IP Surveillance API
User Guide

Version 2.0

Revision 4

2013-12

HIKVISION

http://www.hikvision.com/

Revision History	Description	Reviser	Date
Version 2.0 Revision 1	Initial version	Hong Meng	2012-04
Version 2.0 Revision 2	merge qi's document	Hong Meng	2012-06
Version 2.0 Revision 3	add bond and holiday service	Minglei Yu	2012-10
Version 2.0 Revision 4	Combine front-end devices and back-end devices together	Minglei Yu Linming He Guangmu Ma	2013-12

http://www.hikvision.com/

Notices

The information in this documentation is subject to change without notice and does not represent any commitment on behalf of HIKVISION. HIKVISION disclaims any liability whatsoever for incorrect data that may appear in this documentation. The product(s) described in this documentation are furnished subject to a license and may only be used in accordance with the terms and conditions of such license.

This documentation is provided "as-is," without warranty of any kind. Please send any comments regarding the documentation to:
overseabusiness@hikvision.com

Find out more about HIKVISION at www.hikvision.com

Contents

1 Scope

This specification defines a HTTP-based application programming interface that enables physical security and video management systems to communicate with IP media devices in a particular way.

With regard to Media Streaming, please refer to “develop API of RTSP protocol”.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] RFC2616 Hypertext Transfer Protocol-HTTP/1.1
[2] W3C XML 1.0 specification
[3] W3C Character encodings
[4] RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax and Semantics
[5] RFC 2617 HTTP Authentication:Basic and Digest Access Authentication
[6] International Electrotechnical Commission “ISO/IEC standard on UPnP device architecture makes networking simple and easy”, 2008-12-09. Retrieved on 2009-05-07.
[7] International Organization for Standardization “ISO/IEC standard on UPnP device architecture makes networking simple and easy”, 2008-12-10. Retrieved on 2009-05-07.
[8] UPnP Forum “UPnP Specifications Named International Standard for Device Interoperability for IP-based Network Devices”, 2009-02-05. Retrieved on 2009-05-07.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Standard Resources: “index”, “indexr”, “description” and “capabilities” resources, that are contained in all Services and General Resources, and provide a special description for these resources.
Services: a set of resources consisting of relevant General Resources.
General Resources: physical resources that supported by the devices.
Node: Services and General Resources.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

FQDN	Fully Qualified Domain Name
REST	REpresentational State Transfer
IO	Input/Output
UPnP	Universal Plug and Play

4 Architecture and Transmission Mechanism

The IP Media Device Management Protocol is based on REST architecture. The management and control interfaces defined in this specification are treated as resources utilizing the REpresentational State Transfer (REST) architecture. This architecture facilitates users by grouping related resources within hierarchical namespaces, and is more flexible for service discovery and future expansion.

REST architecture consists of clients and servers, among which clients initiate request to servers, while servers handle requests and response accordingly. Requests and responses are established via the transmission of “representations” of “resources”. REST architecture need to be based on an Application Layer protocol which provides various of standard communication formats for applications based on the transfer of meaningful representational state. HTTP[1] has a very rich vocabulary in terms of verbs(or “methods”),

URIs, request and response headers, Internet media types, HTTP request and response codes etc. In addition, HTTP also has some features particularly suitable for REST architecture. So HTTP is used as external Application Layer protocol in this specification. In the architecture, clients are physical security and video management systems; servers are IP media devices.

This specification also contains full XML schema for the introduced resources.

4.1 REST and HTTP Methods

The following table shows how HTTP verbs are typically used to implement a web service based on REST architecture.

Table 1

Resource	GET	PUT	POST	DELETE
Collection URI, such as http://webServer/resources	**List** the members of collection, complete with their member URIs for further navigation.	Meaning defined as "**replace** the entire collection with another collection".	**Create** a new entry in the collection where the ID is assigned automatically by the collection. The ID created is usually included as part of the data returned by this operation.	Meaning defined as "**delete** the entire collection".
Member URI, such as http://webServer/resources/7416	**Retrieve** a representation of the addressed member of the collection expressed in an appropriate MIME type.	**Update** the addressed member of the collection or **create** it with the specified ID.	Treat the addressed member as a collection in its own right and **create** a new subordinate of it.	**Delete** the addressed member of the collection.

4.2 XML

A device must support the syntax defined by W3C XML 1.0 specification [2] and UTF-8 character set [3]. All XML files must adopt UTF-8 encoding according to RFC3629. Additionally, all resources share a common XML schema as defined in Annex.

Any resources can specify separate input and output XML Documents. If a specific data

structure is defined inside these documents, then they must be specified as XML Schema Documents (xsd) in Annex.

Lists contained in XML blocks will be represented in the format of <ISAPIList>, and each <ISAPIList> tag may contain one or more nodes.

4.3 Resources overview

Three kinds of resources are defined in this specification. They are “Standard Resources”, “Services” and “General Resources”. Related General Resources are grouped by Services. Services and General Resources contain Standard Resources. Figure 1 shows their relationship.

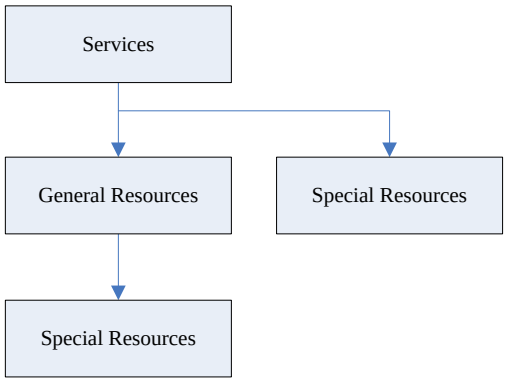


Figure 1

The “index”, “indexr”, “description” and “capabilities” are defined as Standard Resources in this specification. Both “index” and “description” will be mandatorily included by each node, and both “indexr” and “capabilities” will be optionally included by each node. For more detailed description see Section 6.

Services defined in this specification are divided into different services categories. Each category has its own name spaces (see Section 4.6 for the name space definitions). The following services are defined:

Table 2

Services	Description	Reference
System	Configure and operate the general system functions.	8.1
Network	Configure network interfaces.	8.2
IO	Configure the Input/Output (IO).	8.3
Video	Handle video-related configuration.	8.4
Audio	Configure the Audio.	8.5
Two way audio	Control two ways audio.	8.6
Serial	Configure and control the Serial ports.	8.7
Security	Provide Security functions.	8.8

Services	Description	Reference
Streaming	Configure and control the streaming media content.	8.9
Motion Detection	Configure and control the motion detection of the device	8.10
Event	Provide event notification functions.	8.11
PTZ	Control the device pan tilt and zoom.	8.12

4.4 Protocol URL

The URL scheme is used to locate device resources via a specific protocol in the network. This section defines the syntax and semantics for http(s) URLs.

```
<protocol>://<host>[:port][abs_path [?query]]
```

protocol: URL scheme for the particular request. The http and https protocols are allowed in this specification.

host: The host field refer to the hostname, IP address, or the FQDN 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. For HTTPS, the default port 443.

abs_path: The Request-URI [1] for the resources is abs_path [4]. The abs_path in this specification is most often of the form "[/Services][/General Resources][/Standard Resources]", which is suitable for resources to update or restore device configurations. "*ID*" which appears in the abs_path identifies one resource of a list resource in this specification.

query: The query field is a string of information to be interpreted by the resource. It can include some resource-related parameters. It must be listed in name-value pair syntax (p1=v1&p2=v2&…&pn=vn). Each resource can define a set of parameters. Defining input data which is specific to the resource will be prior than query usage.

4.5 Messages

HTTP messages are used for communication between physical security and video management systems and IP media devices in this specification. In order to configure and control the device, some provisions are specified for these HTTP message.

4.5.1Connection Header Field

Devices that implement HTTP/1.1 should support persistent connections in order to meet video management systems or client applications' requirements that issue multiple HTTP(s) transactions. 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, while also adopt the "Connection: close" HTTP header field for the last transaction of the persistent connection. This process will assume that the application can identify the last request in a sequence of multiple requests.

4.5.2Authorization and WWW-Authenticate Header Fields

When a video management system or client application sends any request to the device, it must be authenticated by means of Basic Access [5] according to RFC 2617, and thus all the devices are required to support Basic Access. Authorization header field is sent along with each request, and if a user is authenticated, the request will follow the normal execution flow. If client HTTP request is with no authentication credentials, unauthorized HTTP response (401) will be returned with WWW-Authenticate header field.

4.5.3Entity Body

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'", "application/octet-stream", etc.

For configuration information, the Content-Type is usually "application/xml; charset='UTF-8'". For example,

HTTP Request Message:

```
GET /ISAPI/System/status HTTP/1.1
…
```

HTTP Response Message:

```
HTTP/1.1 200 OK
…

Content-Type: application/xml; charset="UTF-8"
…
```

```
<?xml version=”1.0” encoding=”UTF-8”?>
<DeviceStatus version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
…
</DeviceStatus>
```

For data (i.e. firmware, configuration file, etc.), the Content-Type may be “application/octet-stream”. For example,

HTTP Request Message:

```
PUT /ISAPI/System/configurationData HTTP/1.1
…
Content-Type: application/octet-stream
…
[proprietary configuration file data content ]
```

HTTP Response Message:

```
HTTP/1.1 200 OK
…

Content-Type: application/xml; charset=”UTF-8”
…
<?xml version=”1.0” encoding=”UTF-8”?>
<ResponseStatus version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
…
</ResponseStatus>
```

4.5.4Operations

Different resources will specify different operation.

- The “set device configuration” resources use PUT operation. If there is an XML block parameter for the request, the inbound XML format is defined according to a resource-special XML schema. Request status will be returned by the XML response information of the device, and can be used for indicating the PUT operation status. The responded XML format is defined by “XML Response Schema” (please refer to section 4.5.5 for detail description). After the device configuration is updated successfully, it will return an XML response with status code “OK”; while another status code will be used for indicating unsuccessful operations. In either case, the device only responses after it is ready to continue normal operation, i.e. accepting streaming request, receiving configuration commands, etc.
- The “get device configuration” resources use GET operation. After a successful GET operation, the result will be returned in XML format according to the

resource description. For an unsuccessful request (i.e. users is not authenticated), the result will be returned in XML format according to “XML Response Schema”.

- Resources to create device configurations information will use the POST operation. If there is an XML block parameter for the request, the inbound XML format is defined according to a resource-special XML schema. The request status will be indicated by the XML response information returned from the device, and can be used to indicate the status of the POST operation. This XML format is defined according to “XML Response Schema” (see section 4.5.5 for details). After successfully creating the data, the device returns an XML response with status code “OK”. A separate status code is used for unsuccessful operations.
- Resources to delete device configurations information will use the Delete operation. If successful, the result will be returned an XML response with status code “OK”. A separate status code is used for unsuccessful operations. This XML format is defined according to “XML Response Schema” (see section 4.5.5 for details).
- Data uploading resources (i.e. firmware upgrade, import configuration, etc.) will use PUT operation. The content of the data will be stored in the body of the HTTP request. If successful, the result will be returned an XML response with status code “OK”. A separate status code is used for unsuccessful operations. This XML format is defined according to “XML Response Schema” (see section 4.5.5 for details).
- Data receiving resources (i.e. export configuration file) use GET operation. If successful, the result will be returned the data according to the resource description. An XML block is used for unsuccessful operations. This XML format is defined according to “XML Response Schema” (see section 4.5.5 for details).
- For Standard Resources, GET operation will be used. For more detailed description see Section 6.

