2175</X>
    <Y>3000</Y>
```

```
</item>
<item>
  <X>4825</X>
  <Y>7000</Y>
</item>
<item>
  <X>5850</X>
  <Y>3600</Y>
</item>
<item>
  <X>4575</X>
  <Y>1700</Y>
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>60</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>60</sensitivity>
```

```
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>60</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>3175</X>
    <Y>4000</Y>
  </item>
  <item>
    <X>5825</X>
    <Y>6000</Y>
  </item>
  <item>
```

```
<X>7850</X>
<Y>4600</Y>
</item>
<item>
  <X>5575</X>
  <Y>2700</Y>
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>70</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>70</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
```

```
<height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>70</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>1175</X>
    <Y>2000</Y>
  </item>
  <item>
    <X>3825</X>
    <Y>6000</Y>
  </item>
  <item>
    <X>4850</X>
    <Y>2600</Y>
  </item>
  <item>
    <X>3575</X>
    <Y>700</Y>
  </item>
</pointGroup>
</objectFilter>
</car>
```

```
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </car>
    <motor>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
```

```
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>175</X>
    <Y>1000</Y>
  </item>
  <item>
    <X>2825</X>
    <Y>5000</Y>
  </item>
  <item>
    <X>3850</X>
    <Y>1600</Y>
  </item>
  <item>
    <X>2575</X>
    <Y>3700</Y>
  </item>
</pointGroup>
</item>
</boundary>
</perimeter>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple<regionInfo>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.3 Line Crossing

10.3.1 GetSmartTripwireConfig

GetSmartTripwireConfig	
Description	To get the IP media device's Tripwire configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartTripwireConfig[/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	<tripwire>
Applicable products	IPC,NVR
<tripwire> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <tripwireDirection><!--Required; Define the direction of tripwire.--> <enum>any</enum> <enum>rightortop</enum> <enum>leftorbotton</enum> </tripwireDirection> </types> <tripwire><!--Required; tripwire configurations.--> <switch type="boolean"><!--Optional; boolean,tripwire is enabled or not.--></switch>	

```

<alarmHoldTime type="uint32" min="" max="">
    <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
</alarmHoldTime>
<saveTargetPicture type="boolean">
    <!--Optional; boolean,save the target picture to Local storage or not.-->
</saveTargetPicture>
<saveSourcePicture type="boolean">
    <!--Optional; boolean,save the source picture to Local storage or not.-->
</saveSourcePicture>
<objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
    <objectFilter> definition in the <lineInfo> element for details. It is reserved for compatibility
    with older versions in history. If you are using it for the first time, please ignore this
    element.-->
</objectFilter>
<lineInfo type="list" maxCount="" count=""><!--Optional; Detection parameters for each
    detection line.-->
    <item>
        <objectFilter><!--Optional; Filter parameters for detecting targets, Refer to the
            definition of the element with the same name in the API interface
            "GetSmartPerimeterConfig". -->
        </objectFilter>
        <direction type="tripwireDirection">
            <!--Required; the direction of tripwire.-->
        </direction>
        <startPoint><!--Required; The starting point coordinates of the line.-->
            <X type="uint32"><!--Required;X-coordinate--></X>
            <Y type="uint32"><!--Required;X-coordinate--></Y>
        </startPoint>
        <endPoint><!--Required; The endpoint coordinates of the line.-->
            <X type="uint32"><!--Required;X-coordinate--></X>
            <Y type="uint32"><!--Required;X-coordinate--></Y>
        </endPoint>
    </item>
</lineInfo>
</tripwire>

```

```
</config>
```

Example of Successful Response:

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

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
```

```
  <types>
```

```
    <tripwireDirection>
```

```
      <enum>rightortop</enum>
```

```
      <enum>leftforbotton</enum>
```

```
    </tripwireDirection>
```

```
  </types>
```

```
  <tripwire>
```

```
    <switch type="boolean">>true</switch>
```

```
    <alarmHoldTime type="uint32" min="1" max="120">3</alarmHoldTime>
```

```
    <saveTargetPicture type="boolean">>false</saveTargetPicture>
```

```
    <saveSourcePicture type="boolean">>false</saveSourcePicture>
```

```
    <objectFilter>
```

```
      <person>
```

```
        <switch type="boolean">>true</switch>
```

```
        <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
```

```
        <minDetectTarget>
```

```
          <width type="uint32" max="10000" min="100" default="300">100</width>
```

```
          <height type="uint32" max="10000" min="100" default="600">100</height>
```

```
        </minDetectTarget>
```

```
        <maxDetectTarget>
```

```
          <width type="uint32" max="10000" min="100" default="9000">9000</width>
```

```
          <height type="uint32" max="10000" min="100" default="9000">9000</height>
```

```
        </maxDetectTarget>
```

```
      </person>
```

```
      <car>
```

```
        <switch type="boolean">>true</switch>
```

```
        <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
```

```
        <minDetectTarget>
```

```
<width type="uint32" max="10000" min="100" default="300">100</width>
<height type="uint32" max="10000" min="100" default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" max="10000" min="100" default="9000">9000</width>
  <height type="uint32" max="10000" min="100" default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100" default="300">100</width>
    <height type="uint32" max="10000" min="100" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100" default="9000">9000</width>
    <height type="uint32" max="10000" min="100" default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<lineInfo type="list" maxCount="4" count="4">
  <item>
    <objectFilter>
      <person>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" max="100" min="1"
          default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" max="10000" min="100"
            default="300">100</width>
          <height type="uint32" max="10000" min="100"
            default="600">100</height>
        </minDetectTarget>
      </person>
    </objectFilter>
  </item>
</lineInfo>
</objectFilter>
</objectFilter>
```

```
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" max="10000" min="100"
    default="9000">9000</width>
  <height type="uint32" max="10000" min="100"
    default="9000">9000</height>
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
```

```
        default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
            default="9000">9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<direction type="tripwireDirection">rightortop</direction>
<startPoint>
    <X type="uint32">975</X>
    <Y type="uint32">4866</Y>
</startPoint>
<endPoint>
    <X type="uint32">8725</X>
    <Y type="uint32">5000</Y>
</endPoint>
</item>
<item>
    <objectFilter>
        <person>
            <switch type="boolean">>true</switch>
            <sensitivity type="uint32" max="100" min="1"
                default="50">50</sensitivity>
            <minDetectTarget>
                <width type="uint32" max="10000" min="100"
                    default="300">100</width>
                <height type="uint32" max="10000" min="100"
                    default="600">100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width type="uint32" max="10000" min="100"
                    default="9000">9000</width>
                <height type="uint32" max="10000" min="100"
                    default="9000">9000</height>
            </maxDetectTarget>
        </person>
    </objectFilter>
</item>
</item>
```

```
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<direction type="tripwireDirection">rightortop</direction>
```

```
<startPoint>
  <X type="uint32">3975</X>
  <Y type="uint32">4866</Y>
</startPoint>
<endPoint>
  <X type="uint32">8725</X>
  <Y type="uint32">9000</Y>
</endPoint>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">>true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">>true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
```

```
<height type="uint32" max="10000" min="100"
  default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" max="10000" min="100"
    default="9000">9000</width>
  <height type="uint32" max="10000" min="100"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" max="100" min="1"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="300">100</width>
    <height type="uint32" max="10000" min="100"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" max="10000" min="100"
      default="9000">9000</width>
    <height type="uint32" max="10000" min="100"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<direction type="tripwireDirection">rightortop</direction>
<startPoint>
  <X type="uint32">1975</X>
  <Y type="uint32">2866</Y>
</startPoint>
<endPoint>
  <X type="uint32">7725</X>
```

```
<Y type="uint32">6000</Y>
```

```
</endPoint>
```

```
</item>
```

```
<item>
```

```
<objectFilter>
```

```
<person>
```

```
<switch type="boolean">>true</switch>
```

```
<sensitivity type="uint32" max="100" min="1"  
default="50">50</sensitivity>
```

```
<minDetectTarget>
```

```
<width type="uint32" max="10000" min="100"  
default="300">100</width>
```

```
<height type="uint32" max="10000" min="100"  
default="600">100</height>
```

```
</minDetectTarget>
```

```
<maxDetectTarget>
```

```
<width type="uint32" max="10000" min="100"  
default="9000">9000</width>
```

```
<height type="uint32" max="10000" min="100"  
default="9000">9000</height>
```

```
</maxDetectTarget>
```

```
</person>
```

```
<car>
```

```
<switch type="boolean">>true</switch>
```

```
<sensitivity type="uint32" max="100" min="1"  
default="50">50</sensitivity>
```

```
<minDetectTarget>
```

```
<width type="uint32" max="10000" min="100"  
default="300">100</width>
```

```
<height type="uint32" max="10000" min="100"  
default="600">100</height>
```

```
</minDetectTarget>
```

```
<maxDetectTarget>
```

```
<width type="uint32" max="10000" min="100"  
default="9000">9000</width>
```

```
<height type="uint32" max="10000" min="100"
```

```
        default="9000">9000</height>
    </maxDetectTarget>
</car>
<motor>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
    <minDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="300">100</width>
        <height type="uint32" max="10000" min="100"
            default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
            default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
            default="9000">9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<direction type="tripwireDirection">rightortop</direction>
<startPoint>
    <X type="uint32">75</X>
    <Y type="uint32">866</Y>
</startPoint>
<endPoint>
    <X type="uint32">8725</X>
    <Y type="uint32">5000</Y>
</endPoint>
</item>
</lineInfo>
</tripwire>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.3.2 SetSmartTripwireConfig

SetSmartTripwireConfig	
Description	To set the IP media device's Tripwire configuration.
Typical URL	POST http://<host>[:port]/SetSmartTripwireConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<tripwire>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><tripwire> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <tripwire><!--Required; tripwire configurations.--> <switch><!--Optional; boolean,tripwire is enabled or not.--></switch> <alarmHoldTime> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <saveTargetPicture> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture> <lineInfo><!--Optional; Detection parameters for each detection line.--></pre>	

```

    <item>
      <objectFilter><!--Optional; Filter parameters for detecting targets, Refer to the
        definition of the element with the same name in the API interface
        "GetSmartPerimeterConfig". -->
    </objectFilter>
    <direction>
      <!--Required; tripwireDirection, the direction of tripwire.-->
    </direction>
    <startPoint><!--Required; The starting point coordinates of the line.-->
      <X><!--Required;uint32,X-coordinate--></X>
      <Y><!--Required; uint32,Y-coordinate--></Y>
    </startPoint>
    <endPoint><!--Required; The endpoint coordinates of the line.-->
      <X><!--Required; uint32,X-coordinate--></X>
      <Y><!--Required; uint32,Y-coordinate--></Y>
    </endPoint>
  </item>
</lineInfo>
</tripwire>
</config>

```

Example of Entity Data

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <tripwire>
    <switch>true</switch>
    <alarmHoldTime>3</alarmHoldTime>
    <saveTargetPicture>>false</saveTargetPicture>
    <saveSourcePicture>>false</saveSourcePicture>
    <lineInfo>
      <item>
        <objectFilter>
          <person>
            <switch>true</switch>

```

```
<sensitivity>50</sensitivity>
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
```

```
        </maxDetectTarget>
    </motor>
</objectFilter>
<direction>rightortop</direction>
<startPoint>
    <X>975</X>
    <Y>4866</Y>
</startPoint>
<endPoint>
    <X>8725</X>
    <Y>5000</Y>
</endPoint>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>60</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </person>
        <car>
            <switch>true</switch>
            <sensitivity>60</sensitivity>
            <minDetectTarget>
                <width>100</width>
```

```
<height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>60</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<direction>rightortop</direction>
<startPoint>
  <X>1975</X>
  <Y>2866</Y>
</startPoint>
<endPoint>
  <X>8725</X>
  <Y>5000</Y>
</endPoint>
</item>
<item>
  <objectFilter>
```

```
<person>
  <switch>true</switch>
  <sensitivity>70</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>70</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>70</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
```

```
<width>9000</width>
<height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<direction>rightortop</direction>
<startPoint>
  <X>3975</X>
  <Y>4866</Y>
</startPoint>
<endPoint>
  <X>7725</X>
  <Y>5000</Y>
</endPoint>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
```

```
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<direction>rightortop</direction>
<startPoint>
  <X>0</X>
  <Y>0</Y>
</startPoint>
<endPoint>
  <X>0</X>
  <Y>0</Y>
</endPoint>
</item>
```

```

    </lineInfo>

    </tripwire>

</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple <lineInfo>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.4 Object Removal

10.4.1 GetSmartOscConfig

GetSmartOscConfig	
Description	To get the IP media device's Object Status Change configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartOscConfig[/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	<osc>
Applicable products	IPC,NVR
<p><osc> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <oscObject><!--Required; Define the type of OSC.--> <enum>abandum</enum> <enum>objstolen</enum> </oscObject> </pre>	

```

</types>
<osc><!--Required; OSC configurations.-->
  <switch type="boolean"><!--Optional; boolean,OSC is enabled or not.--></switch>
  <oscObject type="oscObject"><!--Optional;oscObject,OSC type.--></oscObject>
  <duration type="uint32" min="" max="">
    <!--Optional; Trigger an alarm after the OSC event continues to trigger for a threshold
    duration.-->
  </duration>
  <alarmHoldTime type="uint32" min="" max="">
    <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
  </alarmHoldTime>
  <boundary count=""><!--Optional; Detection parameters for each detection region.-->
    <item>
      <pointGroup><!--Required; Coordinates of the detection region. Refer to the
      definition of the element with the same name in the API interface
      "GetSmartPerimeterConfig"-->
    </pointGroup>
    </item>
  </boundary>
</osc>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <oscObject>
      <enum>abandum</enum>
      <enum>objstolen</enum>
    </oscObject>
  </types>
  <osc>
    <switch type="boolean">true</switch>
    <oscObject type="oscObject">abandum</oscObject>
    <duration type="uint32" min="3" max="3600">10</duration>
  </osc>
</config>

```

```
<alarmHoldTime type="uint32">20</alarmHoldTime>
<boundary type="list" maxCount="1" count="1">
  <item>
    <pointGroup type="list" minCount="4" maxCount="6" count="4">
      <item>
        <X type="uint32">2175</X>
        <Y type="uint32">2000</Y>
      </item>
      <item>
        <X type="uint32">7825</X>
        <Y type="uint32">6000</Y>
      </item>
      <item>
        <X type="uint32">5850</X>
        <Y type="uint32">4600</Y>
      </item>
      <item>
        <X type="uint32">3575</X>
        <Y type="uint32">2700</Y>
      </item>
    </pointGroup>
  </item>
</boundary>
</osc>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.4.2 SetSmartOscConfig

SetSmartOscConfig	
Description	To set the IP media device's Object Status Change configuration.
Typical URL	POST http://<host>[:port]/SetSmartOscConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<osc>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><osc> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <osc><!--Required; OSC configurations.--> <switch><!--Optional; boolean,OSC is enabled or not.--></switch> <oscObject><!--Optional;oscObject,OSC type.--></oscObject> <duration> <!--Optional; Trigger an alarm after the OSC event continues to trigger for a threshold duration.--> </duration> <alarmHoldTime> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <boundary><!--Optional; Detection parameters for each detection region.--> <item> <pointGroup><!--Required; Coordinates of the detection region. Refer to the definition of the element with the same name in the API interface "SetSmartPerimeterConfig".--> </pointGroup> </item> </boundary> </pre>	

```
</osc>
```

Example of Entity Data

```
<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <osc>
    <switch>true</switch>
    <oscObject>abandum</oscObject>
    <duration>10</duration>
    <alarmHoldTime>20</alarmHoldTime>
    <boundary>
      <item>
        <pointGroup count="4">
          <item>
            <X>2175</X>
            <Y>2000</Y>
          </item>
          <item>
            <X>7825</X>
            <Y>6000</Y>
          </item>
          <item>
            <X>5850</X>
            <Y>4600</Y>
          </item>
          <item>
            <X>3575</X>
            <Y>2700</Y>
          </item>
        </pointGroup>
      </item>
    </boundary>
  </osc>
```

</config>

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.5 Exception

10.5.1 GetSmartAvdConfig

GetSmartAvdConfig	
Description	To get the IP media device's Abnormal Video Detection configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartAvdConfig[/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	<avd>
Applicable products	IPC,NVR
<avd> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <avd><!--Required;avd's configuration.--> <alarmHoldTime type="uint32" min="" max=""> <!--Optional; uint32, the duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <sceneChangeSwitch type="boolean"> <!--Optional; boolean, scene change detection or not.--> </sceneChangeSwitch> <clarityAbnormalSwitch type="boolean"> <!--Optional; boolean, clarity abnormal detection or not.--> </avd> </config>	

```

</clarityAbnormalSwitch>
<colorAbnormalSwitch type="boolean">
    <!--Optional; boolean, color abnormal detection or not.-->
</colorAbnormalSwitch>
<sensitivity type="uint32" min="1" max="100">
    <!--Optional; uint32, detection sensitivity.-->
</sensitivity>
</avd>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <avd>
    <alarmHoldTime type="uint32" min="1" max="120">20</alarmHoldTime>
    <sceneChangeSwitch type="boolean">true</sceneChangeSwitch>
    <clarityAbnormalSwitch type="boolean">true</clarityAbnormalSwitch>
    <colorAbnormalSwitch type="boolean">true</colorAbnormalSwitch>
    <sensitivity type="uint32" min="1" max="100">100</sensitivity>
  </avd>
</config>

```

[Tips]:

10.5.2 SetSmartAvdConfig

SetSmartAvdConfig	
Description	To set the IP media device’s Abnormal Video Detection configuration.
Typical URL	POST http://<host>[:port]/SetSmartAvdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None

Entity Data	<avd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><avd> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <avd><!--Required;avd's configuration.--> <alarmHoldTime> <!--Optional; uint32, the duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <sceneChangeSwitch> <!--Optional; boolean, scene change detection or not.--> </sceneChangeSwitch> <clarityAbnormalSwitch> <!--Optional; boolean, clarity abnormal detection or not.--> </clarityAbnormalSwitch> <colorAbnormalSwitch> <!--Optional; boolean, color abnormal detection or not.--> </colorAbnormalSwitch> <sensitivity> <!--Optional; uint32, detection sensitivity.--> </sensitivity> </avd> </config> </pre>	
<p>Example of Entity Data:</p> <pre> <?xml version="1.0" encoding="utf-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <avd> <alarmHoldTime>20</alarmHoldTime> <sceneChangeSwitch>true</sceneChangeSwitch> <clarityAbnormalSwitch>true</clarityAbnormalSwitch> </pre>	