If there is an XML block for the HTTP request or response, the Content-Type and Content-Length will be set in the headers of the HTTP message.

4.5.5Error Handling

As with any other protocol, errors may occur during communications, protocol or message processing, and the specification classifies error handling into categories below:

- Protocol Errors, which are result of an incorrectly formed protocol message. Protocol Errors may contain header value or be received in an not expected or experience a socket timeout. To indicate and interpret protocol error, HTTP protocol has defined a set of standard status codes [e.g., 1xx, 2xx, 3xx, 4xx, 5xx]. According to this specification, the IP devices will use appropriate HTTP protocol defined status codes for error reporting and when received handle accordingly.

- Application Errors, which are generated as a result of REST operations errors. All such application errors must be reported and handled through HTTP messages. The following table indicates the mapping relationship between HTTP status codes and REST operations, and also the information contained in response header and bodies.

Table 3

HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
200	"OK"-The request has succeeded. Header Notes: None Body Notes: The requested resource will be returned in the body.	√	√		√
201	"Created"- The request has created a new resource. Header Notes: The Location header contains the URI of the newly created resource. Body Notes: The response returns an entity describing the newly created resource.		√	√	
204	"No Content" – The request succeeded, but there is no data to return. Header Notes: None Body Notes: No body is allowed.		√		√
301	"Moved Permanently" – The requested resource has moved permanently. Header Notes: The Location Header contains the URI of the new location. Body Notes: The body may contain the new resource location.	√			
302	"Found" – The requested resource should be accessed through this location, but the resource actually lives at another location. This is typically used to set up an alias.	√			

HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
	Header Notes: The Location header contains the URI of the resource. Body Notes: The body may contain the new resource location.				
400	"Bad Request" – The request was badly formed. This is commonly used for creating or updating a resource, but the data was incomplete or incorrect. Header Notes: The Reason-Phrase sent with the HTTP status header may contain information on the error. Body Notes: The response may contain more information of the underlying error that occurred in addition to the Reason-Phrase.		√	√	
401	"Unauthorized" – The request requires user authentication to access this resource. If the request contains invalid authentication data, this code is sent. Header Notes: At least one authentication mechanism must be specified in the WWW-Authenticate header. The Reason-Phrase sent with the HTTP status header may contain information on the error. Body Notes: The response may contain more information of the underlying error that occurred in addition to the Reason-Phrase.	√	√	√	√
403	"Forbidden" – The request is not allowed because the server is refusing to fill the request. A	√	√	√	√

HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
	common reason for this is that the device does not support the requested functionality. Header Notes: The Reason-Phrase sent with the HTTP status header may contain information on the error. Body Notes: The response may contain more information of the underlying error that occurred in addition to the Reason-Phrase.				
404	"Not Found" – The requested resource does not exist. Header Notes: None Body Notes: None	√	√	√	√
405	"Method Not Allowed" – The request used an HTTP method that is not supported for the resource because the specification does not allow this method. If the device does support the functionality but it is a valid operation (that has been defined in this specification), then 403 is returned. Header Notes: The Allow header lists the supported HTTP methods for this resource. Body Notes: None	√	√	√	√
500	"Internal Server Error" - An internal server error has occurred. Header Notes: None Body Notes: None	√	√	√	√
503	"Service Unavailable" – The HTTP Server is up, but the REST service is not available. Typically this is caused by too many client requests. Header Notes: The Retry-After	√	√	√	√

HTTP Status Codes	REST Meaning	GET	PUT	POST	DELETE
	header suggests to the client when to try resubmitting the request. Body Notes: None				

Responses to many resources calls contain data in XML format. XML Response Schema is defined in Annex. XML Response Schema consists of the following sections:

- requestURI - the URI of the corresponding HTTP request message
- statusCode - indicating the status of the REST operations.

Table 4

statusCode	Description
1	"OK" - indicate a successful operation is done (remark: if the request contains some parameters that are not supported, the device will ignore those parameters and return OK as statusCode)
2	"Device Busy" - for a command which cannot be processed at that time (i.e. if the device receives a reboot command during upgrading process)
3	"Device Error" - if the device can not perform the request for a hardware error. An error message in statusString format to indicate operation failure
4	"Invalid Operation" - either if the operation is not supported by the device, or if the user has not passed the authentication, or if the user does not have enough privilege for this operation
5	"Invalid XML Format" - if the XML format is not recognized by the system. There will be statusString returned to represent different errors
6	"Invalid XML Content" - an incomplete message or a message containing an out-of-range parameter. Relative statusString will be return.
7	"Reboot Required" - If a reboot is required before the operation taking effect

- statusString – error type for the not completed operation.
- id – Return the ID created by the device in POST operation
- subStatusCode – detail string indicating the reason the command was not completed.Table 5 contains general subStatusCode,In addition,Each resource may have some special subStatusCode，Each subStatusCode reference resource description.

Table 5

statusCode	subStatusCode	Description
1	ok	indicate a successful operation is done
2	noMemory	Device doesn't have enough memory
	serviceUnavailable	service unavailable

statusCode	subStatusCode	Description
	upgrading	upgrading
	deviceBusy	Device busy or no response
3	deviceError	Device hardware error
	badFlash	Operate flash error
4	notSupport	The device does't support this resource
	lowPrivilege	Not have enough privilege for this operation
	badAuthorization	The user has not passed the authentication
	methodNotAllowed	http method is not allowed
	notSetHdiskRedund	can't set redundancy attribute for hdd disk(system exists more than one non-operate hdd disk,and the attribution of a hdd disk is WR)
	invalidOperation	Invalid operation
5	badXmlFormat	Wrong XML format
6	badParameters	Parameters error
	badHostAddress	Wrong Host Address
	badXmlContent	Wrong XMLcontent
	badIPv4Address	Wrong IPv4 address
	badIPv6Address	Wrong IPv6 address
	conflictIPv4Address	IPV4 address conflict
	conflictIPv6Address	IPV6 address conflict
	badDomainName	Wrong Domain
	connectSreverFail	Failed to connect with Server
	conflictDomainName	Domain conflict
	badPort	Port conflict
	portError	Port error
	importErrorData	Failed to import data
	badNetMask	Wrong subnet mask
	badVersion	Version mismatching
	badDevType	Device type mismatching
	badLanguage	Language mismatching
7	rebootRequired	A reboot is required before the operation taking effect

4.6 Namespaces

The namespace xmlns="http://www.isapi.org/ver20/XMLSchema" is used in this specification.

The following namespaces are referenced by this specification:

- xmlns:xs="http://www.w3.org/2001/XMLSchema"

- xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
- xmlns:xlink=”http://www.w3.org/1999/xlink”

4.7 Security

User-based access control is adopted in this specification. Security policy configuration in this specification based on three different user levels.

- Administrator – the privilege can access all supported resources on IP device.
- Operator – the privilege can access some general-level and higher-level resources. See the Resource Description of each resource for details.
- Viewer – the privilege can only access some general-level resources. See the Resource Description of each resource for details.

In order to access all supported resources, one account with Administrator privilege level must be active at all times. A default user account “admin” is provided by all IP devices. It has an Administrator user level, and must not be deleted. Its default password is “12345”.

5 Device discovery

The IP devices support Universal Plug and Play (UPnP) technology to discovery/locate themselves. A UPnP compatible device will automatically announce its network address, supported devices and services types when connected to a network, and 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].

The foundation for UPnP networking is IP addressing. When a device is connected to the network for the first time, its Dynamic Host Configuration Protocol (DHCP) client will search for a DHCP server. If the device successfully get its domain name via DNS server or DNS forwarding, then it should use this domain name for the following network operations; if the network is unmanaged and no DHCP server is found, the device must assign an address for itself, which is known as “AutoIP” of the UPnP Device Architecture [9][10], and use this IP address for the following network operations.

Once given an IP address, the Discovery process will be executed in UPnP networking. The UPnP discovery protocol is also knows as Simple Service Discovery Protocol (SSDP). When a device is added to the network, SSDP allow that device to announce its services to the control points on the network. Similarly, when a control point is added to the network, SSDP allows that control point to search for relative devices on the network. During the above searching or announcing process, a a discovery message which contains essential device specifics or one of its services will be transfered, for example, device type, identifier, and a pointer to more detailed information.

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. An HTTP GET request for mandatory index Standard Resource will return a list of the resources supported by the device.
Remark: the index resource will only return the first level resources of a node, while the indexr Standard Resource will return a complete folder list in tree structure with the current resource as root folder.

6 Resource Description

6.1 Resource Description Outline

Each resource in this specification is defined using the following format.

Resource_URI	***Type Version***
Operation_Name	
Description	*Description of the operation.*
Query	*Indicates the name/value pairs (p1, p2, p3,…,pn) for the resource.*
Inbound Data	*Indicates inbound data for the resources.*
Success Return	*the Type (if present) and the name of XML Data Block*
Error Status Code	*Special fault code，optional*
Notes: d*escribes any special processing rules for the resource.*	

Type refers to “Standard Resource”, “Service” and “General Resource”.
Version is used to determine the version of the protocol. The version number shall be set to “1.0” in this specification.
Operation_Name refers to “GET”, “PUT”, “POST” and “DELETE”.
Inbound Data includes three types as follows:

- NONE –no input data
- DataBlock – the name of an XML Data Block. Datablocks used here must be defined according to the specification.

- Mime type – mime type for the input data in the HTTP payload. Remark: “application/ xml” is not a valid mime type.

If a device does not support particular XML tags or blocks, then it may not be supported by the resource operations.

Generally, if a field is not provided in the inbound XML, then its current values shall not be modified in the device’s repository.

If a required field did not exist in the device’s repository, then it must be provided in the applicable resource operations.

Success Return and Error Return detailed description see Section 4.5.5.

6.2 Built-in Types

Table 6

Type	Description
BaudRate	A positive numerical value indicating the data transmission rate in symbols per second. Value is >=0. Example: 9600
Color	RGB triplet in hexadecimal format (3 bytes) without the preceding “0x”. Example: “FF00FF”
Coordinate	A positive numerical value in pixels. A coordinate pair of 0,0 (x,y) indicates the bottom-left corner of the video image. Value is >=0. Maximum value is dependent on video resolution.
FPS	Frame rate multiplied by 100. Example: 2500 [PAL]
IPv4 Address	Notation is ISAPI.ISAPI.ISAPI.ISAPI Example: 3.137.217.220
MAC	MAC Address Notation is aa:bb:cc:dd:ee:ff with 6 hex bytes.