```

    <colorAbnormalSwitch>true</colorAbnormalSwitch>

    <sensitivity>100</sensitivity>

  </avd>
</config>

```

[Tips]:

10.5.3 GetSmartAsdConfig

GetSmartAsdConfig	
Description	To get the IP media device's Abnormal Audio Detection configuration.
Typical URL	POST or GET http://<host>[:port]/GetSmartAsdConfig[/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	<asd>
Applicable products	IPC,NVR
<p><asd> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <asd><!--Required;asd's configuration.--> <switch type="boolean"><!--Optional; boolean,ASD is enabled or not.--></switch> <objectFilter><!--Optional; boolean,ASD is enabled or not.--> <soundRise><!--Optional; sound rise detection configuration.--> <switch type="boolean"> <!--Optional; sound rise detection enabled or not.--> </switch> <sensitivity type="uint32" min="" max="" default=""> <!--Optional; detection sensitivity.--> </sensitivity> <soundThreshold type="uint32" min="" max="" default=""> <!--Optional; Sound intensity threshold, triggering an alarm when the sound </pre>	

```
intensity exceeds this threshold.-->
    </soundThreshold>
</soundRise>
<soundReduce><!--Optional; sound reduce detection configuration.-->
    <switch type="boolean">
        <!--Optional; sound reduce detection enabled or not.-->
    </switch>
    <sensitivity type="uint32" min="" max="" default="">
        <!--Optional; detection sensitivity.-->
    </sensitivity>
</soundReduce>
</objectFilter>
<alarmHoldTime type="uint32" min="" max="">
    <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
</alarmHoldTime>
</asd>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <asd>
        <switch type="boolean">true</switch>
    <objectFilter>
        <soundRise>
            <switch type="boolean">false</switch>
            <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
            <soundThreshold type="uint32" min="1" max="100"
                default="50">50</soundThreshold>
        </soundRise>
        <soundReduce>
            <switch type="boolean">false</switch>
            <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
```

```

        </soundReduce>
    </objectFilter>
    <alarmHoldTime type="uint32" min="1" max="120">5</alarmHoldTime>
</asd>
</config>

```

[Tips]:

10.5.4 SetSmartAsdConfig

SetSmartAsdConfig	
Description	To set the IP media device's Abnormal Audio Detection configuration.
Typical URL	POST http://<host>[:port]/SetSmartAsdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<asd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><asd> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <asd><!--Required;asd's configuration.--> <switch><!--Optional; boolean,ASD is enabled or not.--></switch> <objectFilter><!--Optional; boolean,ASD is enabled or not.--> <soundRise><!--Optional; sound rise detection configuration.--> <switch><!--Optional; sound rise detection enabled or not.--></switch> <sensitivity><!--Optional; uint32,detection sensitivity.--></sensitivity> <soundThreshold> <!--Optional; Sound intensity threshold, triggering an alarm when the sound intensity exceeds this threshold.--> </soundThreshold> </soundRise> </objectFilter> </asd> </config> </pre>	

```

    </soundRise>
    <soundReduce><!--Optional; sound reduce detection configuration.-->
        <switch><!--Optional; sound reduce detection enabled or not.--></switch>
        <sensitivity><!--Optional; detection sensitivity.--></sensitivity>
    </soundReduce>
</objectFilter>
<alarmHoldTime>
    <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
</alarmHoldTime>
</asd>
</config>

```

Example of Entity Data:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <asd>
        <switch>true</switch>
        <objectFilter>
            <soundRise>
                <switch>>false</switch>
                <sensitivity>50</sensitivity>
                <soundThreshold>50</soundThreshold>
            </soundRise>
            <soundReduce>
                <switch>>false</switch>
                <sensitivity>50</sensitivity>
            </soundReduce>
        </objectFilter>
        <alarmHoldTime>5</alarmHoldTime>
    </asd>
</config>

```

[Tips]:

10.6 Region Entrance

10.6.1 GetSmartAoiEntryConfig

GetSmartAoiEntryConfig	
Description	To get aoientry's details.
Typical URL	POST or GET http://<host>[:port]/GetSmartAoiEntryConfig[/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	<aoientry>
Applicable products	IPC,NVR
<p><aoientry> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <aoientry><!--Required; aoi entry configurations.--> <switch type="boolean"><!--Optional; boolean,aoi entry is enabled or not.--></switch> <alarmHoldTime type="uint32" min="" max=""> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <saveTargetPicture type="boolean"> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture type="boolean"> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture></pre>	

```
<boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
element with the same name in the API interface "GetSmartPerimeterConfig". -->
```

```
</boundary>
```

```
</aoientry>
```

```
</config>
```

Example of Successful Response:

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

```
<config version="2.0.0"
```

```
  xmlns="http://www.ipc.com/ver10">
```

```
  <aoientry>
```

```
    <switch type="boolean">true</switch>
```

```
    <alarmHoldTime type="uint32" min="1" max="120">20</alarmHoldTime>
```

```
    <objectFilter>
```

```
      <person>
```

```
        <switch type="boolean">true</switch>
```

```
        <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
```

```
        <minDetectTarget>
```

```
          <width type="uint32" min="100" max="10000" default="300">100</width>
```

```
          <height type="uint32" min="100" max="10000" default="600">100</height>
```

```
        </minDetectTarget>
```

```
        <maxDetectTarget>
```

```
          <width type="uint32" min="100" max="10000" default="9000">9000</width>
```

```
          <height type="uint32" min="100" max="10000" default="9000">9000</height>
```

```
        </maxDetectTarget>
```

```
      </person>
```

```
      <car>
```

```
        <switch type="boolean">true</switch>
```

```
        <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
```

```
        <minDetectTarget>
```

```
          <width type="uint32" min="100" max="10000" default="300">100</width>
```

```
          <height type="uint32" min="100" max="10000" default="600">100</height>
```

```
        </minDetectTarget>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="9000">9000</width>
  <height type="uint32" min="100" max="10000" default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture type="boolean">>false</saveTargetPicture>
<saveSourcePicture type="boolean">>false</saveSourcePicture>
<boundary type="list" maxCount="4" count="4">
  <item>
    <objectFilter>
      <person>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="300">100</width>
          <height type="uint32" min="100" max="10000"
            default="600">100</height>
        </minDetectTarget>
      </person>
    </objectFilter>
  </item>
</boundary>
</saveSourcePicture>
</saveTargetPicture>
</objectFilter>
</motor>
</car>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
```

```
<height type="uint32" min="100" max="10000"
  default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="4">
  <item >
    <X type="uint32">1925</X>
    <Y type="uint32">3366</Y>
  </item>
  <item >
    <X type="uint32">1950</X>
    <Y type="uint32">9166</Y>
  </item>
  <item >
    <X type="uint32">8325</X>
    <Y type="uint32">9166</Y>
  </item>
  <item >
    <X type="uint32">8350</X>
    <Y type="uint32">2900</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="300">100</width>
        <height type="uint32" min="100" max="10000">
```

```
        default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
            default="9000">9000</width>
        <height type="uint32" min="100" max="10000"
            default="9000">9000</height>
    </maxDetectTarget>
</person>
<car>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
    <minDetectTarget>
        <width type="uint32" min="100" max="10000"
            default="300">100</width>
        <height type="uint32" min="100" max="10000"
            default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
            default="9000">9000</width>
        <height type="uint32" min="100" max="10000"
            default="9000">9000</height>
    </maxDetectTarget>
</car>
<motor>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
    <minDetectTarget>
        <width type="uint32" min="100" max="10000"
            default="300">100</width>
        <height type="uint32" min="100" max="10000"
            default="600">100</height>
    </minDetectTarget>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="300">100</width>
        <height type="uint32" min="100" max="10000"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="9000">9000</width>
        <height type="uint32" min="100" max="10000"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="300">100</width>
```

```
<height type="uint32" min="100" max="10000"
  default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
```

```
<minDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="300">100</width>
  <height type="uint32" min="100" max="10000"
    default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
```

```

        <height type="uint32" min="100" max="10000"
            default="600">100</height>

        </minDetectTarget>

        <maxDetectTarget>

            <width type="uint32" min="100" max="10000"
                default="9000">9000</width>

            <height type="uint32" min="100" max="10000"
                default="9000">9000</height>

        </maxDetectTarget>

    </motor>
</objectFilter>

    <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
</aoientry>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.6.2 SetSmartAoiEntryConfig

SetSmartAoiEntryConfig	
Description	To set aoientry's details.
Typical URL	POST http://<host>[:port]/SetSmartAoiEntryConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<aoientry>
Successful Response	The standard successful result response that described in 1.3.5.

Applicable products

IPC,NVR

<aoientry> XML Block:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="" xmlns="http://www.ipc.com/ver10">
  <aoientry><!--Required; aoI leave configurations.-->
    <switch><!--Optional; boolean,detection is enabled or not.--></switch>
    <alarmHoldTime>
      <!--Optional; uint32,The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
      <objectFilter> definition in the <boundary> element for details. It is reserved for
      compatibility with older versions in history. If you are using it for the first time, please
      ignore this element.-->
    </objectFilter>
    <saveTargetPicture>
      <!--Optional; boolean,save the target picture to Local storage or not.-->
    </saveTargetPicture>
    <saveSourcePicture>
      <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
      element with the same name in the API interface "SetSmartPerimeterConfig". -->
    </boundary>
  </aoientry>
</config>
```

Example of Entity Data:

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <aoientry>
    <switch>true</switch>
    <alarmHoldTime>1</alarmHoldTime>
    <saveTargetPicture>true</saveTargetPicture>
    <saveSourcePicture>true</saveSourcePicture>
    <boundary>
```

```
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </car>
    <motor>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
    </motor>
  </objectFilter>
</item>
```

```
</minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>2175</X>
    <Y>3000</Y>
  </item>
  <item>
    <X>4825</X>
    <Y>7000</Y>
  </item>
  <item>
    <X>5850</X>
    <Y>3600</Y>
  </item>
  <item>
    <X>4575</X>
    <Y>1700</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>60</sensitivity>
      <minDetectTarget>
```

```
<width>100</width>
<height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>60</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>60</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
```

```
</objectFilter>
  <pointGroup></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>70</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>70</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </car>
    <motor>
      <switch>true</switch>
      <sensitivity>70</sensitivity>
```

```
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
```

```
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>80</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
</boundary>
</aoientry>
</config>
```

[Tips]: 1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.7 Region Leave

10.7.1 GetSmartAoiLeaveConfig

GetSmartAoiLeaveConfig	
Description	To get aoileave's details
Typical URL	POST or GET http://<host>[:port]/GetSmartAoiLeaveConfig[/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	<aoileave>
Applicable products	IPC,NVR
<aoileave> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <aoileave><!--Required; aoileave configurations.--> <switch type="boolean"><!--Optional; boolean,aoi entry is enabled or not.--></switch> <alarmHoldTime type="uint32" min="" max=""> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <saveTargetPicture type="boolean"> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture type="boolean"> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture> <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the	

element with the same name in the API interface "SetSmartPerimeterConfig". -->

</boundary>

</aoileave>

</config>

Example of Successful Response:

<!--Referring to the example of the API interface "GetSmartAoiEntryConfig". The difference is that the element name of "GetSmartAoiExitConfig" is <aoientry>, while the element name of "GetSmartAoiLeaveConfig" is <aoileave>, and the definition of the element is the same.-->

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.7.2 SetSmartAoiLeaveConfig

SetSmartAoiLeaveConfig	
Description	To set aoileave's details
Typical URL	POST http://<host>[:port]/SetSmartAoiEntryConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<aoileave>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<aoileave> XML Block: <?xml version="1.0" encoding="UTF-8"?> <config version="" xmlns="http://www.ipc.com/ver10"> <aoileave><!--Required; aoi leave configurations.--> <switch><!--Optional; boolean.perimeter is enabled or not.--></switch> <alarmHoldTime>	

```

        <!--Optional; uint32,The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
    <objectFilter> definition in the <boundary> element for details. It is reserved for
    compatibility with older versions in history. If you are using it for the first time, please
    ignore this element.-->
    </objectFilter>
    <saveTargetPicture>
        <!--Optional; boolean,save the target picture to Local storage or not.-->
    </saveTargetPicture>
    <saveSourcePicture>
        <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
    element with the same name in the API interface "SetSmartPerimeterConfig". -->
    </boundary>
</aoileave>
</config>

```

Example of Entity Data:

<!--Referring to the example of the API interface "SetSmartAoiEntryConfig". The difference is that the element name of "SetSmartAoiExitConfig" is <aoientry>, while the element name of "SetSmartAoiLeaveConfig" is <aoileave>, and the definition of the element is the same.-->

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.8 Target Counting

10.8.1 GetSmartPassLineCountConfig

GetSmartPassLineCountConfig	
Description	To get passlinecount's details

Typical URL	POST or GET http://<host>[:port]/GetPassLineCountConfig[/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	<passlinecount>
Applicable products	IPC,NVR

<passlinecount> XML Block:

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

```
  <types>
```

```
    <direction><!--Required; Define the direction of tripwire.-->
```

```
      <enum>rightortop</enum>
```

```
      <enum>leftforbotton</enum>
```

```
    </direction>
```

```
    <countCycleType><!--Required; Define the period for statistical reset.-->
```

```
      <enum>day</enum>
```

```
      <enum>week</enum>
```

```
      <enum>month</enum>
```

```
      <enum>off</enum>
```

```
    </countCycleType>
```

```
  </types>
```

```
  <passlinecount><!--Required; tripwire configurations.-->
```

```
    <switch type="boolean"><!--Optional; boolean,passline count is enabled or not.--></switch>
```

```
    <stayAlarmDelay type="uint32" min="" max="">
```

```
      <!--Optional; The duration threshold for statistical values to continuously exceed the
      specified threshold. Unit: seconds.-->
```

```
    </stayAlarmDelay>
```

```
    <alarmHoldTime type="uint32" min="" max="">
```

```
      <!--Optional;The duration of the alarm after the event ends, Unit: seconds.-->
```

```
    </alarmHoldTime>
```

```
    <sendTarget type="boolean">
```

```
<!--Optional; Whether to send the target's image to an external party. In some scenarios,
users only need statistical values and do not need images to save network traffic.NVR is
not supported.-->
```

```
</sendTarget>
```

```
<saveTargetPicture type="boolean">
```

```
<!--Optional; boolean,save the target picture to Local storage or not.-->
```

```
</saveTargetPicture>
```

```
<saveSourcePicture type="boolean">
```

```
<!--Optional; boolean,save the source picture to Local storage or not.-->
```

```
</saveSourcePicture>
```

```
<objectFilter><!--Optional; Filter parameters for detecting targets.-->
```

```
<person><!--Optional; person filter parameters.-->
```

```
<switch type="boolean"><!--Required;Enabled or not.--></switch>
```

```
<sensitivity type="uint32" min="" max="" default="">
```

```
<!--Required;uint32,Detection sensitivity.-->
```

```
</sensitivity>
```

```
<stayAlarmThreshold type="uint32" min="" max="" default="">
```

```
<!--Optional;uint32, Statistical threshold for triggering alarms.-->
```

```
</stayAlarmThreshold>
```

```
<minDetectTarget><!--Optional; The minimum target size that needs to be detected.-->
```

```
<width type="uint32" min="" max="" default="">
```

```
<!--Required;Minimum target width.-->
```

```
</width>
```

```
<height type="uint32" min="" max="" default="">
```

```
<!--Required;Minimum target height.-->
```

```
</height>
```

```
</minDetectTarget>
```

```
<maxDetectTarget><!--Optional; The maximum target size that needs to be detected.-->
```

```
<width type="uint32" min="" max="" default="">
```

```
<!--Required;Minimum target width.-->
```

```
</width>
```

```
<height type="uint32" min="" max="" default="">
```

```
<!--Required;Minimum target height.-->
```

```
</height>
  </maxDetectTarget>
</person>
<car><!--Optional; car filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" min="" max="" default=""></sensitivity>
  <stayAlarmThreshold type="uint32" min="" max="" default="">
</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" min="" max="" default=""></sensitivity>
  <stayAlarmThreshold type="uint32" min="" max="" default="">
</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </maxDetectTarget>
</motor>
</objectFilter>
```

```

<countPeriod><!--Optional; The period for resetting statistical values to 0.-->
  <countTimeType type="countCycleType">
    <!--Required; countCycleType,Type of reset cycle.-->
  </countTimeType>
  <daily><!--Dependent; Valid when countTimeType is 'day'.-->
    <dateTimeSpan type="string">
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </daily>
  <weekly><!--Dependent; Valid when countTimeType is 'week'.-->
    <dateSpan type="uint32">
      <!--Required; Define the days of the week.Range[0, 6]. 0: Sunday, 1: Monday,
      2: Tuesday, 3: Wednesday, and so on.-->
    </dateSpan>
    <dateTimeSpan type="string">
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </weekly>
  <monthly><!--Dependent; Valid when countTimeType is 'month'.-->
    <dateSpan type="uint32">
      <!--Required; Define the days of the month. 1: The 1st day of the month, 2: The
      2nd day of the month, and so on.-->
    </dateSpan>
    <dateTimeSpan type="string">
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </monthly>
</countPeriod>
<countOSD><!--Optional; OSD configuration for statistical values.-->
  <switch type="boolean"><!--Optional; OSD display enabled or not.--></switch>
  <X type="uint32">
    <!--Optional; The X-coordinate in the upper left corner of the OSD content.-->
  </X>

```

```
<Y type="uint32">
    <!--Optional; The Y-coordinate in the upper left corner of the OSD content.-->
</Y>
<showEnterOsd type="boolean">
    <!--Optional; Display the "Enter" target statistics or not.-->
</showEnterOsd>
<osdEntranceName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Enter".-->
</osdEntranceName>
<showExitOsd type="boolean">
    <!--Optional; Display the "Exit" target statistics or not.-->
</showExitOsd>
<osdExitName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Exit".-->
</osdExitName>
<showStayOsd type="boolean">
    <!--Optional; Display the "Stay" target statistics or not.-->
</showStayOsd>
<osdStayName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Stay".-->
</osdStayName>
<osdAlarmName type="string" maxCharNum="">
    <!--Optional; string, The content displayed on the OSD when an alarm occurs.-->
</osdAlarmName>
<osdWelcomeName type="string" maxCharNum="">
    <!--Optional; string, The content displayed on the OSD when there is no alarm.-->
</osdWelcomeName>
</countOSD>
<line type="list" maxCount="" count=""><!--Optional; Configuration of rule lines.-->
  <item>
    <direction type="direction">
      <!--Optional; direction,Cross line direction.-->
    </direction>
  </item>
</line>
</countOSD>
```

```

        </direction>
        <startPoint><!--Optional; The starting point of the rule line.-->
            <X type="uint32">
                <!--Required; The starting point x coordinate of the rule line.-->
            </X>
            <Y type="uint32">
                <!--Required; The starting point y coordinate of the rule line.-->
            </Y>
        </startPoint>
        <endPoint><!--Optional; The endpoint of the rule line.-->
            <X type="uint32">
                <!--Required; The starting point x coordinate of the rule line.-->
            </X>
            <Y type="uint32">
                <!--Required; The starting point y coordinate of the rule line.-->
            </Y>
        </endPoint>
    </item>
</line>
<passlinecount>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <direction>
            <enum>rightortop</enum>
            <enum>leftforbotton</enum>
        </direction>
        <countCycleType>
            <enum>day</enum>
            <enum>week</enum>

```

```
<enum>month</enum>
<enum>off</enum>
</countCycleType>
</types>
<passlinecount>
  <switch type="boolean">true</switch>
  <stayAlarmDelay type="uint32" min="0" max="999" default="0">100</stayAlarmDelay>
  <alarmHoldTime type="uint32" min="1" max="120">20</alarmHoldTime>
  <sendTarget type="boolean">>false</sendTarget>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
      <stayAlarmThreshold type="uint32" min="0" max="10000"
        default="100">0</stayAlarmThreshold>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000" default="300">100</width>
        <height type="uint32" min="100" max="10000" default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000" default="9000">9000</width>
        <height type="uint32" min="100" max="10000" default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
      <stayAlarmThreshold type="uint32" min="0" max="10000"
        default="100">0</stayAlarmThreshold>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000" default="300">100</width>
        <height type="uint32" min="100" max="10000" default="600">100</height>
      </minDetectTarget>
    </car>
  </objectFilter>
</passlinecount>
</types>
</countCycleType>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="9000">9000</width>
  <height type="uint32" min="100" max="10000" default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <stayAlarmThreshold type="uint32" min="0" max="10000"
    default="100">0</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture type="boolean">true</saveTargetPicture>
<saveSourcePicture type="boolean">true</saveSourcePicture>
<countPeriod>
  <countTimeType type="countCycleType">day</countTimeType>
  <daily>
    <dateSpan type="uint32">0</dateSpan>
    <dateTimeSpan type="string">23:00:00</dateTimeSpan>
  </daily>
</countPeriod>
<countOSD>
  <switch type="boolean">true</switch>
  <X type="uint32">2185</X>
  <Y type="uint32">1033</Y>
```

```
<showEnterOsd type="boolean">true</showEnterOsd>
<osdEntranceName type="string" maxCharNum="12">
  <![CDATA[Entry]]>
</osdEntranceName>
<showExitOsd type="boolean">true</showExitOsd>
<osdExitName type="string" maxCharNum="12">
  <![CDATA[Exit]]>
</osdExitName>
<showStayOsd type="boolean">true</showStayOsd>
<osdStayName type="string" maxCharNum="12">
  <![CDATA[Stay]]>
</osdStayName>
<osdAlarmName type="string" maxCharNum="16">
  <![CDATA[Please wait]]>
</osdAlarmName>
<osdWelcomeName type="string" maxCharNum="16">
  <![CDATA[Welcome]]>
</osdWelcomeName>
</countOSD>
<line type="list" maxCount="2" count="1">
  <item>
    <direction type="direction">rightortop</direction>
    <startPoint>
      <X type="uint32">0</X>
      <Y type="uint32">0</Y>
    </startPoint>
    <endPoint>
      <X type="uint32">4675</X>
      <Y type="uint32">6300</Y>
    </endPoint>
  </item>
</line>
```

<pre></passlinecount> </config></pre>
<p>[Tips]:</p> <p>1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.</p>

10.8.2 SetSmartPassLineCountConfig

SetSmartPassLineCountConfig	
Description	To set passlinecount's details
Typical URL	POST http://<host>[:port]/SetSmartPassLineCountConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<passlinecount>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><passlinecount> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <passlinecount><!--Required; tripwire configurations.--> <switch><!--Optional; boolean,passline count is enabled or not.--></switch> <stayAlarmDelay> <!--Optional; uint32,The duration threshold for statistical values to continuously exceed the specified threshold. Unit: seconds.--> </stayAlarmDelay> <alarmHoldTime> <!--Optional;uint32,The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <sendTarget> <!--Optional;boolean,Whether to send the target's image to an external party. In some</pre>	

scenarios, users only need statistical values and do not need images to save network traffic.NVR is not supported.-->

</sendTarget>

<saveTargetPicture>

<!--Optional; boolean,save the target picture to Local storage or not.-->

</saveTargetPicture>

<saveSourcePicture>

<!--Optional; boolean,save the source picture to Local storage or not.-->

</saveSourcePicture>

<objectFilter><!--Optional; Filter parameters for detecting targets.-->

<person><!--Optional; person filter parameters.-->

<switch><!--Required;boolean,Enabled or not.--></switch>

<sensitivity>

<!--Required;uint32,Detection sensitivity.-->

</sensitivity>

<stayAlarmThreshold>

<!--Optional;uint32, Statistical threshold for triggering alarms.-->

</stayAlarmThreshold>

<minDetectTarget><!--Optional; The minimum target size that needs to be detected.-->

<width><!--Required;uint32,Minimum target width.--></width>

<height><!--Required;uint32,Minimum target height.--></height>

</minDetectTarget>

<maxDetectTarget><!--Optional; The maximum target size that needs to be detected.-->

<width><!--Required;uint32,Minimum target width.--></width>

<height><!--Required;uint32,Minimum target height.--></height>

</maxDetectTarget>

</person>

<car><!--Optional; car filter parameters, refer to element <person>.-->

<switch ></switch>

<sensitivity></sensitivity>

<stayAlarmThreshold></stayAlarmThreshold>

<minDetectTarget>

<width></width>

```
<height></height>
</minDetectTarget>
<maxDetectTarget>
  <width></width>
  <heigh></height>
</maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
  <switch></switch>
  <sensitivity></sensitivity>
  <stayAlarmThreshold></stayAlarmThreshold>
  <minDetectTarget>
    <width></width>
    <height></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width></width>
    <height></height>
  </maxDetectTarget>
</motor>
</objectFilter>
<countPeriod><!--Optional; The period for resetting statistical values to 0.-->
  <countTimeType>
    <!--Required; countCycleType,Type of reset cycle.-->
  </countTimeType>
  <daily><!--Dependent; Valid when countTimeType is 'day'.-->
    <dateTimeSpan>
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </daily>
  <weekly><!--Dependent; Valid when countTimeType is 'week'.-->
    <dateSpan>
      <!--Required; Define the days of the week. Range[0, 6]. 0: Sunday, 1: Monday,
```

```
                2: Tuesday, 3: Wednesday, and so on.-->
            </dateSpan>
            <dateTimeSpan>
                <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
            </dateTimeSpan>
        </weekly>
        <monthly><!--Dependent; Valid when countTimeType is 'month'.-->
            <dateSpan>
                <!--Required; Define the days of the month. 1: The 1st day of the month, 2: The
                2nd day of the month, and so on.-->
            </dateSpan>
            <dateTimeSpan>
                <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
            </dateTimeSpan>
        </monthly>
    </countPeriod>
    <countOSD><!--Optional; OSD configuration for statistical values.-->
        <switch><!--Optional; OSD display enabled or not.--></switch>
        <X>
            <!--Optional; The X-coordinate in the upper left corner of the OSD content.-->
        </X>
        <Y>
            <!--Optional; The Y-coordinate in the upper left corner of the OSD content.-->
        </Y>
        <showEnterOsd>
            <!--Optional; Display the "Enter" target statistics or not.-->
        </showEnterOsd>
        <osdEntranceName>
            <!--Optional; string, the OSD name for the target statistics of "Enter".-->
        </osdEntranceName>
        <showExitOsd>
            <!--Optional; Display the "Exit" target statistics or not.-->
        </showExitOsd>
```

```

<osdExitName>
    <!--Optional; string, the OSD name for the target statistics of "Exit".-->
</osdExitName>
<showStayOsd>
    <!--Optional; Display the "Stay" target statistics or not.-->
</showStayOsd>
<osdStayName>
    <!--Optional; string, the OSD name for the target statistics of "Stay".-->
</osdStayName>
<osdAlarmName>
    <!--Optional; string, The content displayed on the OSD when an alarm occurs.-->
</osdAlarmName>
<osdWelcomeName>
    <!--Optional; string, The content displayed on the OSD when there is no alarm.-->
</osdWelcomeName>
</countOSD>
<line><!--Optional; Configuration of rule lines.-->
    <item>
        <direction><!--Optional; direction,Cross line direction.--></direction>
        <startPoint><!--Optional; The starting point of the rule line.-->
            <X><!--Required; The starting point x coordinate of the rule line.--></X>
            <Y><!--Required; The starting point y coordinate of the rule line.--></Y>
        </startPoint>
        <endPoint><!--Optional; The endpoint of the rule line.-->
            <X><!--Required; The starting point x coordinate of the rule line.--></X>
            <Y><!--Required; The starting point y coordinate of the rule line.--></Y>
        </endPoint>
    </item>
</line>
<passlinecount>
</config>

```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <passlinecount>
    <switch>true</switch>
    <stayAlarmDelay>100</stayAlarmDelay>
    <alarmHoldTime>20</alarmHoldTime>
    <sendTarget>false</sendTarget>
    <objectFilter>
      <person>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <stayAlarmThreshold>0</stayAlarmThreshold>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </person>
      <car>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <stayAlarmThreshold>0</stayAlarmThreshold>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </car>
    </objectFilter>
  </passlinecount>
</config>
```

```
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <stayAlarmThreshold>0</stayAlarmThreshold>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture>true</saveTargetPicture>
<saveSourcePicture>true</saveSourcePicture>
<countPeriod>
  <countTimeType>day</countTimeType>
  <daily>
    <dateTimeSpan>23:00:00</dateTimeSpan>
  </daily>
</countPeriod>
<countOSD>
  <switch>true</switch>
  <X>2185</X>
  <Y>1033</Y>
  <showEnterOsd>true</showEnterOsd>
  <osdEntranceName>
    <![CDATA[Entry]]>
  </osdEntranceName>
```

```
<showExitOsd>true</showExitOsd>
<osdExitName>
  <![CDATA[Exit]]>
</osdExitName>
<showStayOsd>true</showStayOsd>
<osdStayName>
  <![CDATA[Stay]]>
</osdStayName>
<osdAlarmName>
  <![CDATA[Please wait]]>
</osdAlarmName>
<osdWelcomeName>
  <![CDATA[Welcome]]>
</osdWelcomeName>
</countOSD>
<line>
  <item>
    <direction>rightortop</direction>
    <startPoint>
      <X>0</X>
      <Y>0</Y>
    </startPoint>
    <endPoint>
      <X>4675</X>
      <Y>6300</Y>
    </endPoint>
  </item>
</line>
</passlinecount>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution

of 10000 * 10000.

2. If manually triggering the statistical clearing of 0, the interface is as follows(Manual Reset):

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <passlinecount>
    <forceReset>true</forceReset>
  </passlinecount>
</config>
```

10.12.3 GetPassLineCountStatistics

GetPassLineCountStatistics	
Description	Get current statistics
Typical URL	POST or GET http://<host>[:port]/GetPassLineCountStatistics[/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	<config>
Applicable products	IPC,NVR
<config> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <entranceCount><!--Required; Target statistics for entering the rule line.--> <person type="uint32"><!--Optional; person's statistical value.--></person> <car type="uint32"><!--Optional; car's statistical value.--></car> <bike type="uint32"><!--Optional; bike's statistical value.--></bike> </entranceCount> <exitCount><!--Required; Target statistics that leave the rule line.--> <person type="uint32"><!--Optional.--></person> <car type="uint32"><!--Optional.--></car></pre>	

```
<bike type="uint32"><!--Optional.--></bike>
</exitCount>
<stayCount><!--Required; The number of stay targets.-->
  <person type="uint32"><!--Optional.--></person>
  <car type="uint32"><!--Optional.--></car>
  <bike type="uint32"><!--Optional.--></bike>
</stayCount>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
  <entranceCount>
    <person type="uint32">100</person>
    <car type="uint32">200</car>
    <bike type="uint32">100</bike>
  </entranceCount>
  <exitCount>
    <person type="uint32">100</person>
    <car type="uint32">100</car>
    <bike type="uint32">0</bike>
  </exitCount>
  <stayCount>
    <person type="uint32">0</person>
    <car type="uint32">100</car>
    <bike type="uint32">100</bike>
  </stayCount>
</config>
```

[Tips]:

10.9 Heat Map

10.9.1 GetSmartHeatMapConfig

GetSmartHeatMapConfig	
Description	To get the IP media device's "heatMap" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartHeatMapConfig[/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	<heatMap>
Applicable products	IPC,NVR
<heatMap> XML Block: <pre><config version="" xmlns="http://www.ipc.com/ver10"> <heatMap><!--Required; heatMap configurations.--> <switch type="boolean"><!--Optional; boolean,heatMap detection is enabled or not.--></switch> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the element with the same name in the API interface "GetSmartPerimeterConfig". --> </boundary> </heatMap> </config></pre>	
Example of Successful Response: <pre><?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <heatMap> <switch type="boolean">false</switch></pre>	

```
<objectFilter>
  <person>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
    <minDetectTarget>
      <width type="uint32" min="100" max="10000" default="300">100</width>
      <height type="uint32" min="100" max="10000" default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width type="uint32" min="100" max="10000" default="9000">9000</width>
      <height type="uint32" min="100" max="10000" default="9000">9000</height>
    </maxDetectTarget>
  </person>
  <car>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
    <minDetectTarget>
      <width type="uint32" min="100" max="10000" default="300">100</width>
      <height type="uint32" min="100" max="10000" default="600">100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width type="uint32" min="100" max="10000" default="9000">9000</width>
      <height type="uint32" min="100" max="10000" default="9000">9000</height>
    </maxDetectTarget>
  </car>
  <motor>
    <switch type="boolean">true</switch>
    <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
    <minDetectTarget>
      <width type="uint32" min="100" max="10000" default="300">100</width>
      <height type="uint32" min="100" max="10000" default="600">100</height>
    </minDetectTarget>
  </motor>
</objectFilter>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="9000">9000</width>
  <height type="uint32" min="100" max="10000" default="9000">9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<boundary maxCount="4" count="4">
  <item>
    <objectFilter>
      <person>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="300">100</width>
          <height type="uint32" min="100" max="10000"
            default="600">100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="9000">9000</width>
          <height type="uint32" min="100" max="10000"
            default="9000">9000</height>
        </maxDetectTarget>
      </person>
      <car>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="300">100</width>
          <height type="uint32" min="100" max="10000"
            default="600">100</height>
        </minDetectTarget>
      </car>
    </objectFilter>
  </item>
</boundary>
</objectFilter>
</motor>
</objectFilter>
```

```
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="4">
  <item>
    <X type="uint32">100</X>
    <Y type="uint32">100</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
```

```
<person>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</person>

<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</car>

<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
```



```
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</car>

<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
```

</item>

<item>

<objectFilter>

<person>

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

<sensitivity type="uint32" min="1" max="100"
default="50">50</sensitivity>

<minDetectTarget>

<width type="uint32" min="100" max="10000"
default="300">100</width>

<height type="uint32" min="100" max="10000"
default="600">100</height>

</minDetectTarget>

<maxDetectTarget>

<width type="uint32" min="100" max="10000"
default="9000">9000</width>

<height type="uint32" min="100" max="10000"
default="9000">9000</height>

</maxDetectTarget>

</person>

<car>

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

<sensitivity type="uint32" min="1" max="100"
default="50">50</sensitivity>

<minDetectTarget>

<width type="uint32" min="100" max="10000"
default="300">100</width>

<height type="uint32" min="100" max="10000"
default="600">100</height>

</minDetectTarget>

<maxDetectTarget>

<width type="uint32" min="100" max="10000"
default="9000">9000</width>

<height type="uint32" min="100" max="10000"
default="9000">9000</height>

</maxDetectTarget>

```

    </car>
    <motor>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="300">100</width>
        <height type="uint32" min="100" max="10000"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="9000">9000</width>
        <height type="uint32" min="100" max="10000"
          default="9000">9000</height>
      </maxDetectTarget>
    </motor>
  </objectFilter>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
</heatMap>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.9.2 SetSmartHeatMapConfig

SetSmartHeatMapConfig	
Description	To set the IP media device's "heatMap" element.

Typical URL	POST http://<host>[:port]/SetSmartHeatMapConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<heatMap>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<heatMap> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <heatMap><!--Required; heatMap configurations.-->
    <switch><!--Optional; boolean,heatMap detection is enabled or not.--></switch>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
      <objectFilter> definition in the <boundary> element for details. It is reserved for
      compatibility with older versions in history. If you are using it for the first time, please
      ignore this element.-->
    </objectFilter>
    <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
      element with the same name in the API interface "SetSmartPerimeterConfig". -->
    </boundary>
  </heatMap>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <heatMap>
    <switch>>false</switch>
    <objectFilter>
      <person>
        <switch>>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
```

```
<width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
```

```
</objectFilter>
<boundary>
  <item>
    <objectFilter>
      <person>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </person>
      <car>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </car>
      <motor>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
```

```
<width>100</width>
  <height>100</height>
</minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item>
    <X>100</X>
    <Y>100</Y>
  </item>
  <item>
    <X>200</X>
    <Y>200</Y>
  </item>
  <item>
    <X>600</X>
    <Y>600</Y>
  </item>
  <item>
    <X>800</X>
    <Y>800</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
```

```
<sensitivity>50</sensitivity>
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
```

```
        </maxDetectTarget>
    </motor>
</objectFilter>
    <pointGroup></pointGroup>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </person>
        <car>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </car>
    </motor>
```

```
<switch>true</switch>
<sensitivity>50</sensitivity>
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
```

```
<width>100</width>
    <height>100</height>
</minDetectTarget>
<maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
</maxDetectTarget>
</car>
<motor>
    <switch>true</switch>
    <sensitivity>50</sensitivity>
    <minDetectTarget>
        <width>100</width>
        <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
</boundary>
</heatMap>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.10 Region Statistics

10.10.1 GetSmartTrafficConfig

GetSmartTrafficConfig	
Description	To get the IP media device's "traffic" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartTrafficConfig[/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	<traffic>
Applicable products	IPC,NVR
<traffic> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <types> <countCycleType><!--Required; Define the period for statistical reset.--> <enum>day</enum> <enum>week</enum> <enum>month</enum> <enum>off</enum> </countCycleType> </types> <traffic><!--Required; tripwire configurations.--> <switch type="boolean"><!--Optional; boolean,traffic count is enabled or not.--></switch> <stayAlarmDelay type="uint32" min="" max=""> <!--Optional; The duration threshold for statistical values to continuously exceed the specified threshold. Unit: seconds.--> </stayAlarmDelay> <alarmHoldTime type="uint32" min="" max=""> <!--Optional;The duration of the alarm after the event ends, Unit: seconds.--> </traffic> </config>	

```
</alarmHoldTime>
<saveTargetPicture type="boolean">
    <!--Optional; boolean,save the target picture to Local storage or not.-->
</saveTargetPicture>
<saveSourcePicture type="boolean">
    <!--Optional; boolean,save the source picture to Local storage or not.-->
</saveSourcePicture>
<objectFilter><!--Optional; Filter parameters for detecting targets.-->
    <person><!--Optional; person filter parameters.-->
        <switch type="boolean"><!--Required;Enabled or not.--></switch>
        <sensitivity type="uint32" min="" max="" default="">
            <!--Required;uint32,Detection sensitivity.-->
        </sensitivity>
        <stayAlarmThreshold type="uint32" min="" max="" default="">
            <!--Optional;uint32, Statistical threshold for triggering alarms.-->
        </stayAlarmThreshold>
        <minDetectTarget><!--Optional; The minimum target size that needs to be detected.-->
            <width type="uint32" min="" max="" default="">
                <!--Required;Minimum target width.-->
            </width>
            <height type="uint32" min="" max="" default="">
                <!--Required;Minimum target height.-->
            </height>
        </minDetectTarget>
        <maxDetectTarget><!--Optional; The maximum target size that needs to be detected.-->
            <width type="uint32" min="" max="" default="">
                <!--Required;Minimum target width.-->
            </width>
            <height type="uint32" min="" max="" default="">
                <!--Required;Minimum target height.-->
            </height>
        </maxDetectTarget>
```