6.3 Annotation

The XML Data Blocks described in this document contains annotations for the field’s properties. Please refer to the XML schema definitions for detail description.

The following annotation content is inserted into the comments to describe the data carried in the field:

Table 6

Annotation	Description

req	Required field.
Opt	Optional field. For data uploaded to the device, if the field is present but the device does not support it, it should be ignored.
Dep	This field is required depending on the value of another field.
Ro	Read-only. For XML data that is both read and written to the device, this field is only present in XML returned from the device. If this field is present in XML uploaded to the device, it should be ignored.
Wo	Write-only. This field is only present in XML that can be uploaded to the device. This field should never be present in data returned from the device. [This is used for uploading passwords].
Xs:<type>	A type defined in XML Schema Part 2: Datatypes Second Edition, see http://www.w3.org/TR/xmlschema-2

Remark: optional XML structures may contain required fields for the operation, which mean that even if the entire XML block is optional, some of its contained fields may still be necessary if required.

7 Standard Resources

This section describes the standard resources.
Standard Resources do not contain themselves.
The requestURIs "/index", "/description" are required.

7.1 index

index	**Standard Resource v2.0**
GET	
Description	Enumerate child resources of a resource.
Query	None
Inbound Data	None
Success Return	<**ResourceList**>
Notes: Returns a non-recursive resource listing of all child resources.	

7.2 indexr

indexr	**Standard Resource v2.0**
GET	
Description	Enumerate child resources of a resource.
Query	None

Inbound Data	None
Success Return	<**ResourceList>**
Notes: Returns a recursive resource listing of all child resources.	

7.3 description

description	**Standard Resource v2.0**
GET	
Description	Describe the corresponding resource
Query	None
Inbound Data	None
Success Return	**<ResourceDescription>**
Notes: <version> set the version of resource. In this specification, its value is “2.0”.	

A version attribute is included in the description. This means resources with different versions may exist within the same Services. In that case, the version of Services is the version of the contained resource with the lowest version, and all resources in the Services container must be backward compatible. If any resource of a Service container can not maintain backward compatibility with previous versions, a new Services version should be introduced.

7.4 capabilities

capabilities	**Standard Resource v2.0**
GET	
Description	Describe the capabilities of the corresponding resource
Query	None
Inbound Data	None
Success Return	Resource-specified
Notes:	

For the General Resource, which inbound data is specified as an XML payload, the Standard Resource (capabilities) is provided for video management systems or client applications to query an IP device and understand what XML tags are supported.

“Capabilities” is essentially an XML instance of the corresponding General Resource XML Data Block. “Capabilities” must contain the acceptable values for each attribute.

While XML Schema Document are also required of any XML data defined by this specification and xsd documents are capable of defining the acceptable range of values

for any attribute, using a global xsd to define capacities would imply that all devices support the same options for any parameter. By allowing devices to respond to the capabilities request, each device can support different values for any attribute, within the constraints of the schema.

Table 7

Capabilities Attribute	Description	Syntax	Applicable XML Data Types
min	The minimum character length for a string, or the minimum numerical value of a number	Examples: min="0" min="19" min="-74"(numerical only) min="1.6"	All except fixed data types[1)]
max	The maximum character length for a string, or the maximum numerical value of a number	Examples: max="4" max="37" max="8192" max="14.61"	All except fixed data types[1)]
range	Indicates the possible range of numerical values within the "min" and "max" attributes of an element. This attribute should only be used if the possible value for an XML element does not include the entire numerical range between "min" and "max" attributes	Ranges are listed in numerical order separated by a "," character. A range has the form "x~y" where x is the range floor and y is the range ceiling. Single numbers may also be used. Example: if an XML element supports values 0, 456, 1674 to 2009 and 2012, the syntax would be: range="0, 456, 1674~2009, 2012"	All numerical data types
opt	All except fixed data types	If all options are supported, the syntax is "all". Otherwise, supported options are listed separated by a ", " character. Examples:	All except fixed data types

Capabilities Attribute	Description	Syntax	Applicable XML Data Types
		opt=”all” opt=”1, 4, 6, 7”	
def	Indicates the default value of the XML element. If the element has not default value, this attribute should not be used	Examples: def=”7416” def=”ace”	All data types
reqReboot	Indicates if configuration of this XML element requires a device reboot before taking effect. If an element does not require a boot, this attribute should not be used	reqReboot=”true”	All data types
dynamic	Indicates if an XML element has dynamic capabilities dependent on other XML configuration. For example, if an element’s data range changes based on another element’s configured value, this attribute must be used. In this case, the element’s capability attributes must always reflect the current device configuration	dynamic=”true”	All data types
Size	Indicates the maximum number of entries in an XML List. This attribute is only applicable to XML list elements. This attribute should not be used for any other type of element	Example: If a device supports 16 users the example would be <UserList size=”16”> <User> … </UserList>	Only supported for list elements

1) Fixed, pre-defined data types do not need certain capability attributes because their formats/data ranges are already defined.

8 Services and General Resources

8.1 /ISAPI/System

/ISAPI/System	Service	v2.0
Notes:		

8.1.1/ISAPI/System/capabilities

/ISAPI/System/capabilities	General Resource	v2.0
GET		
Description	It is used to get device capability.	
Query	None	
Inbound Data	None	
Success Return	**<DeviceCap>**	
Notes: Some capabilities that could not be described by statand capability resource will be listed here. <isSupportDst>: Is this device support daylight saving time.		

DeviceCap XML Block

```
<DeviceCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SysCap> <!-- opt -->
    <isSupportDst> <!-- opt, xs:boolean --> </isSupportDst>
   <NetworkCap/> <!-- opt -->
   <IOCap/> <!-- opt -->
   <SerialCap/> <!-- opt -->
   <VideoCap/> <!-- opt -->
   <AudioCap/> <!-- opt -->
  </SysCap>
  < voicetalkNums > <!-- opt, xs:integer --> </ voicetalkNums >
  <isSupportSnapshot> <!-- opt, xs:boolean --> </isSupportSnapshot>
  <SecurityCap/> <!-- opt -->
  <EventCap/> <!-- opt -->
  <ImageCap/> <!-- opt -->
  <RacmCap/> <!-- opt -->
```

```
  <SmartCap/> <!—- opt -->
  <WLAlarmCap/> <!—- opt-->
</DeviceCap>
```

8.1.2/ISAPI/System/reboot

/ISAPI/System/reboot			General Resource v2.0
PUT			
Description	Reboot the device.		
Query	None		
Inbound Data	None		
Success Return	**<ResponseStatus>**		
Error Status Code	**statusCode**	**subStausCode**	**description**
	2	**upgrading**	**Device is upgrading**
Notes: <ResponseStatus> is returned before the device proceeds to reboot.			

8.1.3/ISAPI/System/updateFirmware

/ISAPI/System/updateFirmware			General Resource v2.0
PUT			
Description	Updatethe firmware of the device.		
Query	None		
Inbound Data	**Opaque Data**		
Success Return	**<ResponseStatus>**		
Error Status Code	**statusCode**	**subStatusCode**	**description**
	2	**upgrading**	**device upgrading**
	3	**badFlash**	**Flash error**
	6	**badLanguage**	**Language mismatch**
Notes: After successful completion of this API, the <ResponseStatus> XML data is returned, and the device proceeds to reboot.			

8.1.4/ISAPI/System/configurationData

/ISAPI/System/configurationData			General Resource v2.0
GET			
Description	Get device's configuration data.		
Query	None		
Inbound Data	None		
Success Return	**Opaque Data**		
PUT			
Description	Update device's configuration data.		
Query	None		
Inbound Data	**Opaque Data**		
Success Return	**<ResponseStatus>**		
Error Status Code	**statusCode**	**subStatusCode**	**description**
	2	**upgrading**	**Device upgrading**
	3	**badFlash**	**Flash error**
	6	**badVersion**	**Version mismatch**
	6	**badDevType**	**Device type mismatch**
	6	**badLanguage**	**Language mismatch**
Notes: Configuration file is device-dependant – it may be binary or any other format. May reboot device after configuration file is applied.			

8.1.5/ISAPI/System/factoryReset

/ISAPI/System/factoryReset	General Resource v2.0
PUT	
Description	It is used to reset the configuration for the device to the factory default.
Query	mode
Inbound Data	None
Success Return	**<ResponseStatus>**
Notes: Two factory reset modes are supported: "full" resets all device parameters and settings to their factory values. "basic" resets all device parameters and settings except the values in Network Service.	

The default mode is “full”. The device may be rebooted after it is reset.

8.1.6/ISAPI/System/deviceInfo

/ISAPI/System/deviceInfo	General Resource v2.0
GET	
Description	It is used to get device information.
Query	None
Inbound Data	None
Success Return	**<DeviceInfo>**
PUT	
Description	It is used to update device information.
Query	None
Inbound Data	**<DeviceInfo>**
Success Return	**<ResponseStatus>**

Notes:

Some fields are read-only and may not be set. If these fields are present in the inbound XML block, they are ignored.
For the <DeviceInfo> uploaded to the device during a PUT operation, all fields are considered optional and any fields that are not present in the inbound XML are not changed on the device. This allows setting of the fields individually without having to load the entire XML block to the device.
<deviceDescription> is a description of the device as defined in RFC1213.
For IPC the <deviceDescription> value is IPCamera;
For IP speed Dome the <deviceDescription> value is IPDome;
For DVR or DVS the <deviceDescription> value is DVRDVS;
<deviceLocation> is the location of the device as defined in RFC1213
<systemContact> is the contact information for the device as defined in RFC1213.

DeviceInfo XML Block

```
<DeviceInfo version="2.0" xmlns=“http://www.isapi.org/ver20/XMLSchema”>
   <deviceName>        <!-- req, xs:string -->          </deviceName>
   <deviceID>          <!-- ro, req, xs:string, uuid-->          </deviceID>
   <deviceDescription>   <!—opt, xs:string-->   </deviceDescription>
   <deviceLocation>       <!—opt, xs:string -->       </deviceLocation>
   <systemContact>       <!-- opt, req, xs:string -->         </systemContact>
   <model>               <!-- ro, req, xs:string -->       </model>
   <serialNumber>        <!-- ro, req, xs:string -->       </serialNumber>
```

```
<macAddress>          <!-- ro, req, xs:string;     -->   </macAddress>
<firmwareVersion>      <!-- ro, req, xs:string -->      </firmwareVersion>
<firmwareReleasedDate>   <!-- ro, opt, xs:string -->      </firmwareReleasedDate>
<bootVersion>          <!-- ro, opt, xs:string -->      </bootVersion>
<bootReleasedDate>      <!-- ro, opt, xs:string -->      </bootReleasedDate>
<hardwareVersion>      <!-- ro, opt, xs:string -->      </hardwareVersion>
<encoderVersion> <!-- ro, opt, xs:string> </encoderVersion>
<encoderReleasedDate> <!-- ro, opt, xs:stirng --> </encoderReleasedDate>
<decoderVersion> <!-- ro, opt, xs:string> </decoderVersion>
<decoderReleasedDate> <!-- ro, opt, xs:stirng --> </decoderReleasedDate>
<deviceType>
   <!--ro, req, xs:string; “IPCamera, IPDome, DVR, HybirdNVR, NVR, DVS, IPZoom”-->
<deviceType>
<telecontrolID> <!-- opt, xs:integer; “1-255”> <telecontrolID>
<supportBeep/> <!--opt, xs:boolean --> </supportBeep>
</DeviceInfo>
```

8.1.7/ISAPI/System/status

/ISAPI/System/status	General Resource v2.0
GET	
Description	It is used to get the status information of the device.
Query	None
Inbound Data	None
Success Return	**DeviceStatus**
Notes:	

DeviceStatus XML Block

```
<DeviceStatus version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
<currentDeviceTime>     <!-- opt, xs:datetime -->      </currentDeviceTime>
<deviceUpTime>     <!-- opt, xs:integer, seconds --> </deviceUpTime>
<TemperatureList>
   <!-- opt -->
   <Temperature>
      <tempSensorDescription>     <!-- req, xs:string -->   </tempSensorDescription>
      <temperature>  <!-- req, xs:float -->     </temperature>
   </Temperature>
</TemperatureList>
```

```
<FanList>
  <!-- opt -->
  <Fan>
    <fanDescription>  <!-- req, xs:string -->  </fanDescription>
    <speed>  <!-- req, xs:integer --> </speed>
  </Fan>
</FanList>
<PressureList>
  <!-- opt -->
  <Pressure>
    <pressureSensorDescription>  <!-- req, xs:string
--></pressureSensorDescription>
    <pressure>  <!-- req, xs:integer -->  </pressure>
  </Pressure>
</PressureList>
<TamperList>
  <!-- opt -->
  <Tamper>
    <tamperSensorDescription>  <!-- req, xs:string -->  </tamperSensorDescription>
    <tamper>  <!-- req, xs:boolean -->  </tamper>
  </Tamper>
</TamperList>
<CPUList>
  <!-- opt -->
  <CPU>
    <cpuDescription>  <!-- req, xs:string -->  </cpuDescription>
    <cpuUtilization>  <!-- req, xs:integer, percentage 0..100 --> </cpuUtilization>
  </CPU>
</CPUList>
<MemoryList>
  <!-- opt -->
  <Memory>
    <memoryDescription>  <!-- req, xs:string -->  </memoryDescription>
    <memoryUsage>  <!-- req, xs:float, in MB --> </memoryUsage>
    <memoryAvailable><!-- req, xs:float, in MB--> </memoryAvailable>
  </Memory>
</MemoryList>
<openFileHandles> <!-- opt, xs:integer -->  </openFileHandles>
```

```
</DeviceStatus>
```

8.1.8/ISAPI/System/time

/ISAPI/System/time	General Resource v2.0
GET	
Description	Get the device time information.
Query	None
Inbound Data	None
Success Return	**Time**
PUT	
Description	Udpate the device time information.
Query	None
Inbound Data	**Time**
Success Return	**ResponseStatus**
Notes: If <timeMode> is present and set to “local”, the <localTime> and <timeZone> fields are required. The <localTime> block sets the device time. If <timeMode> is present and set to “NTP”, only the <timeZone> field is required. The device time is set by synchronizing with NTP.	

Time XML Block

```
<Time version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
   <timeMode>     <!-- req, xs:string, “NTP, manual” -->       </timeMode>
   <localTime>      <!-- req, xs:datetime -->              </localTime>
   <timeZone>     <!-- req, xs:string, POSIX time zone string -->       </timeZone>
</Time>
```

8.1.9/ISAPI/System/time/localTime

/ISAPI/System/time/localTime	General Resource v2.0
GET	
Description	It is used to get the device local time information.
Query	None
Inbound Data	None
Success Return	**ISO 8601 Date-Time String**
PUT	
Description	It is used to udpate the device local time information.
Query	None

Inbound Data	**ISO 8601 Date-Time String**
Success Return	**ResponseStatus**
Notes: An ISO 8601 Date/Time string is accepted and returned. If the date/time value has a time zone, the time is converted into the device's local time zone. If the device time mode is set to "ntp" setting this value has no effect.	

8.1.10 /ISAPI/System/time/timeZone

/ISAPI/System/time/timeZone	**General Resource v2.0**
GET	
Description	It is used to get the device time zone information.
Query	None
Inbound Data	None
Success Return	**Time zone string**
PUT	
Description	It is used to udpate the device time zone information.
Query	None
Inbound Data	**Time zone string**
Success Return	**ResponseStatus**

Notes:

Time zones are defined by POSIX 1003.1 section 8.3 time zone notations. Note that the value following the +/- is the amount of time that must be added to the local time to result in UTC.

Example:

EST+5EDT01:00:00,M3.2.0/02:00:00,M11.1.0/02:00:00

Defines eastern standard time as "EST" with a GMT-5 offset. Daylight savings time is called "EDT", is one hour later and begins on the second Sunday of March at 2am and ends on the first Sunday of November at 2am.

CET-1CEST01:00:00,M3.5.0/02:00:00,M10.5.0/03:00:00

Defines central European time as GMT+1 with a one-hour daylight savings time ("CEST") that starts on the last Sunday in March at 2am and ends on the last Sunday in October at 3am.
Check whether the device supports DST capability from 8.1.6 device capabilities

8.1.11 /ISAPI/System/time/NtpServers

/ISAPI/System/time/ntpServers	General Resource v2.0
GET	
Description	It is used to get the configuration of NTP servers for the device.
Query	None
Inbound Data	None
Success Return	**NTPServerList**
PUT	
Description	It is used to update the configuration of NTP servers for the device.
Query	None
Inbound Data	**NTPServerList**
Success Return	**ResponseStatus**
POST	
Description	It is used to add the configuration of a NTP server for the device.
Query	None
Inbound Data	**NTPServer**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the configuration of NTP servers for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: When the <timeMode> is set to "NTP", the servers in this list are used to synchronize the device's system time. To determine whether it is possible to dynamically create or delete ntp server, check the defined HTTP methods in /ISAPI/System/time/ntpServers/description.	

NTPServerList XML Block

```
<NTPServerList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <NTPServer/> <!-- opt -->
</ NTPServerList>
```

8.1.12 /ISAPI/System/time/ntpServers/<ID>

/ISAPI/System/time/ntpServers/*ID*	General Resource v2.0
GET	
Description	It is used to get the configuration of a NTP server for the device.

Query	None
Inbound Data	None
Success Return	**NTPServer**
PUT	
Description	It is used to update the configuration of a NTP server for the device.
Query	None
Inbound Data	**NTPServer**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the configuration of a NTP server for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Depending on the value of <addressingFormatType>, either the <hostName> or the IP address fields will be used to locate the NTP server.	

NTPServer XML Block

```
<NTPServer version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string; id -->  </id>
  <addressingFormatType>
    <!-- req, xs:string, "ipaddress,hostname"-->
  </addressingFormatType>
  <hostName>  <!-- dep, xs:string -->  </hostName>
  <ipAddress>  <!-- dep, xs:string -->  </ipAddress>
  <ipv6Address> <!-- dep, xs:string -->  </ipv6Address>
  <portNo>  <!-- opt, xs:integer --> </portNo>
  <synchronizeInterval> <!--opt, xs:integer, minutes --> </synchronizeInterval>
</NTPServer>
```

8.1.13 /ISAPI/System/Holidays

URI	**/ISAPI/System/Holidays**		**Type**	Resource
Function	Access the list of holidays			
Methods	**Query String(s)**	**Inbound Data**	**Return Result**	
GET			<holidayList >	
PUT		<holidayList>	<ResponseStatus>	

Notes	

holidayList XML Block

```
<HolidayList version="2.0" xmlns="http://urn:selfextension:ISAPIext-ver10-xsd">
  <holiday/> <!-- opt -->
</HolidayList>
```

8.1.14 /ISAPI/System/Holidays/<ID>

URI	/ISAPI/System/Holidays/ID/		Type	Resource
Function	Access a holiday.			
Methods	Query String(s)	Inbound Data	Return Result	
GET			<holiday >	
PUT		<holiday>	<ResponseStatus>	
Notes	<holidayMode> decides whether <holidayDate>,<holidayWeek>or <holidayMonth> is contained. <holidayMode>:date: example form May 5th,1900 to June 8th,1900. <holidayMode>:week: example form May 1st week to May 2nd week. <holidayMode>:month: example form May 1st to May 5th.			

holiday XML Block

```
<holiday version="2.0" xmlns="http://urn:selfextension:ISAPIext-ver10-xsd">
  <id>                    <!-- req, xs:string;id -->                    </id>
  <enabled>               <!-- req, xs:boolean -->                     </enabled>
  <holidayMode/>    <!-- req, xs:string, "date, weeek, month" -->  <holidayName>          <!-- req, xs:string -->  </holidayName>
  <holidayDate>  <!-- dep -->
   <startDate> <!-- req, xs:date --> </startDate>
   <endDate>    <!-- req, xs:date --> </endDate>
  </holidayDate>
  <holidayWeek>  <!-- dep -->
   <startWeek>    <!-- req -->
     <monthOfYear> <!-- req --> </monthOfYear>
     <sequence> <!-- req, xs:integer, 1…5 --> </sequence>
     <dayOfWeek>
```

```
        <!-- req, ISO8601 weekday number, 1=Monday" -->
      </dayOfWeek>
    </startWeek>
    <endWeek>     <!-- req -->
      <monthOfYear> <!-- req --> </monthOfYear>
      <sequence> <!-- req, xs:integer, 1…5 --> </sequence>
      <dayOfWeek>
            <!-- req, ISO8601 weekday number, 1=Monday" -->
      </dayOfWeek>
    </endWeek>
  </holidayWeek>
  <holidayMonth>    <!-- dep -->
   <startMonth>     <!-- req -->
    <monthOfYear> <!-- req, xs:integer, "1…12" --> </monthOfYear>
    <dayOfMonth> <!-- req, xs:integer, "1…31" --> </dayOfMonth>
   </startMonth>
   <endMonth>     <!-- req -->
    <monthOfYear> <!-- req, xs:integer, "1…12" --> </monthOfYear>
    <dayOfMonth> <!-- req, xs:integer, "1…31" -->    <dayOfMonth>
   </endMonth>
  </holidayMonth>
</holiday>
```

8.1.15 /ISAPI/System/upgradeStatus

/ISAPI/System/upgradeStatus	General Resource v2.0
GET	
Description	It is used to get upgrade status of the device.
Query	None
Inbound Data	None
Success Return	**upgradeStatus**
Notes:	

upgradeStatus XML Block

```
<upgradeStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <upgrading>     <!-- ro, req, xs:boolean --> </upgrading>
  <percent>  <!-- ro, req, xs:integer "0-100" -->     </percent>
```

```
</upgradeStatus>
```

8.2 /ISAPI/System/Network

/ISAPI/System/Network	Service	v2.0
Notes: Network configuration.		