```
</person>
<car><!--Optional; car filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" min="" max="" default=""></sensitivity>
  <stayAlarmThreshold type="uint32" min="" max="" default="">
</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
  <switch type="boolean"> </switch>
  <sensitivity type="uint32" min="" max="" default=""></sensitivity>
  <stayAlarmThreshold type="uint32" min="" max="" default="">
</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="" max="" default=""></width>
    <height type="uint32" min="" max="" default=""></height>
  </maxDetectTarget>
</motor>
</objectFilter>
<countPeriod><!--Optional; The period for resetting statistical values to 0.-->
  <countTimeType type="countCycleType">
```

```
<!--Required; countCycleType,Type of reset cycle.-->
</countTimeType>
<daily><!--Dependent; Valid when countTimeType is 'day'.-->
  <dateTimeSpan type="string">
    <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
  </dateTimeSpan>
</daily>
<weekly><!--Dependent; Valid when countTimeType is 'week'.-->
  <dateSpan type="uint32">
    <!--Required; Define the days of the week. Range[0, 6]. 0: Sunday, 1: Monday, 2:
    Tuesday, 3: Wednesday, and so on.-->
  </dateSpan>
  <dateTimeSpan type="string">
    <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
  </dateTimeSpan>
</weekly>
<monthly><!--Dependent; Valid when countTimeType is 'month'.-->
  <dateSpan type="uint32">
    <!--Required; Define the days of the month. 1: The 1st day of the month, 2: The
    2nd day of the month, and so on.-->
  </dateSpan>
  <dateTimeSpan type="string">
    <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
  </dateTimeSpan>
</monthly>
</countPeriod>
<countOSD><!--Optional; OSD configuration for statistical values.-->
  <switch type="boolean"><!--Optional; OSD display enabled or not.--></switch>
  <X type="uint32">
    <!--Optional; The X-coordinate in the upper left corner of the OSD content.-->
  </X>
  <Y type="uint32">
    <!--Optional; The Y-coordinate in the upper left corner of the OSD content.-->
```

```
</Y>
<showEnterOsd type="boolean">
    <!--Optional; Display the "Enter" target statistics or not.-->
</showEnterOsd>
<osdEntranceName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Enter".-->
</osdEntranceName>
<showExitOsd type="boolean">
    <!--Optional; Display the "Exit" target statistics or not.-->
</showExitOsd>
<osdExitName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Exit".-->
</osdExitName>
<showStayOsd type="boolean">
    <!--Optional; Display the "Stay" target statistics or not.-->
</showStayOsd>
<osdStayName type="string" maxCharNum="">
    <!--Optional; string, the OSD name for the target statistics of "Stay".-->
</osdStayName>
<osdAlarmName type="string" maxCharNum="">
    <!--Optional; string, The content displayed on the OSD when an alarm occurs.-->
</osdAlarmName>
<osdWelcomeName type="string" maxCharNum="">
    <!--Optional; string, The content displayed on the OSD when there is no alarm.-->
</osdWelcomeName>
</countOSD>
<boundary><!--Optional; Detection parameters for each detection region.-->
    <item>
        <pointGroup><!--Required; Coordinates of the detection region. Refer to the definition
            of the element with the same name in the API interface
            "GetSmartPerimeterConfig".-->
        </pointGroup>
    </item>
```

```
</boundary>
< traffic>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <countCycleType>
      <enum>day</enum>
      <enum>week</enum>
      <enum>month</enum>
      <enum>off</enum>
    </countCycleType>
  </types>
  <traffic>
    <switch type="boolean">true</switch>
    <alarmHoldTime type="uint32" min="1" max="120">20</alarmHoldTime>
    <stayAlarmDelay type="uint32" min="0" max="999" default="0">10</stayAlarmDelay>
    <objectFilter>
      <person>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
        <stayAlarmThreshold type="uint32" min="0" max="10000"
          default="100">100</stayAlarmThreshold>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000" default="300">100</width>
          <height type="uint32" min="100" max="10000" default="600">100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width type="uint32" min="100" max="10000" default="9000">9000</width>
          <height type="uint32" min="100" max="10000" default="9000">9000</height>
        </maxDetectTarget>
      </person>
    </objectFilter>
  </traffic>
</config>
```

```
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <stayAlarmThreshold type="uint32" min="0" max="10000"
    default="100">100</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <stayAlarmThreshold type="uint32" min="0" max="10000"
    default="100">100</stayAlarmThreshold>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture type="boolean">false</saveTargetPicture>
<saveSourcePicture type="boolean">false</saveSourcePicture>
<countPeriod>
```

```
<countTimeType type="countCycleType">day</countTimeType>
<daily>
  <dateTimeSpan type="string">00:00:00</dateTimeSpan>
</daily>
</countPeriod>
<countOSD>
  <switch type="boolean">true</switch>
  <X type="uint32">6600</X>
  <Y type="uint32">2400</Y>
  <showEnterOsd type="boolean">true</showEnterOsd>
  <osdEntranceName type="string" maxCharNum="12">
    <![CDATA[Entry]]>
  </osdEntranceName>
  <showExitOsd type="boolean">true</showExitOsd>
  <osdExitName type="string" maxCharNum="12">
    <![CDATA[Exit]]>
  </osdExitName>
  <showStayOsd type="boolean">true</showStayOsd>
  <osdStayName type="string" maxCharNum="12">
    <![CDATA[Stay]]>
  </osdStayName>
  <osdAlarmName type="string" maxCharNum="16">
    <![CDATA[Please wait]]>
  </osdAlarmName>
  <osdWelcomeName type="string" maxCharNum="16">
    <![CDATA[Welcome]]>
  </osdWelcomeName>
</countOSD>
<boundary type="list" maxCount="1" count="1">
  <item>
    <pointGroup type="list" minCount="4" maxCount="6" count="4">
      <item>
```

```

        <X type="uint32">1300</X>
        <Y type="uint32">1933</Y>
    </item>
    <item>
        <X type="uint32">1350</X>
        <Y type="uint32">8633</Y>
    </item>
    <item>
        <X type="uint32">7200</X>
        <Y type="uint32">8566</Y>
    </item>
    <item>
        <X type="uint32">7425</X>
        <Y type="uint32">1800</Y>
    </item>
</pointGroup>
</item>
</boundary>
</traffic>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.10.2 SetSmartTrafficConfig

SetSmartTrafficConfig	
Description	To set the IP media device's "traffic" element.
Typical URL	POST http://<host>[:port]/SetSmartTrafficConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.

Action name	None
Entity Data	<traffic>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<traffic> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <traffic><!--Required; tripwire configurations.-->
    <switch><!--Optional; boolean,traffic count is enabled or not.--></switch>
    <stayAlarmDelay>
      <!--Optional; uint32,The duration threshold for statistical values to continuously exceed the
      specified threshold. Unit: seconds.-->
    </stayAlarmDelay>
    <alarmHoldTime>
      <!--Optional;uint32,The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <saveTargetPicture>
      <!--Optional; boolean,save the target picture to Local storage or not.-->
    </saveTargetPicture>
    <saveSourcePicture>
      <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <objectFilter><!--Optional; Filter parameters for detecting targets.-->
      <person><!--Optional; person filter parameters.-->
        <switch "><!--Required;boolean,Enabled or not.--></switch>
        <sensitivity><!--Required;uint32,Detection sensitivity.--></sensitivity>
        <stayAlarmThreshold>
          <!--Optional;uint32, Statistical threshold for triggering alarms.-->
        </stayAlarmThreshold>
        <minDetectTarget><!--Optional; The minimum target size that needs to be detected.-->
          <width><!--Required;uint32,Minimum target width.--></width>
          <height><!--Required;uint32,Minimum target height.--></height>

```

```
</minDetectTarget>
  <maxDetectTarget><!--Optional; The maximum target size that needs to be detected.-->
    <width><!--Required;uint32,Minimum target width.--></width>
    <height><!--Required;uint32,Minimum target height.--></height>
  </maxDetectTarget>
</person>
<car><!--Optional; car filter parameters, refer to element <person>.-->
  <switch></switch>
  <sensitivity></sensitivity>
  <stayAlarmThreshold></stayAlarmThreshold>
  <minDetectTarget>
    <width></width>
    <height></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width></width>
    <height></height>
  </maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <person>.-->
  <switch></switch>
  <sensitivity></sensitivity>
  <stayAlarmThreshold></stayAlarmThreshold>
  <minDetectTarget>
    <width></width>
    <height></height>
  </minDetectTarget>
  <maxDetectTarget>
    <width></width>
    <height></height>
  </maxDetectTarget>
</motor>
```

```
</objectFilter>
<countPeriod><!--Optional; The period for resetting statistical values to 0.-->
  <countTimeType>
    <!--Required; countCycleType,Type of reset cycle.-->
  </countTimeType>
  <daily><!--Dependent; Valid when countTimeType is 'day'.-->
    <dateTimeSpan>
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </daily>
  <weekly><!--Dependent; Valid when countTimeType is 'week'.-->
    <dateSpan>
      <!--Required; Define the days of the week. 0: Sunday, 1: Monday, 2: Tuesday, 3:
      Wednesday, and so on.-->
    </dateSpan>
    <dateTimeSpan>
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </weekly>
  <monthly><!--Dependent; Valid when countTimeType is 'month'.-->
    <dateSpan>
      <!--Required; Define the days of the month. 1: The 1st day of the month, 2: The
      2nd day of the month, and so on.-->
    </dateSpan>
    <dateTimeSpan>
      <!--Required; Time to reset statistical values.format, HH:MM:SS.-->
    </dateTimeSpan>
  </monthly>
</countPeriod>
<countOSD><!--Optional; OSD configuration for statistical values.-->
  <switch><!--Optional;boolean,OSD display enabled or not.--></switch>
  <X>
    <!--Optional; The X-coordinate in the upper left corner of the OSD content.-->
```

```
</X>
<Y>
    <!--Optional; The Y-coordinate in the upper left corner of the OSD content.-->
</Y>
<showEnterOsd>
    <!--Optional; boolean,Display the 'Enter' target statistics or not.-->
</showEnterOsd>
<osdEntranceName>
    <!--Optional; string, the OSD name for the target statistics of 'Enter'.-->
</osdEntranceName>
<showExitOsd>
    <!--Optional; boolean,Display the 'Exit' target statistics or not.-->
</showExitOsd>
<osdExitName>
    <!--Optional; string, the OSD name for the target statistics of 'Exit'.-->
</osdExitName>
<showStayOsd>
    <!--Optional; boolean,Display the 'Stay' target statistics or not.-->
</showStayOsd>
<osdStayName>
    <!--Optional; string, the OSD name for the target statistics of 'Stay'.-->
</osdStayName>
<osdAlarmName>
    <!--Optional; string, The content displayed on the OSD when an alarm occurs.-->
</osdAlarmName>
<osdWelcomeName">
    <!--Optional; string, The content displayed on the OSD when there is no alarm.-->
</osdWelcomeName>
</countOSD>
<boundary><!--Optional; Detection parameters for each detection region.-->
    <item>
        <pointGroup><!--Required; Coordinates of the detection region. Refer to the definition
            of the element with the same name in the API interface
```

"SetSmartPerimeterConfig"-->

```
    </pointGroup>
  </item>
</boundary>
< traffic>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <traffic>
    <switch>true</switch>
    <alarmHoldTime>20</alarmHoldTime>
    <stayAlarmDelay>10</stayAlarmDelay>
    <objectFilter>
      <person>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <stayAlarmThreshold>100</stayAlarmThreshold>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </person>
      <car>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <stayAlarmThreshold>100</stayAlarmThreshold>
        <minDetectTarget>
```

```
<width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <stayAlarmThreshold>100</stayAlarmThreshold>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture>>false</saveTargetPicture>
<saveSourcePicture>>false</saveSourcePicture>
<countPeriod>
  <countTimeType>day</countTimeType>
  <daily>
    <dateTimeSpan>00:00:00</dateTimeSpan>
  </daily>
</countPeriod>
<countOSD>
  <switch>true</switch>
```

```
<X>6600</X>
<Y>2400</Y>
<showEnterOsd>true</showEnterOsd>
<osdEntranceName>
  <![CDATA[Entry]]>
</osdEntranceName>
<showExitOsd>true</showExitOsd>
<osdExitName>
  <![CDATA[Exit]]>
</osdExitName>
<showStayOsd>true</showStayOsd>
<osdStayName>
  <![CDATA[Stay]]>
</osdStayName>
<osdAlarmName>
  <![CDATA[Please wait]]>
</osdAlarmName>
<osdWelcomeName>
  <![CDATA[Welcome]]>
</osdWelcomeName>
</countOSD>
<boundary>
  <item>
    <pointGroup>
      <item>
        <X>1300</X>
        <Y>1933</Y>
      </item>
      <item>
        <X>1350</X>
        <Y>8633</Y>
      </item>
    </pointGroup>
  </item>
</boundary>
```

```

        <item>
            <X>7200</X>
            <Y>8566</Y>
        </item>
        <item>
            <X>7425</X>
            <Y>1800</Y>
        </item>
    </pointGroup>
</item>
</boundary>
</traffic>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.10.3 GetTrafficCountStatistics

GetTrafficCountStatistics	
Description	Get current statistics
Typical URL	POST or GET http://<host>[:port]/GetTrafficCountStatistics[/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	<config>
Applicable products	IPC,NVR
<config> XML Block: <config version="" xmlns="http://www.ipc.com/ver10">	

```

<entranceCount><!--Required; Target statistics for entering the rule line.-->
  <person type="uint32"><!--Optional; person's statistical value.--></person>
  <car type="uint32"><!--Optional; car's statistical value.--></car>
  <bike type="uint32"><!--Optional; bike's statistical value.--></bike>
</entranceCount>

<exitCount><!--Required; Target statistics that leave the rule line.-->
  <person type="uint32"><!--Optional.--></person>
  <car type="uint32"><!--Optional.--></car>
  <bike type="uint32"><!--Optional.--></bike>
</exitCount>

<stayCount><!--Required; The number of stay targets.-->
  <person type="uint32"><!--Optional.--></person>
  <car type="uint32"><!--Optional.--></car>
  <bike type="uint32"><!--Optional.--></bike>
</stayCount>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="utf-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10" >
  <entranceCount>
    <person type="uint32">100</person>
    <car type="uint32">200</car>
    <bike type="uint32">100</bike>
  </entranceCount>
  <exitCount>
    <person type="uint32">100</person>
    <car type="uint32">100</car>
    <bike type="uint32">100</bike>
  </exitCount>
  <stayCount>
    <person type="uint32">0</person>
    <car type="uint32">10</car>

```

```

    <bike type="uint32">0</bike>
  </stayCount>
</config>

```

[Tips]:

10.11 Video Metadata Detection

10.11.1 GetSmartVsdConfig

GetSmartVsdConfig	
Description	To get the IP media device's "vsd" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartVsdConfig[/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	<vsd>
Applicable products	IPC,NVR
<p><vsd> XML Block:</p> <pre> <?xml version="1.0" encoding="UTF-8"?> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <countCycleType><!--Required; Define the period for statistical reset.--> <enum>day</enum> <enum>week</enum> <enum>month</enum> <enum>off</enum> </countCycleType> </pre>	

```

</types>
<vsd><!--Required; vsd configurations.-->
  <switch type="boolean"><!--Optional; boolean,traffic count is enabled or not.--></switch>
  <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
    <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility
    with older versions in history. If you are using it for the first time, please ignore this
    element.-->
  </objectFilter>
  <saveTargetPicture type="boolean">
    <!--Optional; boolean,save the target picture to Local storage or not.-->
  </saveTargetPicture>
  <saveSourcePicture type="boolean">
    <!--Optional; boolean,save the source picture to Local storage or not.-->
  </saveSourcePicture>
  <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
    element with the same name in the API interface "GetSmartPerimeterConfig". -->
  </boundary>
  <maskArea type="list" maxcount="" count=""><!--Optional; Definition of areas that do not
    require detection.-->
    <item>
      <pointGroup type="list" minCount="" maxCount="" count=""><!--Required;
        Coordinates of the mask area. Specifically, when the number of points is 0, it
        means clearing the mask area.-->
        <item>
          <X type="uint32"><!--Required;X-coordinate--></X>
          <Y type="uint32"><!--Required;Y-coordinate--></Y>
        </item>
      </pointGroup>
    </item>
  </maskArea>
  <osdConfig><!--Optional;OSD configuration.-->
    <personcfg><!--Optional; OSD configuration for the target 'person' attribute.-->
      <sexSwitch type="boolean" index="">
        <!--Optional; Display 'sex' or not. The attribute 'index' indicates the order in which
        the attributes are displayed on the OSD. From left to right and top to bottom, the
        index increases. -->