8.2.1/ISAPI/System/Network/capabilities

/ISAPI/System/Network/capabilities	General Resource	v2.0
GET		
Description	It is used to get network capability.	
Query	None	
Inbound Data	None	
Success Return	**<NetworkCap>**	
Notes:		

NetworkCap XML Block

```
<NetworkCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <isSupportWireless> <!-- req, xs:boolean --> <isSupportWireless>
   <isSupportPPPoE> <!-- req, xs:boolean --> <isSupportPPPoE>
   <isSupportBond> <!-- req, xs:boolean -->    <isSupportBond>
   <isSupport802_1x> <!-- req, xs:boolean --> </isSupport802_1x>
   <isSupportNtp> <!-- opt, xs:boolean -->    </ isSupportNtp>
   <isSupportFtp> <!-- opt, xs:boolen --> </isSupportFtp>
   < isSupportUpnp> <!-- opt, xs:boolean -->    </ isSupportUpnp>
   < isSupportPNP> <!-- opt, xs:boolean -->    </isSupportPNP>
   < isSupportDdns> <!-- opt, xs:boolean -->    </ isSupportDdns>
   <isSupportHttps> <!-- opt, xs:boolean -->    </isSupportHttps>
   <SnmpCap><!-- opt -->
        <isSupport> <!-- req, xs:boolean --> </isSupport>
   </SnmpCap>
   <isSupportExtNetCfg> <!-- opt, xs:boolean -->    </isSupportExtNetCfg>
   <isSupportIPFilter> <!-- opt, xs:boolean -->    </isSupportIPFilter>
</NetworkCap>
```

8.2.2/ISAPI/System/Network/interfaces

/ISAPI/System/Network/interfaces	General Resource v2.0
GET	
Description	It is used to get the device network interfaces.
Query	None
Inbound Data	None
Success Return	**NetworkInterfaceList**
Notes: As hardwired system resources, network interfaces cannot be created or destroyed.	

NetworkInterfaceList XML Block

```
<NetworkInterfaceList version="2.0"
  xmlns="http://www.isapi.org/ver20/XMLSchema">
    <NetworkInterface/>   <!-- opt -->
</NetworkInterfaceList>
```

8.2.3/ISAPI/System/Network/interfaces/<ID>

/ISAPI/System/Network/interfaces/*ID*			General Resource v2.0
GET			
Description	It is used to get a particular network interface.		
Query	None		
Inbound Data	None		
Success Return	**NetworkInterface**		
PUT			
Description	It is used to update a particular network interface.		
Query	None		
Inbound Data	**NetworkInterface**		
Success Return	**ResponseStatus**		
Error Status Code	**statusCode**	**subStatusCode**	**Description**
	6	**badIPv6Address**	**error IPv6 address**
	6	**conflictIPv6Address**	**conflictIPv6Address**
	6	**badNetMask**	**error subnet mask**
	6	**conflictIPv4Address**	**conflictIPv4Address**
	6	**badIPv4Address**	**error IPv4 address**
Notes: defaultConnection: default network connection, required when device has more than one			

interface.

NetworkInterface XML Block

```
<NetworkInterface version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id>          <!-- req, xs:string -->       </id>
   <IPAddress/>    <!-- req -->
   <Wireless/>      <!-- opt -->
   <Discovery/>    <!-- opt -->
   <Link /> <!-- opt -->
   <defaultConnection> <!-- opt, xs:boolean--> </defaultConnection>
</NetworkInterface>
```

8.2.4/ISAPI/System/Network/interfaces/<ID>/ipAddress

/ISAPI/System/Network/interfaces/*ID*/ipAddress			**General Resource v2.0**
GET			
Description	It is used to get the ip address of a particular network interface.		
Query	None		
Inbound Data	None		
Success Return	**IPAddress**		
PUT			
Description	It is used to update the ip address of a particular network interface.		
Query	None		
Inbound Data	**IPAddress**		
Success Return	**ResponseStatus**		
Error Status Code	**statusCode**	**subStatusCode**	**Description**
	6	**badIPv6Address**	**error IPv6 address**
	6	**conflictIPv6Address**	**conflictIPv6Address**
	6	**badNetMask**	**error subnet mask**
	6	**conflictIPv4Address**	**conflictIPv4Address**
	6	**badIPv4Address**	**error IPv4 address**

Notes:

If <addressingType> is dynamic, fields below it need not be provided.
If <addressingType> is dynamic, a DHCP client is used for the device.
If <addressingType> is static the device IP address is configured manually and the gateway and DNS fields are optional.
If <addressingType> refers to APIPA, the device IP address is automatically configured without DHCP. In this case the gateway and DNS fields are optional.

Use of <ipAddress> or <ipv6Address> in fields is dictated by the <ipVersion> field. If <ipVersion> is “v4” the <ipAddress> fields are used; if <ipVersion> is “v6” the <ipv6Address> fields are used. If <ipVersion> is "dual", both <ipAddress> and <ipv6Address> fields may be used.
<subnetMask> notation is “ISAPI.ISAPI.ISAPI.ISAPI”.
<IPV6Address> is “ISAPIx:ISAPIx:ISAPIx:ISAPIx:ISAPIx:ISAPIx:ISAPIx:ISAPIx” using CIDR notation.

IPAddress XML Block

```
<IPAddress version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <ipVersion>      <!-- req, xs:string, “v4,v6,dual” --></ipVersion>
  <addressingType>  <!-- req, xs:string, “static,dynamic,apipa” --> </addressingType>
  <ipAddress>     <!-- dep, xs:string -->  </ipAddress>
  <subnetMask> <!-- dep, xs:string, subnet mask for IPv4 address -->  </subnetMask>
  <ipv6Address> <!-- dep, xs:string -->  </ipv6Address>
  <bitMask> <!-- dep, xs:integer, bitmask IPv6 address -->  </bitMask>
  <DefaultGateway>  <!-- dep -->
    <ipAddress> <!-- dep, xs:string -->  </ipAddress>
    <ipv6Address> <!-- dep, xs:string -->      </ipv6Address>
  </DefaultGateway>
  <PrimaryDNS> <!-- dep -->
    <ipAddress> <!-- dep, xs:string -->  </ipAddress>
    <ipv6Address> <!-- dep, xs:string -->      </ipv6Address>
  </PrimaryDNS>
  <SecondaryDNS>   <!-- dep -->
    <ipAddress> <!-- dep, xs:string -->  </ipAddress>
    <ipv6Address> <!-- dep, xs:string -->      </ipv6Address>
  </SecondaryDNS>
  <Ipv6Mode>     <!-- opt -->
    <ipV6AddressingType>
      <-- dep, xs:string,"ra,manual,dhcp>
    </ipV6AddressingType>
    <ipv6AddressList>
      <v6Address>
        <id> <!-- dep, xs:string;id --> </id>
        <type> <-- dep, xs:string,"ra,manual,dhcp>   </type>
        <address>   <!-- dep, xs:string -->   </address>
        <bitMask><!-- dep, xs:integer -->     </bitMask>
      </v6Address>
    </ipv6AddressList>
  </Ipv6Mode>
</IPAddress>
```

8.2.5/ISAPI/System/Network/interfaces/<ID>/wireless

/ISAPI/System/Network/interfaces/*ID*/wireless	General Resource v2.0
GET	
Description	It is used to get the wireless settings of a particular network interface.
Query	None
Inbound Data	None
Success Return	**Wireless**
PUT	
Description	It is used to update the wireless settings of a particular network interface.
Query	None
Inbound Data	**Wireless**
Success Return	**ResponseStatus**
Notes:	

Wireless XML Block

```
<Wireless version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled>  <!-- req, xs:boolean -->  </enabled>
  <wirelessNetworkMode>
    <!-- opt, xs:string, "infrastructure,adhoc" -->
  </wirelessNetworkMode>
  <channel>  <!-- opt, xs:string, "1-14,auto" -->  </channel>
  <ssid>  <!-- opt, xs:string -->  </ssid>
  <wmmEnabled>  <!-- opt, xs:boolean -->  </wmmEnabled>
  <WirelessSecurity> <!-- opt -->
    <securityMode>
      <!-- opt, xs:string,
        "disable,WEP,WPA-personal,WPA2-personal,WPA-RADIUS,
         WPA-enterprise,WPA2-enterprise"-->
    </securityMode>
    <WEP>
      <!-- dep, depends on <securityMode> -->
      <authenticationType>
        <!-- req, xs:string, "open,sharedkey,auto" -->
      </authenticationType>
      <defaultTransmitKeyIndex>
       <!-- req, xs:integer -->
      </defaultTransmitKeyIndex>
      <wepKeyLength> <!-- opt, xs:integer "64,128" --> </wepKeyLength>
```

```
            <EncryptionKeyList>
              <encryptionKey>
                <!-- req, xs:hexBinary, WEP encryption key in hexadecimal format -->
              </encryptionKey>
            </EncryptionKeyList>
          </WEP>
          <WPA>
            <!-- dep, depends on <securityMode> -->
            <algorithmType>  <!-- req, xs:string, “TKIP,AES,TKIP/AES”--> </algorithmType>
            <sharedKey><!-- req, xs:string, pre-shared key used in WPA --> </sharedKey>
            <wpaKeyLength> <!-- req, xs: integer, “8-63”--> </wpaKeyLength>
          </WPA>
        </WirelessSecurity>
      </Wireless>
```

8.2.6/ISAPI/System/Network/interfaces/<ID>/wireless/accessPointList

/ISAPI/System/Network/interfaces/*ID*/wireless/accessPointList	**General Resource v2.0**
GET	
Description	It is used to get the valid access points on the wireless interface.
Query	None
Inbound Data	None
Success Return	**accessPointList**
Notes:	

accessPointList XML Block

```
<accessPointList version=“2.0”    xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <accessPoint/>
</accessPointList
```

8.2.7/ISAPI/System/Network/interfaces/<ID>/wireless/accessPointList/<ID>

/ISAPI/System/Network/interfaces/*ID*/wireless/accessPointList/ID	**General Resource v2.0**

GET	
Description	It is used to get a particular access point.
Query	None
Inbound Data	None
Success Return	**accessPoint**
Notes:	

accessPoint XML Block

```
<accessPoint version="2.0"    xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id>   <!-- req, xs:integer-->       </id>
   <networkMode>
      <!-- opt, xs:string, "infrastructure,adhoc" -->
   </networkMode>
   <channel> <!-- opt, xs:string, "1-14,auto" --> </channel>
   <ssid> <!-- req, xs:string --> </ssid>
   <speed> <!-- opt, xs:Integer, in Mbps--></speed>
   <signalStrength><!-- opt, xs:Integer,"0-100">< /signalStrength>
   <securityMode>
      <!-- req, xs:string, "disable,WEP,WPA-personal,WPA2-personal,WPA-RADIUS,
         WPA-enterprise,WPA2-enterprise" -->
   </securityMode>
</accessPoint>
```

8.2.8/ISAPI/System/Network/interfaces/<ID>/discovery

/ISAPI/System/Network/interfaces/*ID*/discovery	General Resource v2.0
GET	
Description	It is used to get the discovery settings of a particular network interface.
Query	None
Inbound Data	None
Success Return	**Discovery**
PUT	
Description	It is used to update the discovery settings of a particular network interface.
Query	None
Inbound Data	**Discovery**
Success Return	**ResponseStatus**

Notes:

Discovery XML Block

```
<Discovery version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <UPnP>            <!-- req -->
      <enabled>      <!-- req, xs:boolean -->   </enabled>
   </UPnP>
   <Zeroconf>        <!-- opt -->
      <enabled>      <!-- req, xs:boolean -->   </enabled>
   </Zeroconf>
</Discovery>
```

8.2.9/ISAPI/System/Network/interfaces/<ID>/Link

/ISAPI/System/Network/interfaces/*ID*/link	**General Resource v2.0**
GET	
Description	It is used to get the link layer settings of a particular network interface.
Query	None
Inbound Data	None
Success Return	**Link**
PUT	
Description	It is used to update the link layer settings of a particular network interface.
Query	None
Inbound Data	**Link**
Success Return	**ResponseStatus**
Notes:	

Link XML Block

```
<Link xmlns="http://www.isapi.org/ver20/XMLSchema">
   <MACAddress> <!-- req, xs:string> </MACAddress>
   <autoNegotiation> <!-- req, xs:boolean> </autoNegotiation>
   <speed> <!-- req, xs:integer, "10, 100, 1000" --><speed>
   <duplex> <!-- req, xs:string, "half, full"> </duplex>
   <MTU> <!-- req, xs:integer --> </MTU>
</Link>
```

8.2.10 Examples

Example: Getting the Network Settings

```
GET /ISAPI/System/Network/interfaces HTTP/1.1
…
HTTP/1.1 200 OK
Content-Type: application/xml; charset="UTF-8"
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8"?>
<NetworkInterfaceList version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <NetworkInterface>
    <id>1</id>
    <IPAddress>
      <ipVersion>v4</ipVersion>
      <addressingType>static</addressingType>
      <ipAddress>172.6.64.7</ipAddress>
      <subnetMask>255.255.255.0</subnetMask>
      <DefaultGateway>
        <ipAddress>172.6.64.1</ipAddress>
      </DefaultGateway>
      <PrimaryDNS>
        <ipAddress>192.0.0.200</ipAddress>
      </PrimaryDNS>
    </IPAddress>
    <Discovery>
      <UPnP>
        <enabled>true</enabled>
      </UPnP>
      <Zeroconf>
        <enabled>true</enabled>
      </Zeroconf>
    </Discovery>
    <Link>
      <MACAddress> 00:40:48:4C:7F:F2</MACAddress>
      <autoNegotiation>true</autoNegotiation>
      <speed>1000<speed>
      <duplex>full</duplex>
      <MTU>1500</MTU>
    </Link>
  <NetworkInterface>
```

```
</NetworkInterfaceList>
```

Example: Setting the IP Address

```
PUT /ISAPI/System/Network/interfaces/1/ipAddress HTTP/1.1
…
Content-Type: application/xml; charset="UTF-8"
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8"?>
<IPAddress version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <ipVersion>v4</ipVersion>
   <addressingType>static</addressingType>
   <ipAddress>172.6.64.16</ipAddress>
   <subnetMask>255.255.255.0</subnetMask>
   <DefaultGateway>
      <ipAddress>172.6.64.1</ipAddress>
   </DefaultGateway>
   <PrimaryDNS>
      <ipAddress>192.0.0.200</ipAddress>
   </PrimaryDNS>
</IPAddress>

HTTP/1.1 200 OK
…
Content-Type: application/xml; charset="UTF-8"
Content-Length:xxx

<?xml version="1.0" encoding="UTF-8"?>
<ResponseStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <requestURL>/Network/interfaces/1/ipAddress</requestURL>
   <statusCode>1</statusCode>
   <statusString>OK</statusString>
</ResponseStatus>
```

8.2.11 /ISAPI/System/Network/interfaces/<ID>/WPS

/ISAPI/System/Network/interfaces/*ID*/WPS	**General Resource**	**v2.0**
GET		

Description	It is used to access WPS configuratioin
Query	None
Inbound Data	None
Success Return	**WPS**
PUT	
Description	It is used to access WPS configuratioin
Query	None
Inbound Data	**WPS**
Success Return	**ResponseStatus**
Notes:	

WPSXML Block

```
<WPS version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <enable> <!-- req, xs:boolean--> </enable>
</WPS>
```

8.2.12 /ISAPI/System/Network/interfaces/ID/WPS/AutoConnect

/ISAPI/System/Network/interfaces/ID/WPS/AutoConnect	**General Resource**	**v2.0**
PUT		
Description	It is used to WPS auto connection mode	
Query	None	
Inbound Data	**WPS**	
Success Return	**ResponseStatus**	
Notes:		

8.2.13 /ISAPI/System/Network/interfaces/ID/WPS/devicePinCode

/ISAPI/System/Network/interfaces/ID/WPS/devicePinCode	**General Resource**	**v2.0**
GET		
Description	It is used to get WPS device PIN code	
Query	None	

Inbound Data	**None**
Success Return	**PIN code string**
Notes:	

8.2.14 /ISAPI/System/Network/interfaces/ID/WPS/devicePinCodeUpdate

/ISAPI/System/Network/interfaces/ID/WPS/devicePinCodeUpdate	**General Resource**	**v2.0**
GET		
Description	It is used to generate a new device PIN code	
Query	None	
Inbound Data	**None**	
Success Return	**PIN code string**	
Notes:		

8.2.15 /ISAPI/System/Network/interfaces/ID/WPS/ApPinCode

/ISAPI/System/Network/interfaces/ID/WPS/ApPinCode	**General Resource**	**v2.0**
GET		
Description	It is used to access WPS configuratioin	
Query	None	
Inbound Data	None	
Success Return	**WpsApPincode**	
PUT		
Description	It is used to access WPS configuratioin	
Query	None	
Inbound Data	**WpsApPincode**	
Success Return	**ResponseStatus**	
Notes:		

WpsApPincodeXML Block

```
<WpsApPincode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ssid> <!-- req, xs:string --> </ssid>
  <pinCode> <!-- req, xs:string --> </pinCode>
</WpsApPincode>
```

8.2.16 /ISAPI/System/Network/interfaces/ID/ieee802.1x

/ISAPI/System/Network/interfaces/ID/ieee802.1x	General Resource v2.0
GET	
Description	It is used to access IEEE 802.1x settings
Query	None
Inbound Data	None
Success Return	**IEEE802_1x**
PUT	
Description	It is used to configure IEEE 802.1x settings
Query	None
Inbound Data	**IEEE802_1x**
Success Return	**ResponseStatus**

Notes:

If the <authenticatonProtocolType> tag corresponds to "EAP-TTLS", then the <innerTTLSAuthenticationMethod> tag must be provided.
If the <authenticationProtocolType> corresponds to "EAP-PEAP" or "EAP-FAST", then the <innerEAPProtocolType> tag must be provided.
The <anonymousID> tag is optional. If the <authenticationProtocolType> corresponds to "EAP-FAST", then the <autoPACProvisioningEnabled> tag must be provided.
<anonymousID> is the optional anonymous ID to be used in place of the <userName>.

IEEE802_1x XML Block

```
<IEEE802_1x version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <enabled> <!-- req, xs:boolean -->   </enabled>
   <authenticationProtocolType>
     <!-- req, xs:string, "EAP-TLS,EAP-TTLS,EAP-PEAP,EAP-LEAP,EAP-FAST,EAP-MD5"
-->
  </authenticationProtocolType>
   <innerTTLSAuthenticationMethod>
     <!-- dep, xs:string, "MS-CHAP,MS-CHAPv2,PAP,EAP-MD5" -->
  </innerTTLSAuthenticationMethod>
   <innerEAPProtocolType>
```

```
    <!-- dep, xs:string, “EAP-POTP,MS-CHAPv2” -->
  </innerEAPProtocolType>
  <validateServerEnabled>      <!-- dep, xs:boolean -->    </validateServerEnabled>
  <userName>    <!-- dep, xs:string -->   </userName>
  <password>     <!-- dep, xs:string -->   </password>
  <anonymousID>      <!-- opt, xs:string -->    </anonymousID>
  <autoPACProvisioningEnabled> <!-- dep, xs:boolean -->
</autoPACProvisioningEnabled>
  <Extensions> <!-- opt -->
    <EAPOLVersion xmlns=”http://www.isapi.org/ver20/XMLSchema”>
       <!--opt, xs:string, “1, 2”-->
    </EAPOLVersion>
  </Extensions>
</IEEE802_1x>
```

8.2.17 /ISAPI/System/Network/PPPoE

/ISAPI/System/Network/PPPoE	General Resource v2.0
GET	
Description	It is used to get the configurations of pppoe.
Query	None
Inbound Data	None
Success Return	**PPPoEList**
PUT	
Description	It is used to set the configurations of pppoe.
Query	None
Inbound Data	**PPPoEList**
Success Return	**ResponseStatus**
Notes:	

PPPoEList XML Block

```
<PPPoEList xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <PPPoE/> <!--req-->
</PPPoEList>
```

8.2.18 /ISAPI/System/Network/PPPoE/status

/ISAPI/System/Network/PPPoE/status	General Resource v2.0
GET	

Description	It is used to get the status of pppoe.
Query	None
Inbound Data	None
Success Return	**PPPoEStatusList**
Notes:	

PPPoEStatusList XML Block

```
<PPPoEStatusList xmlns="http://www.isapi.org/ver20/XMLSchema">
  <PPPoEStatus/> <!--req-->
</PPPoEStatusList>
```

8.2.19 /ISAPI/System/Network/PPPoE/<ID>

/ISAPI/System/Network/PPPoE/ID	**General Resource v2.0**
GET	
Description	It is used to get the configuration of a particular pppoe.
Query	None
Inbound Data	None
Success Return	**PPPoE**
PUT	
Description	It is used to set the configurations of a particular pppoe.
Query	None
Inbound Data	**PPPoE**
Success Return	**ResponseStatus**
Notes: <ethernetIfId> links the PPPoE to a network interface that the PPPoE dial up used, see /ISAPI/System/Network/interfaces/<ID>.	