```

```
</sexSwitch>
```

```
<ageSwitch type="boolean" index="" >
```

```
  <!--Optional; Display 'age' or not. The attribute 'index' indicates the order in
  which the attributes are displayed on the OSD. From left to right and top to bottom,
  the index increases. -->
```

```
</ageSwitch>
```

```
<orientationSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'orientation' or not. The attribute 'index' indicates the order in
  which the attributes are displayed on the OSD. From left to right and top to bottom,
  the index increases. -->
```

```
</orientationSwitch>
```

```
<hatSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'hat' or not. The attribute 'index' indicates the order in which
  the attributes are displayed on the OSD. From left to right and top to bottom, the
  index increases. -->
```

```
</hatSwitch>
```

```
<glassesSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'glasses' or not. The attribute 'glasses' indicates the order in
  which the attributes are displayed on the OSD. From left to right and top to
  bottom, the index increases. -->
```

```
</glassesSwitch>
```

```
<backpackSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'backpack' or not. The attribute 'index' indicates the order in
  which the attributes are displayed on the OSD. From left to right and top to bottom,
  the index increases. -->
```

```
</backpackSwitch>
```

```
<shortsleevesSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'shortsleeves' or not. The attribute 'index' indicates the order
  in which the attributes are displayed on the OSD. From left to right and top to
  bottom, the index increases. -->
```

```
</shortsleevesSwitch>
```

```
<upperbodycolorSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'upperbodycolor' or not. The attribute 'index' indicates the
  order in which the attributes are displayed on the OSD. From left to right and top to
  bottom, the index increases. -->
```

```
</upperbodycolorSwitch>
```

```
<shortsSwitch type="boolean" index="">
```

```
  <!--Optional; Display 'shorts' or not. The attribute 'index' indicates the order in
  which the attributes are displayed on the OSD. From left to right and top to bottom,
```

```
        the index increases. -->
    </shortsSwitch>
    <lowerbodycolorSwitch type="boolean" index="">
        <!--Optional; Display 'lowerbodycolor' or not. The attribute 'index' indicates the
        order in which the attributes are displayed on the OSD. From left to right and top to
        bottom, the index increases. -->
    </lowerbodycolorSwitch>
    <skirtSwitch type="boolean" index="">
        <!--Optional; Display 'skirt' or not. The attribute 'index' indicates the order in which
        the attributes are displayed on the OSD. From left to right and top to bottom, the
        index increases. -->
    </skirtSwitch>
    <maskSwitch type="boolean" index="">
        <!--Optional; Display 'mask' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
    </maskSwitch>
    <shoulderbagSwitch type="boolean" index="">
        <!--Optional; Display 'shoulderbag' or not. The attribute 'index' indicates the order
        in which the attributes are displayed on the OSD. From left to right and top to
        bottom, the index increases. -->
    </shoulderbagSwitch>
</personcfg>
<carcfg><!--Optional; OSD configuration for the target 'car' attribute.-->
    <colorSwitch type="boolean" index="">
        <!--Optional; Display 'color' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
    </colorSwitch>
    <categorySwitch type="boolean" index="">
        <!--Optional; Display 'category' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
    </categorySwitch>
    <brandSwitch type="boolean" index="">
        <!--Optional; Display 'brand' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
```

```

    </brandSwitch>
    <modelSwitch type="boolean" index="">
        <!--Optional; Display 'model' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
    </modelSwitch>
</carcfg>
<bikecfg><!--Optional; OSD configuration for the target 'bike' attribute.-->
    <bikeTypeSwitch type="boolean" index="">
        <!--Optional; Display 'bikeType' or not. The attribute 'index' indicates the order in
        which the attributes are displayed on the OSD. From left to right and top to bottom,
        the index increases. -->
    </bikeTypeSwitch>
</bikecfg>
</osdConfig>
</vsd>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <countCycleType>
            <enum>day</enum>
            <enum>week</enum>
            <enum>month</enum>
            <enum>off</enum>
        </countCycleType>
    </types>
    <vsd>
        <switch type="boolean">true</switch>
        <objectFilter>
            <person>
                <switch type="boolean">true</switch>
            </person>
        </objectFilter>
    </vsd>
</config>

```

```
<sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
<minDetectTarget>
  <width type="uint32" min="100" max="10000" default="300">100</width>
  <height type="uint32" min="100" max="10000" default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="9000">9000</width>
  <height type="uint32" min="100" max="10000" default="9000">9000</height>
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="300">100</width>
    <height type="uint32" min="100" max="10000" default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="9000">9000</width>
    <height type="uint32" min="100" max="10000" default="9000">9000</height>
```

```
</maxDetectTarget>

</motor>

</objectFilter>

<saveTargetPicture type="boolean">>false</saveTargetPicture>

<saveSourcePicture type="boolean">>false</saveSourcePicture>

<boundary type="list" maxCount="4" count="4">

  <item>

    <objectFilter>

      <person>

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

        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>

        <minDetectTarget>

          <width type="uint32" min="100" max="10000"
            default="300">100</width>

          <height type="uint32" min="100" max="10000"
            default="600">100</height>

        </minDetectTarget>

        <maxDetectTarget>

          <width type="uint32" min="100" max="10000"
            default="9000">9000</width>

          <height type="uint32" min="100" max="10000"
            default="9000">9000</height>

        </maxDetectTarget>

      </person>

      <car>

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

        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>

        <minDetectTarget>

          <width type="uint32" min="100" max="10000"
            default="300">100</width>

          <height type="uint32" min="100" max="10000"
            default="600">100</height>

        </minDetectTarget>

      </car>

    </item>

  </boundary>

</saveSourcePicture>

</saveTargetPicture>

</objectFilter>

</motor>

</maxDetectTarget>
```

```
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="4">
  <item >
    <X type="uint32">23</X>
    <Y type="uint32">0</Y>
  </item>
  <item >
    <X type="uint32">9880</X>
    <Y type="uint32">158</Y>
  </item>
  <item >
```

```
<X type="uint32">9904</X>
  <Y type="uint32">9873</Y>
</item>
<item >
  <X type="uint32">0</X>
  <Y type="uint32">9841</Y>
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
        <height type="uint32" max="10000" min="100"
          default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="9000">9000</width>
        <height type="uint32" max="10000" min="100"
          default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" max="100" min="1"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" max="10000" min="100"
          default="300">100</width>
```



```
<minDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="300">100</width>
  <height type="uint32" min="100" max="10000"
    default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</person>
<car>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
```

```
<height type="uint32" min="100" max="10000"
      default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
        default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
        default="9000">9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
            default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
              default="300">100</width>
        <height type="uint32" min="100" max="10000"
              default="600">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
              default="9000">9000</width>
        <height type="uint32" min="100" max="10000"
              default="9000">9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
            default="50">50</sensitivity>
```

```
<minDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="300">100</width>
  <height type="uint32" min="100" max="10000"
    default="600">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="9000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="9000">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100"
    default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="300">100</width>
    <height type="uint32" min="100" max="10000"
      default="600">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000"
      default="9000">9000</width>
    <height type="uint32" min="100" max="10000"
      default="9000">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
<maskArea type="list" maxCount="4" count="4">
  <item>
```

```
<pointGroup type="list" minCount="4" maxCount="6" count="4">
  <item >
    <X type="uint32">23</X>
    <Y type="uint32">0</Y>
  </item>
  <item >
    <X type="uint32">9880</X>
    <Y type="uint32">158</Y>
  </item>
  <item >
    <X type="uint32">9904</X>
    <Y type="uint32">9873</Y>
  </item>
  <item >
    <X type="uint32">0</X>
    <Y type="uint32">9841</Y>
  </item>
</pointGroup>
</item>
<item>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</maskArea>
<osdConfig>
  <personcfg>
    <sexSwitch type="boolean" index="0">true</sexSwitch>
```

```
<ageSwitch type="boolean" index="1" >true</ageSwitch>
<orientationSwitch type="boolean" index="2">true</orientationSwitch>
<hatSwitch type="boolean" index="3">true</hatSwitch>
<glassesSwitch type="boolean" index="4">true</glassesSwitch>
<backpackSwitch type="boolean" index="5">true</backpackSwitch>
<shortsleevesSwitch type="boolean" index="6">true</shortsleevesSwitch>
<upperbodycolorSwitch type="boolean" index="7">true</upperbodycolorSwitch>
<shortsSwitch type="boolean" index="8">true</shortsSwitch>
<lowerbodycolorSwitch type="boolean" index="9">true</lowerbodycolorSwitch>
<skirtSwitch type="boolean" index="10">true</skirtSwitch>
<maskSwitch type="boolean" index="11">true</maskSwitch>
<shoulderbagSwitch type="boolean" index="12">true</shoulderbagSwitch>
</personcfg>
<carcfg>
  <colorSwitch type="boolean" index="0">true</colorSwitch>
  <categorySwitch type="boolean" index="1">true</categorySwitch>
  <brandSwitch type="boolean" index="2">true</brandSwitch>
  <modelSwitch type="boolean" index="3">true</modelSwitch>
</carcfg>
<bikecfg>
  <bikeTypeSwitch type="boolean" index="0">true</bikeTypeSwitch>
</bikecfg>
</osdConfig>
</vsd>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.11.2 SetSmartVsdConfig

SetSmartVsdConfig

Description	To set the IP media device's "vsd" element.
Typical URL	POST http://<host>[:port]/SetSmartVsdConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<vsd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR

<vsd> XML Block:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="" xmlns="http://www.ipc.com/ver10">
  <vsd><!--Required; vsd configurations.-->
    <switch><!--Optional; boolean,traffic count is enabled or not.--></switch>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
      <objectFilter> definition in the <boundary> element for details. It is reserved for
      compatibility with older versions in history. If you are using it for the first time, please
      ignore this element.-->
    </objectFilter>
    <saveTargetPicture>
      <!--Optional; boolean,save the target picture to Local storage or not.-->
    </saveTargetPicture>
    <saveSourcePicture>
      <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <boundary><!--Optional; Filter parameters for detecting targets, Refer to the definition of the
      element with the same name in the API interface "SetSmartPerimeterConfig". -->
    </boundary>
    <maskArea><!--Optional; Definition of areas that do not require detection.-->
      <item>
        <pointGroup><!--Required; Coordinates of the mask area. Specifically, when the
          number of points is 0, it means clearing the mask area.-->
        <item><!--Optional.-->

```

```
<X><!--Required;X-coordinate--></X>
<Y><!--Required;Y-coordinate--></Y>
</item>
</pointGroup>
</item>
</maskArea>
<osdConfig><!--Optional;OSD configuration.-->
  <personcfg><!--Optional; OSD configuration for the target 'person' attribute.-->
    <sexSwitch index="">
      <!--Optional; Display 'sex' or not. The attribute 'index' indicates the order in
      which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
    </sexSwitch>
    <ageSwitch index="" >
      <!--Optional; Display 'age' or not. The attribute 'index' indicates the order in
      which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
    </ageSwitch>
    <orientationSwitch index="">
      <!--Optional; Display 'orientation' or not. The attribute 'index' indicates the order
      in which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
    </orientationSwitch>
    <hatSwitch index="">
      <!--Optional; Display 'hat' or not. The attribute 'index' indicates the order in
      which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
    </hatSwitch>
    <glassesSwitch index="">
      <!--Optional; Display 'glasses' or not. The attribute 'glasses' indicates the order
      in which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
    </glassesSwitch>
    <backpackSwitch index="">
      <!--Optional; Display 'backpack' or not. The attribute 'glasses' indicates the order
      in which the attributes are displayed on the OSD. From left to right and top to
      bottom, the index increases. -->
```

```
</backpackSwitch>
```

```
<shortsleevesSwitch index="">
```

```
<!--Optional; Display 'shortsleeves' or not. The attribute 'glasses' indicates the
order in which the attributes are displayed on the OSD. From left to right and top
to bottom, the index increases. -->
```

```
</shortsleevesSwitch>
```

```
<upperbodycolorSwitch index="">
```

```
<!--Optional; Display 'upperbodycolor' or not. The attribute 'glasses' indicates the
order in which the attributes are displayed on the OSD. From left to right and top
to bottom, the index increases. -->
```

```
</upperbodycolorSwitch>
```

```
<shortsSwitch index="">
```

```
<!--Optional; Display 'shorts' or not. The attribute 'glasses' indicates the order in
which the attributes are displayed on the OSD. From left to right and top to
bottom, the index increases. -->
```

```
</shortsSwitch>
```

```
<lowerbodycolorSwitch index="">
```

```
<!--Optional; Display 'lowerbodycolor' or not. The attribute 'glasses' indicates the
order in which the attributes are displayed on the OSD. From left to right and top
to bottom, the index increases. -->
```

```
</lowerbodycolorSwitch>
```

```
<skirtSwitch index="">
```

```
<!--Optional; Display 'skirt' or not. The attribute 'glasses' indicates the order in
which the attributes are displayed on the OSD. From left to right and top to
bottom, the index increases. -->
```

```
</skirtSwitch>
```

```
<maskSwitch index="">
```

```
<!--Optional; Display 'mask' or not. The attribute 'glasses' indicates the order in
which the attributes are displayed on the OSD. From left to right and top to
bottom, the index increases. -->
```

```
</maskSwitch>
```

```
<shoulderbagSwitch index="">
```

```
<!--Optional; Display 'shoulderbag' or not. The attribute 'glasses' indicates the
order in which the attributes are displayed on the OSD. From left to right and top
to bottom, the index increases. -->
```

```
</shoulderbagSwitch>
```

```
</personcfg>
```

```
<carcfg><!--Optional; OSD configuration for the target 'car' attribute.-->
```

```
<colorSwitch index="">
    <!--Optional; Display 'color' or not. The attribute 'glasses' indicates the order in
    which the attributes are displayed on the OSD. From left to right and top to
    bottom, the index increases. -->

</colorSwitch>

<categorySwitch index="">
    <!--Optional; Display 'category' or not. The attribute 'glasses' indicates the order
    in which the attributes are displayed on the OSD. From left to right and top to
    bottom, the index increases. -->

</categorySwitch>

<brandSwitch index="">
    <!--Optional; Display 'brand' or not. The attribute 'glasses' indicates the order in
    which the attributes are displayed on the OSD. From left to right and top to
    bottom, the index increases. -->

</brandSwitch>

<modelSwitch index="">
    <!--Optional; Display 'model' or not. The attribute 'glasses' indicates the order in
    which the attributes are displayed on the OSD. From left to right and top to
    bottom, the index increases. -->

</modelSwitch>

</carcfg>

<bikecfg><!--Optional; OSD configuration for the target 'bike' attribute.-->

    <bikeTypeSwitch index="">
        <!--Optional; Display 'bikeType' or not. The attribute 'glasses' indicates the order
        in which the attributes are displayed on the OSD. From left to right and top to
        bottom, the index increases. -->

    </bikeTypeSwitch>

</bikecfg>

</osdConfig>

</vsd>

</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <vsd>
        <switch>true</switch>
```

```
<objectFilter>
  <person>
    <switch>true</switch>
    <sensitivity>50</sensitivity>
    <minDetectTarget>
      <width>100</width>
      <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width>9000</width>
      <height>9000</height>
    </maxDetectTarget>
  </person>
  <car>
    <switch>true</switch>
    <sensitivity>50</sensitivity>
    <minDetectTarget>
      <width>100</width>
      <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width>9000</width>
      <height>9000</height>
    </maxDetectTarget>
  </car>
  <motor>
    <switch>true</switch>
    <sensitivity>50</sensitivity>
    <minDetectTarget>
      <width>100</width>
      <height>100</height>
    </minDetectTarget>
```

```
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture>>false</saveTargetPicture>
<saveSourcePicture>>false</saveSourcePicture>
<boundary>
  <item>
    <objectFilter>
      <person>
        <switch>>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </person>
      <car>
        <switch>>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
```

```
<height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup>
  <item >
    <X>23</X>
    <Y>0</Y>
  </item>
  <item >
    <X>9880</X>
    <Y>158</Y>
  </item>
  <item >
    <X>9904</X>
    <Y>9873</Y>
  </item>
  <item >
    <X>0</X>
    <Y>9841</Y>
  </item>
</pointGroup>
</objectFilter>
</car>
```

```
        </item>
    </pointGroup>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </person>
        <car>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </car>
        <motor>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
```

```
<minDetectTarget>
  <width>100</width>
  <height>100</height>
</minDetectTarget>
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
    <car>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
```

```
        </minDetectTarget>
        <maxDetectTarget>
            <width>9000</width>
            <height>9000</height>
        </maxDetectTarget>
    </car>
    <motor>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
            <width>100</width>
            <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
            <width>9000</width>
            <height>9000</height>
        </maxDetectTarget>
    </motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
```

```
<height>9000</height>
</maxDetectTarget>
</person>
<car>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>50</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
</boundary>
<maskArea>
```

```
<item>
  <pointGroup>
    <item >
      <X>23</X>
      <Y>0</Y>
    </item>
    <item >
      <X>9880</X>
      <Y>158</Y>
    </item>
    <item >
      <X>9904</X>
      <Y>9873</Y>
    </item>
    <item >
      <X>0</X>
      <Y>9841</Y>
    </item>
  </pointGroup>
</item>
<item>
  <pointGroup></pointGroup>
</item>
<item>
  <pointGroup></pointGroup>
</item>
<item>
  <pointGroup></pointGroup>
</item>
</maskArea>
<osdConfig>
  <personcfg>
```

```
<sexSwitch index="0">true</sexSwitch>
<ageSwitch index="1" >true</ageSwitch>
<orientationSwitch index="2">true</orientationSwitch>
<hatSwitch index="3">true</hatSwitch>
<glassesSwitch index="4">true</glassesSwitch>
<backpackSwitch index="5">true</backpackSwitch>
<shortsleevesSwitch index="6">true</shortsleevesSwitch>
<upperbodycolorSwitch index="7">true</upperbodycolorSwitch>
<shortsSwitch index="8">true</shortsSwitch>
<lowerbodycolorSwitch index="9">true</lowerbodycolorSwitch>
<skirtSwitch type="boolean" index="10">true</skirtSwitch>
<maskSwitch index="11">true</maskSwitch>
<shoulderbagSwitch index="12">true</shoulderbagSwitch>
</personcfg>
<carcfg>
  <colorSwitch index="0">true</colorSwitch>
  <categorySwitch index="1">true</categorySwitch>
  <brandSwitch index="2">true</brandSwitch>
  <modelSwitch index="3">true</modelSwitch>
</carcfg>
<bikecfg>
  <bikeTypeSwitch index="0">true</bikeTypeSwitch>
</bikecfg>
</osdConfig>
</vsd>
</config>
```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.12 Illegal Parking Detection

10.12.1 GetSmartPvdConfig

GetSmartPvdConfig	
Description	To get the IP media device's "pvd"(Illegal Parking Detection) element.
Typical URL	POST or GET http://<host>[:port]/GetSmartPvdConfig[/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	<pvd>
Applicable products	IPC,NVR
<pvd> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <pvd><!--Required; pvd configurations.--> <switch type="boolean"><!--Optional; boolean,detection is enabled or not.--></switch> <duration type="uint32" min="" max=""> <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.--> </duration> <alarmHoldTime type="uint32" min="" max=""> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <saveTargetPicture type="boolean"> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture type="boolean">	

```
<!--Optional; boolean,save the source picture to Local storage or not.-->
</saveSourcePicture>
<boundary type="list" maxCount="" count=""><!--Optional; Detection parameters for each
detection region.-->
  <item>
    <objectFilter><!--Optional; Filter parameters for detecting targets.-->
      <car><!--Optional; car filter parameters.-->
        <switch type="boolean"><!--Required;Enabled or not.--></switch>
        <sensitivity type="uint32" min="" max="" default="">
          <!--Required;Detection sensitivity.-->
        </sensitivity>
        <minDetectTarget><!--Optional; The minimum target size that needs to be
detected.-->
          <width type="uint32" min="" max="" default="">
            <!--Required;Minimum target width.-->
          </width>
          <height type="uint32" min="" max="" default="">
            <!--Required;Minimum target height.-->
          </height>
        </minDetectTarget>
        <maxDetectTarget><!--Optional; The maximum target size that needs to be
detected.-->
          <width type="uint32" min="" max="" default="">
            <!--Required;Minimum target width.-->
          </width>
          <height type="uint32" min="" max="" default="">
            <!--Required;Minimum target height.-->
          </height>
        </maxDetectTarget>
      </car>
      <motor><!--Optional; moter filter parameters, refer to element <car>.-->
        <switch type="boolean"></switch>
        <sensitivity type="uint32" min="" max="" default=""></sensitivity>
```

```

        <minDetectTarget>
            <width type="uint32" min="" max="" default=""></width>
            <height type="uint32" min="" max="" default=""></height>
        </minDetectTarget>
        <maxDetectTarget>
            <width type="uint32" min="" max="" default=""></width>
            <height type="uint32" min="" max="" default=""></height>
        </maxDetectTarget>
    </motor>
</objectFilter>
<pointGroup type="list" minCount="" maxCount="" count=""><!--Required;
Coordinates of the detection area. Specifically, when count is 0, it indicates that no
detection area has been set.-->
    <item><!--Optional.-->
        <X type="uint32"><!--Required;X-coordinate--></X>
        <Y type="uint32"><!--Required;Y-coordinate--></Y>
    </item>
</pointGroup>
</item>
</boundary>
</pvd>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <pvd>
        <switch type="boolean">>false</switch>
        <duration type="uint32" min="10" max="3600">10</duration>
        <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
        <objectFilter>
            <car>
                <switch type="boolean">>true</switch>
                <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
            </car>
        </objectFilter>
    </pvd>
</config>

```

```
<minDetectTarget>
  <width type="uint32" min="100" max="10000" default="972">100</width>
  <height type="uint32" min="100" max="10000" default="972">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="5000">9000</width>
  <height type="uint32" min="100" max="10000" default="6111">9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch type="boolean">true</switch>
  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
  <minDetectTarget>
    <width type="uint32" min="100" max="10000" default="972">100</width>
    <height type="uint32" min="100" max="10000" default="972">100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width type="uint32" min="100" max="10000" default="5000">9000</width>
    <height type="uint32" min="100" max="10000" default="6111">9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<saveTargetPicture type="boolean">>false</saveTargetPicture>
<saveSourcePicture type="boolean">>false</saveSourcePicture>
<boundary type="list" maxCount="4" count="4">
  <item>
    <objectFilter>
      <car>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="972">100</width>
```

```
<height type="uint32" min="100" max="10000"
  default="972">100</height>

</minDetectTarget>

<maxDetectTarget>

  <width type="uint32" min="100" max="10000"
    default="5000">9000</width>

  <height type="uint32" min="100" max="10000"
    default="6111">9000</height>

</maxDetectTarget>

</car>

<motor>

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

  <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>

  <minDetectTarget>

    <width type="uint32" min="100" max="10000"
      default="972">100</width>

    <height type="uint32" min="100" max="10000"
      default="972">100</height>

  </minDetectTarget>

  <maxDetectTarget>

    <width type="uint32" min="100" max="10000"
      default="5000">9000</width>

    <height type="uint32" min="100" max="10000"
      default="6111">9000</height>

  </maxDetectTarget>

</motor>

</objectFilter>

<pointGroup type="list" minCount="4" maxCount="6" count="4">

  <item>

    <X type="uint32">0</X>

    <Y type="uint32">0</Y>

  </item>

  <item>

    <X type="uint32">500</X>

    <Y type="uint32">0</Y>
```

```
</item>
<item>
  <X type="uint32">0</X>
  <Y type="uint32">500</Y>
</item>
<item>
  <X type="uint32">500</X>
  <Y type="uint32">500</Y>
</item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="972">100</width>
        <height type="uint32" min="100" max="10000"
          default="972">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="5000">9000</width>
        <height type="uint32" min="100" max="10000"
          default="6111">9000</height>
      </maxDetectTarget>
    </car>
    <motor>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
```

```
        default="972">100</width>
        <height type="uint32" min="100" max="10000"
            default="972">100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
            default="5000">9000</width>
        <height type="uint32" min="100" max="10000"
            default="6111">9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
    <objectFilter>
        <car>
            <switch type="boolean">true</switch>
            <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
            <minDetectTarget>
                <width type="uint32" min="100" max="10000"
                    default="972">100</width>
                <height type="uint32" min="100" max="10000"
                    default="972">100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width type="uint32" min="100" max="10000"
                    default="5000">9000</width>
                <height type="uint32" min="100" max="10000"
                    default="6111">9000</height>
            </maxDetectTarget>
        </car>
    </objectFilter>
    <motor>
        <switch type="boolean">true</switch>
        <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
```

```
<minDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="972">100</width>
  <height type="uint32" min="100" max="10000"
    default="972">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="5000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="6111">9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <car>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="972">100</width>
        <height type="uint32" min="100" max="10000"
          default="972">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="5000">9000</width>
        <height type="uint32" min="100" max="10000"
          default="6111">9000</height>
      </maxDetectTarget>
    </car>
  </objectFilter>
</item>
</motor>
```

```

<switch type="boolean">true</switch>
<sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
<minDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="972">100</width>
  <height type="uint32" min="100" max="10000"
    default="972">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000"
    default="5000">9000</width>
  <height type="uint32" min="100" max="10000"
    default="6111">9000</height>
</maxDetectTarget>
</motor>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
</pvd>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.12.2 SetSmartPvdConfig

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

Entity Data	<pvd>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><pvd> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <pvd><!--Required; pvd configurations.--> <switch><!--Optional; boolean,detection is enabled or not.--></switch> <duration> <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.--> </duration> <alarmHoldTime> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <saveTargetPicture> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture> <boundary><!--Optional; Detection parameters for each detection region.--> <item> <objectFilter><!--Optional; Filter parameters for detecting targets.--> <car><!--Optional; car filter parameters.--> <switch><!--Required;boolean,Enabled or not.--></switch> <sensitivity> <!--Required;uint32,Detection sensitivity.--> </sensitivity> </car> </item> </boundary> </pvd> </config> </pre>	

```
</sensitivity>
<minDetectTarget><!--Optional; The minimum target size that needs to be
detected.-->
    <width><!--Required;Minimum target width.--></width>
    <height><!--Required;Minimum target height.--></height>
</minDetectTarget>
<maxDetectTarget><!--Optional; The maximum target size that needs to be
detected.-->
    <width><!--Required;Minimum target width.--></width>
    <height><!--Required;Minimum target height.--></height>
</maxDetectTarget>
</car>
<motor><!--Optional; moter filter parameters, refer to element <car>.-->
    <switch></switch>
    <sensitivity></sensitivity>
    <minDetectTarget>
        <width></width>
        <height></height>
    </minDetectTarget>
    <maxDetectTarget>
        <width></width>
        <height></height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup><!--Required; list,Coordinates of the detection area. Specifically, when
the count of points is 0, it means clearing the detection area.-->
    <item><!--Optional.-->
        <X><!--Required:uint32,X-coordinate--></X>
        <Y><!--Required:uint32,Y-coordinate--></Y>
    </item>
</pointGroup>
</item>
```

```
</boundary>
</pvd>
</config>
```

Example of Entity Data:

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <pvd>
    <switch>true</switch>
    <duration>10</duration>
    <alarmHoldTime>1</alarmHoldTime>
    <saveTargetPicture>true</saveTargetPicture>
    <saveSourcePicture>true</saveSourcePicture>
    <boundary>
      <item>
        <objectFilter>
          <car>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
              <width>100</width>
              <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
              <width>9000</width>
              <height>9000</height>
            </maxDetectTarget>
          </car>
          <motor>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
              <width>100</width>
              <height>100</height>
            </minDetectTarget>
          </motor>
        </objectFilter>
      </item>
    </boundary>
  </pvd>
</config>
```

```
        </minDetectTarget>
        <maxDetectTarget>
            <width>9000</width>
            <height>9000</height>
        </maxDetectTarget>
    </motor>
</objectFilter>
<pointGroup>
    <item>
        <X>2175</X>
        <Y>3000</Y>
    </item>
    <item>
        <X>4825</X>
        <Y>7000</Y>
    </item>
    <item>
        <X>5850</X>
        <Y>3600</Y>
    </item>
    <item>
        <X>4575</X>
        <Y>1700</Y>
    </item>
</pointGroup>
</item>
<item>
    <objectFilter>
        <car>
            <switch>true</switch>
            <sensitivity>60</sensitivity>
            <minDetectTarget>
```

```
        <width>100</width>
        <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
    </maxDetectTarget>
</car>
<motor>
    <switch>true</switch>
    <sensitivity>60</sensitivity>
    <minDetectTarget>
        <width>100</width>
        <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
    </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
    <objectFilter>
        <car>
            <switch>true</switch>
            <sensitivity>70</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
```

```
<maxDetectTarget>
  <width>9000</width>
  <height>9000</height>
</maxDetectTarget>
</car>
<motor>
  <switch>true</switch>
  <sensitivity>70</sensitivity>
  <minDetectTarget>
    <width>100</width>
    <height>100</height>
  </minDetectTarget>
  <maxDetectTarget>
    <width>9000</width>
    <height>9000</height>
  </maxDetectTarget>
</motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
<item>
  <objectFilter>
    <car>
      <switch>true</switch>
      <sensitivity>80</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
```

```
        </maxDetectTarget>
    </car>
    <motor>
        <switch>true</switch>
        <sensitivity>80</sensitivity>
        <minDetectTarget>
            <width>100</width>
            <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
            <width>9000</width>
            <height>9000</height>
        </maxDetectTarget>
    </motor>
</objectFilter>
<pointGroup></pointGroup>
</item>
</boundary>
</pvd>
</config>
```

- [Tips]:**
1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
 2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.13 Loitering Detection

10.13.1 GetSmartLoiteringConfig

GetSmartLoiteringConfig

Description	To get the IP media device's "loitering" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartLoiteringConfig[/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	<loitering>
Applicable products	IPC,NVR

<loitering> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <triggerModeE><!--Required; Define the mode of trigger.-->
      <enum>movingObject</enum><!--Required; Only detect moving objects.-->
      <enum>allObject</enum><!--Required; Detect all objects, including moving and stationary objects.-->
    </triggerModeE>
  </types>
  <loitering><!--Required; loitering configurations.-->
    <switch type="boolean"><!--Optional; boolean,detection is enabled or not.--></switch>
    <triggerMode type="triggerModeE">
      <!--Optional; Detect moving objects or all objects (moving objects, stationary objects).-->
    </triggerMode>
    <duration type="uint32" min="" max="">
      <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.-->
    </duration>
    <alarmHoldTime type="uint32" min="" max="">
      <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the
    <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility
    with older versions in history. If you are using it for the first time, please ignore this
  
```

```

    element.-->
</objectFilter>
<saveTargetPicture type="boolean">
    <!--Optional; boolean,save the target picture to Local storage or not.-->
</saveTargetPicture>
<saveSourcePicture type="boolean">
    <!--Optional; boolean,save the source picture to Local storage or not.-->
</saveSourcePicture>
<boundary type="list" maxCount="" count=""><!--Optional; Detection parameters for each
detection region.-->
    <item>
        <objectFilter><!--Optional; Filter parameters for detecting targets.-->
            <person><!--Optional; person filter parameters.-->
                <switch type="boolean"><!--Required;Enabled or not.--></switch>
                <sensitivity type="uint32" min="" max="" default="">
                    <!--Required;Detection sensitivity.-->
                </sensitivity>
                <minDetectTarget><!--Optional; The minimum target size that needs to be
detected.-->
                    <width type="uint32" min="" max="" default="">
                        <!--Required;Minimum target width.-->
                    </width>
                    <height type="uint32" min="" max="" default="">
                        <!--Required;Minimum target height.-->
                    </height>
                </minDetectTarget>
                <maxDetectTarget><!--Optional; The maximum target size that needs to be
detected.-->
                    <width type="uint32" min="" max="" default="">
                        <!--Required;Minimum target width.-->
                    </width>
                    <height type="uint32" min="" max="" default="">
                        <!--Required;Minimum target height.-->
                    </height>
                </maxDetectTarget>
            </person>
        </objectFilter>
    </item>
</boundary>

```

```

        </height>
        </maxDetectTarget>
    </person>
</objectFilter>
<pointGroup type="list" minCount="" maxCount="" count=""><!--Required;
Coordinates of the detection area. Specifically, when count is 0, it indicates that no
detection area has been set.-->
    <item><!--Optional.-->
        <X type="uint32"><!--Required;X-coordinate--></X>
        <Y type="uint32"><!--Required;Y-coordinate--></Y>
    </item>
</pointGroup>
</item>
</boundary>
</loitering>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <triggerModeE>
            <enum>movingObject</enum>
            <enum>allObject</enum>
        </triggerModeE>
    </types>
    <loitering>
        <switch type="boolean">>false</switch>
        <triggerMode type="triggerModeE">movingObject</triggerMode>
        <duration type="uint32" min="10" max="3600">10</duration>
        <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
        <objectFilter>
            <person>
                <switch type="boolean">>true</switch>
            </person>
        </objectFilter>
    </loitering>
</config>

```

```
<sensitivity type="uint32" min="1" max="100" default="50">50</sensitivity>
<minDetectTarget>
  <width type="uint32" min="100" max="10000" default="972">100</width>
  <height type="uint32" min="100" max="10000" default="972">100</height>
</minDetectTarget>
<maxDetectTarget>
  <width type="uint32" min="100" max="10000" default="5000">9000</width>
  <height type="uint32" min="100" max="10000" default="6111">9000</height>
</maxDetectTarget>
</person>
</objectFilter>
<saveTargetPicture type="boolean">>false</saveTargetPicture>
<saveSourcePicture type="boolean">>false</saveSourcePicture>
<boundary type="list" maxCount="4" count="4">
  <item>
    <objectFilter>
      <person>
        <switch type="boolean">>true</switch>
        <sensitivity type="uint32" min="1" max="100"
          default="50">50</sensitivity>
        <minDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="972">100</width>
          <height type="uint32" min="100" max="10000"
            default="972">100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width type="uint32" min="100" max="10000"
            default="5000">9000</width>
          <height type="uint32" min="100" max="10000"
            default="6111">9000</height>
        </maxDetectTarget>
      </person>
    </objectFilter>
  </item>
</boundary>
```

```
<pointGroup type="list" minCount="4" maxCount="6" count="4">
  <item>
    <X type="uint32">0</X>
    <Y type="uint32">0</Y>
  </item>
  <item>
    <X type="uint32">500</X>
    <Y type="uint32">0</Y>
  </item>
  <item>
    <X type="uint32">0</X>
    <Y type="uint32">500</Y>
  </item>
  <item>
    <X type="uint32">500</X>
    <Y type="uint32">500</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="972">100</width>
        <height type="uint32" min="100" max="10000"
          default="972">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="5000">9000</width>
      </maxDetectTarget>
    </person>
  </objectFilter>
</item>
```

```
<height type="uint32" min="100" max="10000"
  default="6111">9000</height>
</maxDetectTarget>
</person>
</objectFilter>
<pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
      <minDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="972">100</width>
        <height type="uint32" min="100" max="10000"
          default="972">100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width type="uint32" min="100" max="10000"
          default="5000">9000</width>
        <height type="uint32" min="100" max="10000"
          default="6111">9000</height>
      </maxDetectTarget>
    </person>
  </objectFilter>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch type="boolean">true</switch>
      <sensitivity type="uint32" min="1" max="100"
        default="50">50</sensitivity>
```

```

    <minDetectTarget>
      <width type="uint32" min="100" max="10000"
        default="972">100</width>
      <height type="uint32" min="100" max="10000"
        default="972">100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width type="uint32" min="100" max="10000"
        default="5000">9000</width>
      <height type="uint32" min="100" max="10000"
        default="6111">9000</height>
    </maxDetectTarget>
  </person>
</objectFilter>
  <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
</loitering>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.13.2 SetSmartLoiteringConfig

SetSmartLoiteringConfig	
Description	To set the IP media device's "loitering" element.
Typical URL	POST http://<host>[:port]/SetSmartLoiteringConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<loitering>

Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><loitering> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <loitering><!--Required; loitering configurations.--> <switch><!--Optional; boolean,detection is enabled or not.--></switch> <triggerMode> <!--Optional; triggerModeE,Detect moving objects or all objects (moving objects, stationary objects).--> </triggerMode> <duration> <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.--> </duration> <alarmHoldTime> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <objectFilter><!--Optional; Filter parameters for detecting targets, please refer to the <objectFilter> definition in the <boundary> element for details. It is reserved for compatibility with older versions in history. If you are using it for the first time, please ignore this element.--> </objectFilter> <saveTargetPicture> <!--Optional; boolean,save the target picture to Local storage or not.--> </saveTargetPicture> <saveSourcePicture> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture> <boundary><!--Optional; Detection parameters for each detection region.--> <item> <objectFilter><!--Optional; Filter parameters for detecting targets.--> <person><!--Optional; person filter parameters.--> <switch><!--Required;boolean,Enabled or not.--></switch> </pre>	

```
<sensitivity>
    <!--Required:uint32,detection sensitivity.-->
</sensitivity>
<minDetectTarget><!--Optional; The minimum target size that needs to be
detected.-->
    <width><!--Required:uint32,Minimum target width.--></width>
    <height><!--Required:uint32,Minimum target height.--></height>
</minDetectTarget>
<maxDetectTarget><!--Optional; The maximum target size that needs to be
detected.-->
    <width><!--Required:uint32,Minimum target width.--></width>
    <height><!--Required:uint32,Minimum target height.--></height>
</maxDetectTarget>
</person>
</objectFilter>
<pointGroup><!--Required; Coordinates of the detection area. Specifically, when the
count of points is 0, it means clearing the detection area.-->
    <item><!--Optional.-->
        <X><!--Required:uint32,X-coordinate--></X>
        <Y><!--Required:uint32,X-coordinate--></Y>
    </item>
</pointGroup>
</item>
</boundary>
</loitering>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <loitering>
        <switch>>false</switch>
        <triggerMode>movingObject</triggerMode>
        <duration>10</duration>
```

```
<alarmHoldTime>20</alarmHoldTime>
<objectFilter>
  <person>
    <switch>true</switch>
    <sensitivity>50</sensitivity>
    <minDetectTarget>
      <width>100</width>
      <height>100</height>
    </minDetectTarget>
    <maxDetectTarget>
      <width>9000</width>
      <height>9000</height>
    </maxDetectTarget>
  </person>
</objectFilter>
<saveTargetPicture>>false</saveTargetPicture>
<saveSourcePicture>>false</saveSourcePicture>
<boundary>
  <item>
    <objectFilter>
      <person>
        <switch>true</switch>
        <sensitivity>50</sensitivity>
        <minDetectTarget>
          <width>100</width>
          <height>100</height>
        </minDetectTarget>
        <maxDetectTarget>
          <width>9000</width>
          <height>9000</height>
        </maxDetectTarget>
      </person>
    </objectFilter>
  </item>
</boundary>
```

```
</objectFilter>
<pointGroup>
  <item>
    <X>2175</X>
    <Y>3000</Y>
  </item>
  <item>
    <X>4825</X>
    <Y>7000</Y>
  </item>
  <item>
    <X>5850</X>
    <Y>3600</Y>
  </item>
  <item>
    <X>4575</X>
    <Y>1700</Y>
  </item>
</pointGroup>
</item>
<item>
  <objectFilter>
    <person>
      <switch>true</switch>
      <sensitivity>50</sensitivity>
      <minDetectTarget>
        <width>100</width>
        <height>100</height>
      </minDetectTarget>
      <maxDetectTarget>
        <width>9000</width>
        <height>9000</height>
      </maxDetectTarget>
    </person>
  </objectFilter>
</item>
```

```
        </maxDetectTarget>
    </person>
</objectFilter>
    <pointGroup></pointGroup>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
                <width>9000</width>
                <height>9000</height>
            </maxDetectTarget>
        </person>
    </objectFilter>
    <pointGroup></pointGroup>
</item>
<item>
    <objectFilter>
        <person>
            <switch>true</switch>
            <sensitivity>50</sensitivity>
            <minDetectTarget>
                <width>100</width>
                <height>100</height>
            </minDetectTarget>
            <maxDetectTarget>
```

```

        <width>9000</width>
        <height>9000</height>
    </maxDetectTarget>
</person>
</objectFilter>
<pointGroup></pointGroup>
</item>
</boundary>
</loitering>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.14 Crowd Gathering Detection

10.14.1 GetSmartCrowdGatheringConfig

GetSmartCrowdGatheringConfig	
Description	To get the IP media device's "CrowdGathering" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartCrowdGatheringConfig[/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	<crowdGathering>
Applicable products	IPC,NVR

<crowdGathering> XML Block:

```
<config version="" xmlns="http://www.ipc.com/ver10">
  <crowdGathering>
    <switch type="boolean"><!--Optional; boolean,detection is enabled or not.--></switch>
    <duration type="uint32" min="" max="">
      <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.-->
    </duration>
    <alarmHoldTime type="uint32" min="" max="">
      <!--Optional The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <saveSourcePicture type="boolean">
      <!--Optional; boolean,save the source picture to Local storage or not.-->
    </saveSourcePicture>
    <boundary type="list" maxCount="" count="">
      <item>
        <sensitivity type="uint32" min="" max="" default="">
          <!--Required;Detection sensitivity.-->
        </sensitivity>
        <pointGroup type="list" minCount="" maxCount="" count=""><!--Required;
          Coordinates of the detection area. Specifically, when count is 0, it indicates that no
          detection area has been set.-->
          <item><!--Optional.-->
            <X type="uint32"><!--Required;X-coordinate--></X>
            <Y type="uint32"><!--Required;Y-coordinate--></Y>
          </item>
        </pointGroup>
      </item>
    </boundary>
  </crowdGathering>
</config>
```

Example of Successful Response:

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

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <crowdGathering>
    <switch type="boolean">true</switch>
    <duration type="uint32" min="10" max="3600">10</duration>
    <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
    <saveSourcePicture type="boolean">>false</saveSourcePicture>
    <boundary type="list" maxCount="4" count="4">
      <item>
        <sensitivity type="uint32" min="1" max="100" default="60">60</sensitivity>
        <pointGroup type="list" minCount="4" maxCount="6" count="6">
          <item >
            <X type="uint32">1900</X>
            <Y type="uint32">3700</Y>
          </item>
          <item >
            <X type="uint32">1950</X>
            <Y type="uint32">7900</Y>
          </item>
          <item>
            <X type="uint32">6800</X>
            <Y type="uint32">7933</Y>
          </item>
          <item >
            <X type="uint32">6675</X>
            <Y type="uint32">3566</Y>
          </item>
          <item >
            <X type="uint32">4500</X>
            <Y type="uint32">4000</Y>
          </item>
          <item >
            <X type="uint32">1825</X>
```

```

        <Y type="uint32">3633</Y>
        </item>
    </pointGroup>
</item>
<item>
    <sensitivity type="uint32" min="1" max="100" default="60">60</sensitivity>
    <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
    <sensitivity type="uint32" min="1" max="100" default="60">60</sensitivity>
    <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
<item>
    <sensitivity type="uint32" min="1" max="100" default="60">60</sensitivity>
    <pointGroup type="list" minCount="4" maxCount="6" count="0"></pointGroup>
</item>
</boundary>
</crowdGathering>
</config>

```

[Tips]:

1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.

10.14.2 SetSmartCrowdGatheringConfig

SetSmartCrowdGatheringConfig	
Description	To set the IP media device's "Crowd Gathering" element.
Typical URL	POST http://<host>[:port]/SetSmartCrowdGatheringConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None

Entity Data	<crowdGathering>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><crowdGathering> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <crowdGathering> <switch><!--Optional; boolean,detection is enabled or not.--></switch> <duration> <!--Optional; Trigger an alarm after the event continues to trigger for a threshold duration.--> </duration> <alarmHoldTime> <!--Optional;uint32,The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> <saveSourcePicture "> <!--Optional; boolean,save the source picture to Local storage or not.--> </saveSourcePicture <boundary> <item> <sensitivity> <!--Required;uint32, Detection sensitivity.--> </sensitivity> <pointGroup><!--Required; Coordinates of the detection area. Specifically, when the count of points is 0, it means clearing the detection area.--> <item><!--Optional.--> <X><!--Required;uint32, X-coordinate--></X> <Y><!--Required;uint32, Y-coordinate--></Y> </item> </pointGroup> </item> </boundary> </pre>	

```
</crowdGathering>
```

```
</config>
```

Example of Entity Data:

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

```
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
```

```
<crowdGathering>
```

```
<switch>true</switch>
```

```
<duration>10</duration>
```

```
<alarmHoldTime>20</alarmHoldTime>
```

```
<saveSourcePicture>>false</saveSourcePicture>
```

```
<boundary>
```

```
<item>
```

```
<sensitivity>60</sensitivity>
```

```
<pointGroup>
```

```
<item >
```

```
<X>1900</X>
```

```
<Y>3700</Y>
```

```
</item>
```

```
<item >
```

```
<X>1950</X>
```

```
<Y>7900</Y>
```

```
</item>
```

```
<item >
```

```
<X>6800</X>
```

```
<Y>7933</Y>
```

```
</item>
```

```
<item >
```

```
<X>6675</X>
```

```
<Y>3566</Y>
```

```
</item>
```

```
<item >
```

```
<X>4500</X>
```

```
        <Y>4000</Y>
      </item>
    <item >
      <X>1825</X>
      <Y>3633</Y>
    </item>
  </pointGroup>
</item>
<item>
  <sensitivity>60</sensitivity>
  <pointGroup></pointGroup>
</item>
<item>
  <sensitivity>60</sensitivity>
  <pointGroup></pointGroup>
</item>
<item>
  <sensitivity>60</sensitivity>
  <pointGroup></pointGroup>
</item>
</boundary>
</crowdGathering>
</config>
```

- [Tips]:**
1. The "X" and "Y" element announce the horizontal and vertical position based in the 10000*10000 resolution. The width and height of minDetectTarget and maxDetectTarget are also based on a resolution of 10000 * 10000.
 2. Specifically, if you need to support multiple<boundary>, you need to set them all at once, and the subsequent operations will overwrite the previous ones.

10.15 Fire

10.15.1 GetSmartFireConfig

GetSmartFireConfig	
Description	To get the IP media device's "fire" element.
Typical URL	POST or GET http://<host>[:port]/GetSmartFireConfig[/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	<fire>
Applicable products	IPC,NVR
<fire> XML Block: <config version="" xmlns="http://www.ipc.com/ver10"> <fire> <switch type="boolean"><!--Required; boolean,detection is enabled or not.--></switch> <fireThreshold type="uint32" min="" max="" default=""> <!--Required; uint32, Sensitivity of detection.--> </fireThreshold> <alarmHoldTime type="uint32" min="" max=""> <!--Optional The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> </fire> </config>	
Example of Successful Response: <?xml version="1.0" encoding="UTF-8"?> <config version="2.0.0" xmlns="http://www.ipc.com/ver10"> <fire> <switch type="boolean">true</switch>	

```

    <fireThreshold type="uint32" min="1" max="100" default="50">50</fireThreshold>
    <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
  </fire>
</config>

```

[Tips]:

10.15.2 SetSmartFireConfig

SetSmartFireConfig	
Description	To set the IP media device's "fire" element.
Typical URL	POST http://<host>[:port]/SetSmartFireConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<fire>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC,NVR
<p><fire> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <fire> <switch><!--Optional; boolean,detection is enabled or not.--></switch> <fireThreshold> <!--Optional; uint32, Range[1~100], Sensitivity of detection.--> </fireThreshold> <alarmHoldTime> <!--Optional;The duration of the alarm after the event ends, Unit: seconds.--> </alarmHoldTime> </fire> </config> </pre>	
<p>Example of Entity Data:</p>	

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <fire>
    <switch>true</switch>
    <fireThreshold>50</fireThreshold>
    <alarmHoldTime>20</alarmHoldTime>
  </fire>
</config>

```

[Tips]:

10.16 Temperature Measurement

10.16.1 GetTemperatureAlarmConfig

GetTemperatureAlarmConfig	
Description	To get the IP media device's "temperatureAlarm" element.
Typical URL	POST or GET http://<host>[:port]/GetTemperatureAlarmConfig[/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	<temperatureAlarm>
Applicable products	IPC,NVR
<p><temperatureAlarm> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <types> <ruleShapeType><!--Required; Type of temperature measurement rules.--> <enum minCount="1" maxCount="1">point</enum> </pre>	

```

    <enum minCount="2" maxCount="2">line</enum>
    <enum minCount="4" maxCount="6">area</enum>
</ruleShapeType>
<alarmRuleType><!--Required;Temperature alarm rules.-->
    <enum>maxtemperabove</enum>
    <enum>maxtemperbelow</enum>
    <enum>mintemperabove</enum>
    <enum>mintemperbelow</enum>
    <enum>avgtemperabove</enum>
    <enum>avgtemperbelow</enum>
    <enum>difftemperabove</enum>
    <enum>difftemperbelow</enum>
</alarmRuleType>
<tempUnitType><!--Required; Define the unit of temperature.-->
    <enum>centigrade</enum>
    <enum>fahrenheit</enum>
</tempUnitType>
<distanceUnitType><!--Required; Define the unit of distance.-->
    <enum>meter</enum>
    <enum>foot</enum>
</distanceUnitType>
</types>
<temperatureAlarm><!--Required; detection configuration.-->
    <switch type="boolean"><!--Required; boolean,detection is enabled or not.--></switch>
    <alarmHoldTime type="uint32" min="" max="">
        <!--Required;The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <tempUnit type="tempUnitType">
        <!--Required;Unit of temperature.-->
    </tempUnit>
    <distanceUnit type="distanceUnitType"><!--Required; Unit of Distance.--></distanceUnits>
    <maxtemperen type="boolean">

```

```
<!--Required; Display the highest temperature or not.-->
</maxtemperen>
<avgtemperen type="boolean">
  <!--Required; Display average temperature or not.-->
</avgtemperen>
<mintemperen type="boolean">
  <!--Required; Display the lowest temperature or not.-->
</mintemperen>
<thermaldisplayen type="boolean">
  <!--Required; The thermal imaging channel displays temperature or not.-->
</thermaldisplayen>
<opticaldisplayen type="boolean">
  <!--Required; The optical channel displays temperature or not.-->
</opticaldisplayen>
<segcolorTemperatureParam><!--Optional; Temperature bar configuration.-->
  <switch type="boolean"><!--Required; Display temperature bar or not.--></switch>
</segcolorTemperatureParam>
<dotTemperatureInfo><!--Optional; Temperature measurement of specified 'point'.-->
  <switch type="boolean">
    <!--Required; Enable/Disable the temperature measurement function for the specified
    'point'.-->
  </switch>
  <emissivity type="uint32" min="" max="" default="">
    <!--Required; uint32, emissivity.-->
  </emissivity>
  <distance type="uint32" min="" max="" default="">
    <!--Required; uint32, distance.-->
  </distance>
  <reflectTemper type="int32" min="" max="" default="">
    <!--Required; Reflection temperature.-->
  </reflectTemper>
</dotTemperatureInfo>
<boundary type="list" maxCount="" count=""><!--Optional; Regions and rules for temperature
```

measurement.-->

<item>

<ruleId type="uint32">

<!--Required; uint32, Rule ID, starting from 0, less than the value of the property
maxCount.-->

</ruleId>

<switch type="boolean"><!--Required; boolean, enable or not.--></switch>

<ruleName type="string">

<!--Required; string, the name of rule.-->

</ruleName>

<emissivity type="uint32" min="" max="" default="">

<!--Required; uint32, emissivity.-->

</emissivity>

<distance type="uint32" min="" max="" default="">

<!--Required; uint32, distance.-->

</distance>

<reflectTemper type="int32" min="" max="" default="">

<!--Required; Reflection temperature.-->

</reflectTemper>

<alarmRule type="alarmRuleType">

<!--Required; Alarm rules.-->

</alarmRule>

<alarmTemper type="int32" min="" max="" default="">

<!--Required; Temperature threshold for alarm.-->

</alarmTemper>

<stayAlarmDelay type="uint32" min="" max="">

<!--Optional; The duration of temperature measurement reaching the alarm
condition. Unit: seconds.-->

</stayAlarmDelay>

<ruleType type="ruleShapeType"><!--Required; the type of rule.--></ruleType>

<pointGroup type="list" maxCount="" count=""><!--Required; Coordinates of the area
requiring temperature measurement.-->

<item><!--Optional.-->

```

        <X type="uint32"><!--Required;uint32,X-coordinate--></X>
        <Y type="uint32"><!--Required;uint32,Y-coordinate--></Y>
    </item>
</pointGroup>
<ioOut type="list" maxCount="" count=""><!--Optional; Linkage IO output after alarm
triggering.-->
    <item>
        <alarmId type="uint32"><!--Required;uint32,alarm out ID.--></alarmId>
        <switch type="boolean"><!--Required;boolean,enabled or not.--></switch>
    </item>
</ioOut>
<audio type="list" maxCount="" count=""><!--Optional;audio alarm configuration.-->
    <item>
        <switch type="boolean"><!--Required;enabled or not.--></switch>
        <audioType type="audioAlarmType">
            <!--Required;The ID of the alarm voice file,refer to the element
            <audioAlarmType> in the successful response of
            "GetAudioAlarmOutConfig". NVR is not supported.-->
        </audioType>
    </item>
</audio>
</item>
</boundary>
</temperatureAlarm>
</config>

```

Example of Successful Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
    <types>
        <ruleShapeType>
            <enum minCount="1" maxCount="1">point</enum>
            <enum minCount="2" maxCount="2">line</enum>
            <enum minCount="4" maxCount="6">area</enum>

```

```
</ruleShapeType>
```

```
<alarmRuleType>
```

```
  <enum>maxtemperabove</enum>
```

```
  <enum>maxtemperbelow</enum>
```

```
  <enum>mintemperabove</enum>
```

```
  <enum>mintemperbelow</enum>
```

```
  <enum>avgtemperabove</enum>
```

```
  <enum>avgtemperbelow</enum>
```

```
  <enum>difftemperabove</enum>
```

```
  <enum>difftemperbelow</enum>
```

```
</alarmRuleType>
```

```
<tempUnitType>
```

```
  <enum>centigrade</enum>
```

```
  <enum>fahrenheit</enum>
```

```
</tempUnitType>
```

```
<distanceUnitType>
```

```
  <enum>meter</enum>
```

```
  <enum>foot</enum>
```

```
</distanceUnitType>
```

```
</types>
```

```
<temperatureAlarm>
```

```
  <switch type="boolean">true</switch>
```

```
  <alarmHoldTime type="uint32" min="1" max="120" default="20">20</alarmHoldTime>
```

```
  <tempUnit type="tempUnitType">centigrade</tempUnit>
```

```
  <distanceUnit type="distanceUnitType">Meter</distanceUnit>
```

```
  <maxtemperen type="boolean">true</maxtemperen>
```

```
  <avgtemperen type="boolean">true</avgtemperen>
```

```
  <mintemperen type="boolean">true</mintemperen>
```

```
  <thermaldisplayen type="boolean">>false</thermaldisplayen>
```

```
  <opticaldisplayen type="boolean">>false</opticaldisplayen>
```

```
  <segcolorTemperatureParam>
```

```
    <switch type="boolean">>false</switch>
```

```
</segcolorTemperatureParam>
<dotTemperatureInfo>
  <switch type="boolean">>false</switch>
  <emissivity type="uint32" min="100" max="10000" default="9600">9600</emissivity>
  <distance type="uint32" min="0" max="1000000" default="50000">50000</distance>
  <reflectTemper type="int32" min="-300000" max="600000"
    default="250000">250000</reflectTemper>
</dotTemperatureInfo>
<boundary type="list" maxCount="10" count="1">
  <item>
    <ruleId type="uint32">0</ruleId>
    <switch type="boolean">>true</switch>
    <ruleName type="string">
      <![CDATA[test]]>
    </ruleName>
    <ruleType type="ruleShapeType">line</ruleType>
    <emissivity type="uint32" min="100" max="10000"
      default="9600">9600</emissivity>
    <distance type="uint32" min="0" max="1000000" default="50000">50000</distance>
    <reflectTemper type="int32" min="-300000" max="600000"
      default="250000">250000</reflectTemper>
    <alarmRule type="alarmRuleType">maxtemperabove</alarmRule>
    <alarmTemper type="int32" min="-200000" max="1500000"
      default="250000">1000000</alarmTemper>
    <stayAlarmDelay min="0" max="999" default="0">0</stayAlarmDelay>
    <pointGroup type="list" maxCount="2" count="2">
      <item>
        <X type="uint32">2239</X>
        <Y type="uint32">5768</Y>
      </item>
      <item>
        <X type="uint32">3239</X>
        <Y type="uint32">6768</Y>
      </item>
    </pointGroup>
  </item>
</boundary>
```

```

</pointGroup>
<ioOut type="list" maxCount="1" count="1">
  <item>
    <alarmId type="uint32">1</alarmId>
    <switch type="uint32">>false</switch>
  </item>
</ioOut>
<audio type="list" maxCount="1" count="1">
  <item>
    <switch type="boolean">>true</switch>
    <audioType type="audioAlarmType">10</audioType>
  </item>
</audio>
</item>
</boundary>
</temperatureAlarm>
</config>

```

[Tips]:

1. In order to measure accuracy, the values of elements<emissivity>,<distance>,<reflectTemper>,<alarmTmper> have been multiplied by 10000.

10.16.2 SetTemperatureAlarmConfig

SetTemperatureAlarmConfig	
Description	To set the IP media device's "temperatureAlarm" element.
Typical URL	POST http://<host>[:port]/SetTemperatureAlarmConfig[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<temperatureAlarm>
Successful Response	The standard successful result response that described in 1.3.5.