PPPoE XML Block

```
<PPPoE xmlns="http://www.isapi.org/ver20/XMLSchema">
<id>              <!-- req, xs:string -->   </id>
  <enabled>       <!-- req, xs:boolean -->   </enabled>
  <ethernetIfId>       <!-- opt, xs:string; id -->        </ethernetIfId>
  <userName>     <!-- req, xs:string -->     </userName>
  <password>     <!-- wo, req, xs:string -->   </password>
</PPPoE>
```

8.2.20 /ISAPI/System/Network/PPPoE/<ID>/status

/ISAPI/System/Network/PPPoE/ID/status	General Resource v2.0
GET	
Description	It is used to get the status of a particular pppoe.
Query	None
Inbound Data	None
Success Return	**PPPoEStatus**
Notes:	

PPPoEStatus XML Block

```
<PPPoEStatus xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string --> </id>
  <enabled> <!-- req, xs:boolean --> </enabled>
  <ethernetIfId> <!-- opt, xs:string; id --> </ethernetIfId>
  <ipAddress> <!-- dep, xs:string --> </ipAddress>
  <subnetMask> <!-- dep, xs:string, subnet mask for IPv4 address --> </subnetMask>
  <ipv6Address> <!-- dep, xs:string --> </ipv6Address>
  <bitMask> <!-- dep, xs:integer, bitmask IPv6 address --> </bitMask>
  <DefaultGateway> <!-- dep -->
    <ipAddress> <!-- dep, xs:string -->  </ipAddress>
    <ipv6Address> <!-- dep, xs:string --> </ipv6Address>
  </DefaultGateway>
  <PrimaryDNS> <!-- dep -->
    <ipAddress> <!-- dep, xs:string --> </ipAddress>
    <ipv6Address> <!-- dep, xs:string --> </ipv6Address>
  </PrimaryDNS>
  <SecondaryDNS> <!-- dep -->
    <ipAddress> <!-- dep, xs:string -->  </ipAddress>
    <ipv6Address> <!-- dep, xs:string --> </ipv6Address>
  </SecondaryDNS>
</PPPoEStatus>
```

8.2.21 /ISAPI/System/Network/Bond

URI	/ISAPI/System/Network/Bond		Type	Service
Function	Get or set the configuration information of Bond net interfaces.			
Methods	Query String(s)	Inbound Data	Return Result	

GET			<BondList>
Notes	Bond NIC configuration		

BondList XML Block

```
<BondList version="2.0" xmlns="urn:selfextension:ISAPIext-ver10-xsd">
  <Bond>
</BondList>
```

8.2.22 /ISAPI/System/Network/Bond/<ID>

URI	**/ISAPI/System/Network/Bond/ID**	**Type**	Resource
Function	Get or set the configuration information of Bond net interface		
Methods	**Query String(s)**	**Inbound Data**	**Return Result**
GET			<Bond>
PUT		<Bond>	<ResponseStatus>
Notes			

Bond XML Block

```
<Bond version="2.0" xmlns="urn:selfextension:ISAPIext-ver10-xsd">
  <id>              <!-- req, xs:string -->   </id>
   <enabled>       <!-- req, xs:boolean -->   </enabled>
   <workMode> <!-- req, xs:string;"balance-rr, active-backup" --> </workMode>
   <primaryIf> <!-- req, xs:string;id --></primaryIf>
   <slaveIfList> <!-- req -->
      <ethernetIfId>       <!-- req, xs:string; id -->        </ethernetIfId>
   </slaveIfList>
   <IPAddress>
      <ipVersion>           <!-- req, xs:string, "v4,v6,dual" --></ipVersion>
      <addressingType>   <!-- req, xs:string, "static,dynamic,apipa" --> </addressingType>
      <ipAddress>           <!-- dep, xs:string -->              </ipAddress>
      <subnetMask>     <!-- dep, xs:string, subnet mask for IPv4 address -->   </subnetMask>
      <ipv6Address>        <!-- dep, xs:string -->             </ipv6Address>
      <bitMask>             <!-- dep, xs:integer, bitmask IPv6 address -->    </bitMask>
      <DefaultGateway>   <!-- dep -->
         <ipAddress>        <!-- dep, xs:string -->             </ipAddress>
         <ipv6Address>       <!-- dep, xs:string -->           </ipv6Address>
      </DefaultGateway>
```

```
    <PrimaryDNS>       <!-- dep -->
      <ipAddress>      <!-- dep, xs:string -->          </ipAddress>
      <ipv6Address>    <!-- dep, xs:string -->          </ipv6Address>
    </PrimaryDNS>
    <SecondaryDNS>     <!-- dep -->
      <ipAddress>      <!-- dep, xs:string -->          </ipAddress>
      <ipv6Address>    <!-- dep, xs:string -->          </ipv6Address>
    </SecondaryDNS>
  </IPAddress>
  <Link xmlns="urn:selfextension:ISAPIext-ver10-xsd">          <!-- opt -->
    <MACAddress> <!-- req, xs:string> </MACAddress>
    <autoNegotiation> <!-- req, xs:boolean> </autoNegotiation>
    <speed> <!-- req, xs:integer, "10, 100, 1000" --><speed>
    <duplex> <!-- req, xs:string, "half, full"> </duplex>
    <MTU> <!-- req, xs:integer --> </MTU>
  </Link>
</Bond>
```

8.2.23 /ISAPI/System/Network/extension

URI	/ISAPI/System/Network/extension		Type	Resource
Function	Get or set the configuration information of network externsion			
Methods	Query String(s)	Inbound Data	Return Result	
GET			<networkExtension>	
PUT		<networkExtension>	<ResponseStatus>	
Notes				

networkExtension XML Block

```
<networkExtension version="2.0" xmlns="urn:selfextension:ISAPIext-ver10-xsd">
  <multicastAddress> <!-- opt -->
    <ipVersion>          <!-- req, xs:string, "v4,v6,dual" --></ipVersion>
    <ipAddress>    <!-- dep, xs:string -->     </ipAddress>
    <ipv6Address>    <!-- dep, xs:string -->  </ipv6Address>
  </multicastAddress>
  <enVirtualHost> <!--opt, xs:Boolean --> <enVirtualHost>
</networkExtension>
```

8.2.24 /ISAPI/System/Network/DDNS

/ISAPI/System/Network/DDNS	General Resource v2.0
GET	
Description	It is used to get the configurations of DDNS.
Query	None
Inbound Data	None
Success Return	**DDNSList**
PUT	
Description	It is used to set the configurations of pppoe.
Query	None
Inbound Data	**DDNSList**
Success Return	**ResponseStatus**
Notes:	

DDNSList XML Block

```
<DDNSList xmlns="http://www.isapi.org/ver20/XMLSchema">
   <DDNS/> <!--req-->
</DDNSList>
```

8.2.25 /ISAPI/System/Network/DDNS/<ID>

/ISAPI/System/Network/DDNS/ID	General Resource v2.0
GET	
Description	It is used to get the configuration of a particular DDNS.
Query	None
Inbound Data	None
Success Return	**DDNS**
PUT	
Description	It is used to set the configurations of a particular pppoe.
Query	None
Inbound Data	**DDNS**
Success Return	**ResponseStatus**
Notes: <serverAddress> DDNS server's address. Depending on the value of <addressingFormatType>, either the <hostName> or the IP	

address fields will be used to locate the NTP server.
Use of IPv4 or IPv6 addresses depends on the value of the <ipVersion> field in /ISAPI/System/Network/interfaces/ID/ipAddress.
When <provider> is “IPServer”, <serverIPAddress> is required.
When <provider> is “DysDNS”, all fields are required except the <portNo>.
When <provider> is “PeanutHall”, all fields are required except the <serverIPAddress> and <portNo>.
<deviceDomainName> the device’s domain name.
<password> is a write-only field.

DDNS XML Block

```
<DDNS version=“2.0” xmlns=”http://www.isapi.org/ver20/XMLSchema”>
   <id> <!-- req, xs:string -->
   <enabled>     <!-- req, xs:boolean --> </enabled>
   <provider>
      <!-- req, xs:string, “IPServer, DynDNS, PeanutHall, HiDDNS …” -->
   </provider>
   <serverAddress>
      <addressingFormatType>
         <!-- req, xs:string, “ipaddress,hostname”-->
      </addressingFormatType>
      <hostName>    <!-- dep, xs:string -->      </hostName>//不能是中文
      <ipAddress>    <!-- dep, xs:string -->      </ipAddress>
      <ipv6Address>    <!-- dep, xs:string -->  </ipv6Address>
   <serverAddress>
   <portNo> <!-- opt, xs:integer --> </portNo>
   <deviceDomainName> <!-- dep, xs:string --> </deviceDomainName>
   <userName> <!-- dep, xs:string --> </userName>//不能是中文
   <password> <!-- wo, dep, xs:string --></password>
</DDNS>
```

8.2.26 /ISAPI/System/Network/SNMP

/ISAPI/System/Network/SNMP	**General Resource v2.0**
GET	
Description	Get SNMP Settings.
Query	None
Inbound Data	None
Success Return	**SNMP**
PUT	
Description	Set SNMP Settings

Query	None
Inbound Data	**SNMP**
Success Return	**ResponseStatus**
Notes: At least one of the <SNMPv2c> block or <SNMPAdvanced> block must be provided. <snmpPort> snmp agent listen port	

SNMP XML Block

```
<SNMP version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <SNMPv1c/>
    <!-- dep, choose one mode in <SNMPv1c> <SNMPv2c> <SNMPAdvanced> is required
-->
   < SNMPv2c/>          <!-- dep -->
   <SNMPAdvanced/>    <!-- dep -->
   <listenPort> <!--opt, xs:integer ,snmp port--><listenPort>
</SNMP>
```

8.2.27 /ISAPI/System/Network/SNMP/v1c

/ISAPI/System/Network/SNMP/v1c	**General Resource v2.0**
GET	
Description	Get SNMP v1c parameters.
Query	None
Inbound Data	None
Success Return	**SNMPv1c**
PUT	
Description	Set SNMP v1c parameters
Query	None
Inbound Data	**SNMPv1c**
Success Return	**ResponseStatus**
Notes: SNMP v1c configuration includes SNMP notification parameters and a set of SNMP trap receivers. SNMP v1c comprises SNMP v1 without the controversial new SNMP v1 security model, using instead the simple community-based security scheme of SNMP v1	

SNMPv1c XML Block

```
<SNMPv1c version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
```

```
<notificationEnabled>          <!-- req, xs:boolean -->    </notificationEnabled>
<SNMPTrapReceiverList/>        <!-- opt -->
<enabled> <!--req, xs:boolean; is enabled snmpv2c--> </enabled>
<writeCommunity> <!--req, xs:string --> </writeCommunity>
<readCommunity> <!-- req, xs:string --> </readCommunity>
</SNMPv1c>
```

8.2.28 /ISAPI/System/Network/SNMP/v1c/trapRecevivers

/ISAPI/System/Network/SNMP/v1c/trapRecevivers	General Resource v2.0
GET	
Description	Get SNMP trap receiver list.
Query	None
Inbound Data	None
Success Return	**SNMPTrapReceiverList**
PUT	
Description	Set SNMP trap receiver list
Query	None
Inbound Data	**SNMPTrapReceiverList**
Success Return	**ResponseStatus**
POST	
Description	create a new SNMP trap receiver
Query	None
Inbound Data	**SNMPTrapReceiver**
Success Return	**ResponseStatus**
DELETE	
Description	Delete SNMP trap receiver list
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

8.2.29 /ISAPI/System/Network/SNMP/v1c/trapReceiver/<ID>

/ISAPI/System/Network/SNMP/v2c/trapRecevivers/<ID>	General Resource v2.0
GET	
Description	Get SNMP trap receiver information.
Query	None
Inbound Data	None
Success Return	**SNMPTrapReceiver**
PUT	
Description	Set SNMP trap receiver information
Query	None
Inbound Data	**SNMPTrapReceiver**
Success Return	**ResponseStatus**
DELETE	
Description	Delete SNMP trap receiver
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

8.2.30 /ISAPI/System/Network/SNMP/v2c

/ISAPI/System/Network/SNMP/v2c	General Resource v2.0
GET	
Description	Get SNMP v2c parameters.
Query	None
Inbound Data	None
Success Return	**SNMPv2c**
PUT	
Description	Set SNMP v2c parameters
Query	None
Inbound Data	**SNMPv2c**
Success Return	**ResponseStatus**
Notes: SNMP v2c configuration includes SNMP notification parameters and a set of SNMP trap receivers.	