Applicable products

IPC,NVR

<temperatureAlarm> XML Block:

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

```
  <temperatureAlarm><!--Required; detection configuration.-->
    <switch><!--Required; boolean,detection is enabled or not.--></switch>
    <alarmHoldTime>
      <!--Required;uint32,The duration of the alarm after the event ends, Unit: seconds.-->
    </alarmHoldTime>
    <tempUnit><!--Required; tempUnitType,Unit of temperature.--></tempUnits>
    <distanceUnit><!--Required; distanceUnitType,Unit of Distance.--></distanceUnits>
    <maxtemperen>
      <!--Required; boolean,Display the highest temperature or not.-->
    </maxtemperen>
    <avgtemperen>
      <!--Required; boolean,Display average temperature or not.-->
    </avgtemperen>
    <mintemperen>
      <!--Required; boolean,Display the lowest temperature or not.-->
    </mintemperen>
    <thermaldisplayen>
      <!--Required; boolean,The thermal imaging channel displays temperature or not.-->
    </thermaldisplayen>
    <opticaldisplayen>
      <!--Required; boolean,The optical channel displays temperature or not.-->
    </opticaldisplayen>
    <segcolorTemperatureParam><!--Optional; Temperature bar configuration.-->
      <switch><!--Required; boolean,Display temperature bar or not.--></switch>
    </segcolorTemperatureParam>
    <dotTemperatureInfo><!--Optional; Temperature measurement of specified 'point'.-->
      <switch>
        <!--Required; boolean,Enable/Disable the temperature measurement function for the
        specified 'point'.-->
```

```
</switch>
<emissivity>
  <!--Required; uint32, emissivity.-->
</emissivity>
<distance>
  <!--Required; uint32, distance.-->
</distance>
<reflectTemper>
  <!--Required; int32, Reflection temperature.-->
</reflectTemper>
</dotTemperatureInfo>
<boundary><!--Optional; Regions and rules for temperature measurement.-->
  <item>
    <ruleId>
      <!--Required; uint32, Rule ID, starting from 0, less than the value of the property
      maxCount.-->
    </ruleId>
    <switch><!--Required; boolean, enable or not.--></switch>
    <ruleName>
      <!--Required; string, the name of rule.-->
    </ruleName>
    <ruleType><!--Required; ruleShapeType, the type of rule.--></ruleType>
    <emissivity>
      <!--Required; uint32, emissivity.-->
    </emissivity>
    <distance>
      <!--Required; uint32, distance.-->
    </distance>
    <reflectTemper>
      <!--Required; int32, Reflection temperature.-->
    </reflectTemper>
    <alarmRule type="alarmRuleType">
      <!--Required; alarmRuleType, Alarm rules.-->
```

```
</alarmRule>
<alarmTemper>
    <!--Required; int32, Temperature threshold for alarm.-->
</alarmTemper>
<stayAlarmDelay>
    <!--Optional; uint32, The duration of temperature measurement reaching the
    alarm condition. Unit: seconds.-->
</stayAlarmDelay>
<pointGroup><!--Required; Coordinates of the area requiring temperature
measurement.-->
    <item><!--Optional.-->
        <X><!--Required;uint32,X-coordinate--></X>
        <Y><!--Required;uint32,Y-coordinate--></Y>
    </item>
</pointGroup>
<ioOut><!--Optional; Linkage IO output after alarm triggering.-->
    <item>
        <alarmId><!--Required;uint32,alarm out ID.--></alarmId>
        <switch><!--Required;boolean,enabled or not.--></switch>
    </item>
</ioOut>
<audio><!--Optional;audio alarm configuration. NVR is not supported.-->
    <item>
        <switch><!--Required;enabled or not.--></switch>
        <audioType>
            <!--Optional;The ID of the alarm voice file,refer to the element
            <audioAlarmType> in the successful response of
            "GetAudioAlarmOutConfig".-->
        </audioType>
    </item>
</audio>
</item>
</boundary>
</temperatureAlarm>
```

```
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <temperatureAlarm>
    <switch>true</switch>
    <alarmHoldTime>20</alarmHoldTime>
    <tempUnit>centigrade</tempUnit>
    <distanceUnit>meter</distanceUnit>
    <maxtemperen>true</maxtemperen>
    <avgtemperen>true</avgtemperen>
    <mintemperen>true</mintemperen>
    <thermaldisplayen>false</thermaldisplayen>
    <opticaldisplayen>false</opticaldisplayen>
    <segcolorTemperatureParam>
      <switch>false</switch>
    </segcolorTemperatureParam>
    <dotTemperatureInfo>
      <switch>false</switch>
      <emissivity>9600</emissivity>
      <distance>50000</distance>
      <reflectTemper>250000</reflectTemper>
    </dotTemperatureInfo>
    <boundary>
      <item>
        <ruleId>0</ruleId>
        <switch>true</switch>
        <ruleName>
          <![CDATA[test1]]>
        </ruleName>
        <ruleType>line</ruleType>
        <emissivity>9600</emissivity>
```

```
<distance>50000</distance>
<reflectTemper>250000</reflectTemper>
<alarmRule>maxtemperabove</alarmRule>
<alarmTemper>1000000</alarmTemper>
<stayAlarmDelay>0</stayAlarmDelay>
<pointGroup>
  <item>
    <X>2239</X>
    <Y>5768</Y>
  </item>
  <item>
    <X>3239</X>
    <Y>6768</Y>
  </item>
</pointGroup>
<ioOut>
  <item>
    <alarmId>1</alarmId>
    <switch>>false</switch>
  </item>
</ioOut>
<audio>
  <item>
    <switch>>true</switch>
    <audioType>10</audioType>
  </item>
</audio>
</item>
</boundary>
</temperatureAlarm>
</config>
```

[Tips]:

1. In order to measure accuracy, the values of elements<emissivity>,<distance>,<reflectTemper>,<alarmTemper>need to be multiplied by 10000.

10.16.3 GetDotTemperature

GetDotTemperature	
Description	To get the temperature value data of the specified coordinate point.
Typical URL	POST or GET http://<host>[:port]/GetDotTemperature[/channelId]
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	None
Entity Data	<dotTemperature>
Successful Response	<config>
Applicable products	IPC,NVR
<p><dotTemperature> XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <dotTemperature><!--Required; Coordinates that require temperature measurement.--> <hotX><!--Required; uint32,X-Coordinates that require temperature measurement.--></hotX> <hotY><!--Required; uint32,Y-Coordinates that require temperature measurement.--></hotY> </dotTemperature> </config></pre> <p><config> of Successful Response XML Block:</p> <pre><config version="" xmlns="http://www.ipc.com/ver10"> <types> <tempUnitType><!--Required; Define the unit of temperature.--> <enum>centigrade</enum> <enum>fahrenheit</enum> </tempUnitType> </types> <switch type="boolean"><!--Required; Function enabled or not.--></switch></pre>	

```
<dotTemperature><!--Dependent; Valid when the value of <switch> is 'true'.-->
  <hotX type="uint32"><!--Required;X-Coordinates.--></hotX>
  <hotY type="uint32"><!--Required;,Y-Coordinates.--></hotY>
  <temperature type="int32">
    <!--Required; Measurement results, temperature values.-->
  </temperature>
  <tempUnit type="tempUnitType"><!--Required; unit of temperature value.--></tempUnits>
</dotTemperature>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <dotTemperature>
    <hotX>2049</hotX>
    <hotY>1764</hotY>
  </dotTemperature>
</config>
```

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <tempUnitType>
      <enum>centigrade</enum>
      <enum>Fahrenheit</enum>
    </tempUnitType>
  </types>
  <switch type="boolean">true</switch>
  <dotTemperature>
    <hotX type="uint32">2049</hotX>
    <hotY type="uint32">1764</hotY>
    <temperature type="int32">20210</temperature>
```

```
<tempUnit type="tempUnitType">fahrenheit</tempUnit>
</dotTemperature>
</config>
```

[Tips]:

1. In order to measure accuracy, the values of elements<temperature> has been multiplied by 10000.

11 Schedule commands

11.1 Schedule

11.1.1 GetScheduleConfig

GetScheduleConfig	
Description	To get the schedule of specified events or alarms with the action_name attached for specific channel.
Typical URL	POST or GET http://<host>[:port]/GetScheduleConfig[/channelId]</action_name>
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1.
Action name	The action names are defined as follows: alarmIn: schedule of alarmIn. In this scenario, the channelId is used as alarmIn ID motion: schedule of motion. record: schedule of record. snap: schedule of snap. tripwire: schedule of Tripwire Detection.

	<p>osc: schedule of Object Status Change.</p> <p>perimeter: schedule of Perimeter Environment Assurance.</p> <p>vfd: schedule of Video Face Detection.</p> <p>vehicle: schedule of Video vehilce Detection.</p> <p>aoientry: schedule of Aoi Entry Detection.</p> <p>aoileave: schedule of Aoi Leave Detection.</p> <p>passlinecount: schedule of Target Counting by Line Detection.</p> <p>traffic: schedule of Target Counting by Area Detection.</p> <p>heatMap: schedule of Heat Map Detection.</p> <p>vsd: schedule of Video Metadata Detection.</p> <p>lightAlarmOut: schedule of light Alarm Out.</p> <p>audioAlarmOut: schedule of audio Alarm Out.</p> <p>asd: schedule of Audio Abnormal Detection.</p> <p>pvd: schedule of Illegal Parking Detection.</p> <p>loitering: schedule of Loitering Detection.</p> <p>crowdGathering: schedule of Crowd Gathering Detection.</p> <p>disarming: schedule of disarming.</p> <p>temperatureAlarm: schedule of Thermal imaging temperature measurement.</p> <p>fire: schedule of fire.</p>
Entity Data	None
Successful Response	<schedule>
Applicable products	IPC

<schedule> XML Block:

```

<config version="" xmlns="http://www.ipc.com/ver10">
  <types>
    <weekDay><!--Required; Define the day type of the week.-->
      <enum>sunday</enum>
      <enum>monday</enum>
      <enum>tuesday</enum>
      <enum>wednesday</enum>
      <enum>thursday</enum>

```

```
<enum>friday</enum>
<enum>saturday</enum>
</weekDay>
</types>
<schedule><!--Required; Schedule infomation.-->
  <weekly type="list" maxCount="" count=""><!--Optional; weekly schedule infomation.-->
    <item>
      <startTime type="string">
        <!--Required; Schedule start time, Format HH:MM.-->
      </startTime>
      <endTime type="string">
        <!--Required; Schedule end time, Format HH:MM.-->
      </endTime>
      <day type="weekDay"><!--Required; Specify which day of the week.--></day>
    </item>
  </weekly>
  <yearly type="list" maxCount="" count=""><!--Optional; yearly schedule infomation.NVR is
  not supported.-->
    <item>
      <startTime type="string">
        <!--Required; Schedule start time, Format HH:MM.-->
      </startTime>
      <endTime type="string">
        <!--Required; Schedule end time, Format HH:MM.-->
      </endTime>
      <date type="string">
        <!--Required; Specify the scheduled date, Format MM-DD.-->
      </date>
    </item>
  </yearly>
</schedule>
</config>
```

Example of URL:

POST http://10.10.10.10:80/GetScheduleConfig/1/motion

Example of Successful Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <types>
    <weekDay>
      <enum>sunday</enum>
      <enum>monday</enum>
      <enum>tuesday</enum>
      <enum>wednesday</enum>
      <enum>thursday</enum>
      <enum>friday</enum>
      <enum>saturday</enum>
    </weekDay>
  </types>
  <schedule>
    <weekly type="list" maxCount="70" count="2">
      <item>
        <startTime type="string"><![CDATA[00:00]]></startTime>
        <endTime type="string"><![CDATA[23:59]]></endTime>
        <day type="weekDay">sunday</day>
      </item>
      <item>
        <startTime type="string"><![CDATA[00:00]]></startTime>
        <endTime type="string"><![CDATA[23:59]]></endTime>
        <day type="weekDay">monday</day>
      </item>
    </weekly>
    <yearly type="list" maxCount="31" count="1">
      <item>
```

```

        <startTime type="string"><![CDATA[00:00]]></startTime>
        <endTime type="string"><![CDATA[23:59]]></endTime>
        <date type="string"><![CDATA[05-12]]></date>
    </item>
</yearly>
</schedule>
</config>

```

[Tips]:

11.1.2 SetScheduleConfig

SetScheduleConfig	
Description	To set the schedule of specified events or alarms with the action_name attached for specific channel.
Typical URL	POST http://<host>[:port]/SetScheduleConfig[/channelId]</action_name>
Channel ID	Optional. If none channel ID included in the URL, the default channel ID is 1
Action name	The action name is the same as the one in "GetScheduleConfig".
Entity Data	<schedule>
Successful Response	The standard successful result response that described in 1.3.5.
Applicable products	IPC
<p><schedule> XML Block:</p> <pre> <config version="" xmlns="http://www.ipc.com/ver10"> <schedule><!--Required; Schedule infomation.--> <weekly><!--Optional; weekly schedule infomation.--> <item> <startTime><!--Required; Schedule start time, Format HH:MM.--></startTime> <endTime><!--Required; Schedule end time, Format HH:MM.--></endTime> <day><!--Required; Specify which day of the week.--></day> </pre>	

```
</item>
</weekly>
<yearly><!--Optional; yearly schedule information.NVR is not supported.-->
  <item>
    <startTime><!--Required; Schedule start time, Format HH:MM.--></startTime>
    <endTime><!--Required; Schedule end time, Format HH:MM.--></endTime>
    <date><!--Required; Specify the scheduled date, Format MM-DD.--></date>
  </item>
</yearly>
</schedule>
</config>
```

Example of Entity Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<config version="2.0.0" xmlns="http://www.ipc.com/ver10">
  <schedule>
    <weekly>
      <item>
        <startTime><![CDATA[00:00]]></startTime>
        <endTime><![CDATA[23:59]]></endTime>
        <day>sunday</day>
      </item>
      <item>
        <startTime><![CDATA[00:00]]></startTime>
        <endTime><![CDATA[23:59]]></endTime>
        <day>monday</day>
      </item>
    </weekly>
  </schedule>
</config>
```

[Tips]:

Annex A

A.1 Change Log

Date	Version	Note
2017-11-22	1.7	<ol style="list-style-type: none">1. add "2.1.6 GetDeviceDetail" section2. "5.3.1GetAlarmStatus" section, add status of smart alarm3. add "5.4 AlarmTrigger" section4. add "11 Smart commands" section5. add "12 Schedule commands" section

Date	Version	Note
2019-10-21	1.8	<ol style="list-style-type: none"> 1. Modify "2.1.6 GetDeviceDetail" add supportVfdMatch supportvehicle supportAoiEntry supportAoiLeave supportPassLineCount supportAudioAlarmOut supportWhiteLightAlarmOut 2. Modify "3.1.1 GetStreamCaps" encodeType add h264plus h265plus h264smart h265smart 3. Modify "3.3.1 GetAudioStreamConfig" add audioInSwitch audioInput audioOutput loudSpeaker 4. Modify "3.3.3 GetVideoStreamConfig" encodeType add h264plus h265plus h264smart h265smart 5. GetPtzConfig 6. SetPtzConfig 7. Modify "4.3.1 PtzGetPresets" presetInfo maxCount 255 -> 360 8. itemType maxLen 11 -> 10 9. Modify "5.4.1 GetAlarmTriggerConfig" Action name add vehicle aoientry aoileave passlinecount 10. Add "5.5 Sound-Light Alarm" section 11. Add "5.6 Alarm PIR" section 12. Modify "11.1 Face Detect & Face Comparison" section 13. Modify "11.5 Line Crossing" section 14. Modify "11.6 Intrusion" section 15. Add "11.9 License Plate Recognition" section 16. Add "11.10 Region Entrance" section 17. Add "11.11 Region Entrance" section 18. Add "11.12 Target Counting" section 19. Modify "12.1 GetScheduleConfig" Action name add vehicle aoientry aoileave passlinecount 20. Modify "12.3 SetScheduleConfigEx" scheduleObject add vehicle aoientry aoileave passlinecount.

Date	Version	Note
2020-05-06	1.9	<ol style="list-style-type: none"> 1. Add "11.13 Thermographic Temperature Measurement" 2. Add "11.14 Infrared temperature control" 3. Modify "2.1.6 GetDeviceDetail" add supportThermal 4. Modify "3.2.1 GetImageConfig" add node "backLightAdjust" 5. Modify "5.5.1 GetAudioStreamConfig" add enum "Abnormal temperature alarm" 6. Modify "11.1.8 SearchSnapFaceByKey" node matchInfo add "temperature" 7. Add "11.12.3 GetPassLineCountStatistics" 8. Modify "12.1.1 GetScheduleConfig" Action name add "thermal" 9. Modify "12.1.3 SetScheduleConfigEx" node scheduleObject add enum "thermal" 10. Modify "2.2.1 GetDateAndTime" add node "timeFormatMode" 11. Add "11.15 Heat Map" 12. Add "11.16 Region Statistics" 13. Add "8.2 Onvif User Management"

Date	Version	Note
2020-06-28	1.9	1. Add "2.3Upgrade"
2022-07-27	1.9	1. Add "9.1.2 channel_talk"
2022-08-27	1.9	1.Modify "4.4.2PtzGetCruise"section of Typical URL
2022-12-27	1.9	<ol style="list-style-type: none"> 1. Modify"2.1.6 GetDeviceDetail"add supportAsd,supportPvd,supportLoitering. 2. Modify"3.3.1 GetAudioStreamConfig" add enum "AAC",add "audioSampleRate","audioBitWidth","audioOutputswitch", "loudSpeakerswitch". 3. Modify "5.2.4 GetAlarmOutConfig" add "manualSwitch" 4. Add "5.2.6 AlarmOutputControl" 5. Modify "5.3.1 GetAlarmStatus" add "pvdAlarm","loiteringAlarm","asdAlarm" 6. Modify "5.4.1 GetAlarmTriggerConfig" add "asd" "pvd" "loitering" 7. Modify "5.5.1 GetAudioAlarmOutConfig" add "switch" "manualSwitch" 8. Modify "5.5.6 GetWhiteLightAlarmOutConfig" add "switch" "manualSwitch" 9. Add "10.18 Illegal Parking Detection" 10. add "10.19 Loitering Detection" 11. Modify "11.1.1 GetScheduleConfig" action_name add "whitelightAlarmOut","audioAlarmOut","asd","pvd","loitering" 12. Modify "11.1.3 SetScheduleConfigEx" add enum "whitelightAlarmOut","audioAlarmOut","asd","pvd","loitering"
2023-02-15	1.9	Optimize typesetting and correct some clerical errors.

Date	Version	Note
2024-05-23	1.9	<ol style="list-style-type: none"> 1. add "10.1.9 GetTargetFaceGroups ","10.1.10 AddTargetFaceGroup","10.1.11 DeleteTargetFaceGroup","10.1.12 EditTargetFaceGroup" 2. Supports NVR configuration and modifies the following interfaces: "10.1.3 AddTargetFace","10.1.3 DeleteTargetFace","10.1.3 EditTargetFace","10.1.3 GetTargetFace".
2024-10-31	2.0.0	<ol style="list-style-type: none"> 1. Add protocol field descriptions to all interfaces. 2. Add API interface: GetSupportedAPIs, GetAlarmInInfo, GetAlarmOutInfo, GetChannelInfo.AddPrivacyMaskConfig, ModifyPrivacyMaskConfig, DeletePrivayMaskConfig, Got oPrivacyMaskLocation. GetLightAlarmOutConfig, SetLightAlarmOutConfig. GetSmartCrowdGatheringConfig, SetSmartCrowdGatheringConfig. GetSmartFireConfig, SetSmartFireConfig. GetTemperatureAlarmConfig, SetTemperatureAlarmConfig, GetDotTemperature. 3. Removed API interface: GetSmartCddConfig, SetSmartCddConfig. GetSmartCpcConfig, SetSmartCpcConfig. GetSmartIpdConfig, SetSmartIpdConfig. Remove the interface related to "Thermographic Temperature Measurement". Remove the interface related to "Infrared Temperature Control".