SNMP v2c comprises SNMP v2 without the controversial new SNMP v2 security model, using instead the simple community-based security scheme of SNMP v1

SNMPv2c XML Block

```
<SNMPv2c version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <notificationEnabled>  <!-- req, xs:boolean -->  </notificationEnabled>
  <SNMPTrapReceiverList/>  <!-- opt -->
  <enabled> <!--req, xs:boolean; is enabled snmpv2c--> </enabled>
  <writeCommunity> <!--req, xs:string --> </writeCommunity>
  <readCommunity> <!-- req, xs:string --> </readCommunity>
</SNMPv2c>
```

8.2.31 /ISAPI/System/Network/SNMP/v2c/trapRecevivers

/ISAPI/System/Network/SNMP/v2c/trapRecevivers	General Resource v2.0
GET	
Description	Get SNMP trap receiver list.
Query	None
Inbound Data	None
Success Return	**SNMPTrapReceiverList**
PUT	
Description	Set SNMP trap receiver list
Query	None
Inbound Data	**SNMPTrapReceiverList**
Success Return	**ResponseStatus**
POST	
Description	create a new SNMP trap receiver
Query	None
Inbound Data	**SNMPTrapReceiver**
Success Return	**ResponseStatus**
DELETE	
Description	Delete SNMP trap receiver list
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

SNMPTrapReceiverList XML Block

```
<SNMPTrapReceiverList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SNMPTrapReceiver/>  <!-- opt -->
</SNMPTrapReceiverList>
```

8.2.32 /ISAPI/System/Network/SNMP/v2c/trapReceiver/<ID>

/ISAPI/System/Network/SNMP/v2c/trapRecevivers/<ID>	General Resource v2.0
GET	
Description	Get SNMP trap receiver information.
Query	None
Inbound Data	None
Success Return	**SNMPTrapReceiver**
PUT	
Description	Set SNMP trap receiver information
Query	None
Inbound Data	**SNMPTrapReceiver**
Success Return	**ResponseStatus**
DELETE	
Description	Delete SNMP trap receiver
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

SNMPTrapReceiver XML Block

```
<SNMPTrapReceiver version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>                    <!-- req, xs:string;id -->                    </id>
  <ReceiverAddress/>      <!-- req -->
  <notificationType/>     <!-- req, xs:string, "trap,inform" -->
  <communityString>       <!-- opt, xs:string -->          </communityString>
</SNMPTrapReceiver>
```

8.2.33 /ISAPI/System/Network/SNMP/advanced

/ISAPI/System/Network/SNMP/advanced	General Resource	v2.0

GET	
Description	Get SNMP Advanced parameters.
Query	None
Inbound Data	None
Success Return	**SNMPAdvanced**
PUT	
Description	Set SNMP Advanced parameters
Query	None
Inbound Data	**SNMPAdvanced**
Success Return	**ResponseStatus**
Notes: <localEngineID> is a hexadecimal string indicating the local device engine ID. <authenticationNotificationEnabled> indicates if SNMP authentication failure notification is enabled on the device. <SNMPNotificationFilterList> is a list to filter traps based on OIDs	

SNMPAdvanced XML Block

```
<SNMPAdvanced version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <localEngineID> <!-- req, xs:hexBinary, see RFC2571 --> </localEngineID>
  <authenticationNotificationEnabled>
    <!-- opt, xs:boolean -->
  </authenticationNotificationEnabled>
  <SNMPUserList/> <!-- opt -->
  <SNMPNotificationFilterList/> <!-- opt -->
  <notificationEnabled> <!-- opt, xs:boolean --> </notificationEnabled>
  <SNMPNotificationReceiverList/> <!-- opt -->
  <enabled> <!--req, xs:boolean --> </enabled>
</SNMPAdvanced>
```

8.2.34 /ISAPI/System/Network/SNMP/advanced/users

/ISAPI/System/Network/SNMP/advanced/users	General Resource v2.0
GET	
Description	Get SNMP advanced user list.
Query	None
Inbound Data	None
Success Return	**SNMPUserList**

PUT	
Description	Set SNMP advanced list
Query	None
Inbound Data	**SNMPUserList**
Success Return	**ResponseStatus**
POST	
Description	create a new SNMP advanced user
Query	None
Inbound Data	**SNMPUser**
Success Return	**ResponseStatus**
DELETE	
Description	Delete SNMP advanced user list
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: Defines the set of SNMP users and their permissions.	

SNMPUserList XML Block

```
<SNMPUserList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SNMPUser/>  <!-- opt -->
</SNMPUserList>
```

8.2.35 /ISAPI/System/Network/SNMP/advanced/users/<ID>

/ISAPI/System/Network/SNMP/advanced/users/ID	General Resource v2.0
GET	
Description	Get SNMP advanced user information.
Query	None
Inbound Data	None
Success Return	**SNMPUser**
PUT	
Description	Set SNMP advanced user information
Query	None
Inbound Data	**SNMPUser**
Success Return	**ResponseStatus**
DELETE	

Description	Delete SNMP advanced user
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:

<remoteEngineID> indicates the remote SNMP entity to which the user is connected.

<snmpAuthenticationMethod> indicates the authentication method used.

<snmpAuthenticationKey> defines the authentication key if encryption is used for <snmpAuthenticationMethod>.

<snmpAuthenticationPassword> optional password used to calculate the <snmpAuthenticationKey> value if encryption is used for <snmpAuthenticationMethod>.

<snmpPrivacyMethod> indicates if messages are protected from disclosure, and if so, the type of privacy protocol used.

<snmpPrivacyKey> defines the privacy key if encryption is used for <snmpPrivacyMethod>.

<snmpPrivacyPassword> optional password used to calculate the <snmpPrivacyKey> value if encryptions is used for <snmpPrivacyMethod>

SNMPUser XML Block

```
<SNMPUser version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>    <!-- req, xs:string;id -->      </id>
  <userName>    <!-- req, xs:string -->   </userName>
  <remoteEngineID>   <!-- req, xs:hexBinary -->   </remoteEngineID>
  <snmpAuthenticationMethod>
    <!-- req, xs:string, "MD5,SHA,none" -->
  </snmpAuthenticationMethod>
  <snmpAuthenticationKey>     <!-- dep, xs:string -->   </snmpAuthenticationKey>
  <snmpAuthenticationPassword>
    <!-- dep, xs:string, see RFC3414 -->
  </snmpAuthenticationPassword>
  <snmpPrivacyMethod>      <!-- req, xs:string, "DES,AES,none" -->
    </snmpPrivacyMethod>
  <snmpPrivacyKey>  <!-- dep, xs:string -->   </snmpPrivacyKey>
  <snmpPrivacyPassword>      <!-- dep, xs:string, see RFC3414 -->
</snmpPrivacyPassword>
</SNMPUser>
```

8.2.36 /ISAPI/System/Network/mailing

/ISAPI/System/Network/mailing	**General Resource v2.0**
GET	
Description	It is used to get the configuration of e-mail.
Query	None

Inbound Data	None
Success Return	**mailingList**
PUT	
Description	It is used to set the configuration of e-mail.
Query	None
Inbound Data	**mailingList**
Success Return	**ResponseStatus**
Notes:	

mailingList XML Block

```
<mailingList xmlns="http://www.isapi.org/ver20/XMLSchema">
   <mailing> <!-- opt, xs:string> </mailing>
</mailingList>
```

8.2.37 /ISAPI/System/Network/mailing/<ID>

/ISAPI/System/Network/mailing/ID	**General Resource v2.0**
GET	
Description	It is used to get the configuration of a particular e-mail.
Query	None
Inbound Data	None
Success Return	**mailingList**
PUT	
Description	It is used to set the configuration of a particular e-mail.
Query	None
Inbound Data	**mailingList**
Success Return	**ResponseStatus**
Notes:	

mailing XML Block

```
<mailing xmlns="http://www.isapi.org/ver20/XMLSchema">
    <id/> <!-- req, xs:string, id -->
  <sender> <!--req-->
    <name> <!--req, xs:string> </name>
    <emailAddress> <!--req, xs:string --> </emailAddress>
    <smtp> <!-- req -->
      <enableAuthorization><!--req, xs:boolean--></enableAuthorization>
      <enableSSL><!--opt, xs:boolean--></enableSSL>
      <addressingFormatType>
        <!-- req, xs:string, "ipaddress,hostname" -->
      </addressingFormatType>
```

```
        <hostName>    <!-- dep, xs:string -->  </hostName>
        <ipAddress>    <!-- dep, xs:string -->  </ipAddress>
        <ipv6Address> <!-- dep, xs:string -->  </ipv6Address>
        <portNo>  <!-- opt, xs:integer --> </portNo>
        <accountName>     <!-- dep, xs:string -->  </accountName>
        <password>     <!-- dep, xs:string -->  </password>
      </smtp>
   </sender>
   <receiverList> <!-- req -->
      <receiver>
        <id> <!--req, xs:string; id --> </id>
        <name> <!--req, xs:string --> </name>
        <emailAddress> <!-- req, xs:string --> </emailAddress>
      </receiver>
   </receiverList>
   <attachment><!--opt-->
      <snapshot> <!--opt-->
        <enabled ><!--req, xs:boolean--></ enabled>
        <interval><!--req, xs:integer, seconds></interval>
      </snapshot>
   </attachment>
</mailing>
```

8.2.38 /ISAPI/System/Network/UPnP

/ISAPI/System/Network/UPnP	General Resource v2.0
GET	
Description	Get theUPnP configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**UPnP**
PUT	
Description	Set theUPnP configuration on an IP media device.
Query	None
Inbound Data	**UPnP**
Success Return	**ResponseStatus**
Notes:	

UPnP XML Block

```
<UPnP version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <enabled/>   <!-- req -->
```

```
    <ports/> <!-- opt -->
</UPnP>
```

8.2.39 /ISAPI/System/Network/UPnP/ports

/ISAPI/System/Network/UPnP/ports	General Resource v2.0
GET	
Description	Get the Ports configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**ports**
PUT	
Description	Set Ports configuration on an IP media device.
Query	None
Inbound Data	**ports**
Success Return	**ResponseStatus**
Notes:	

ports XML Block

```
<ports version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <enabled/> <!-- req -->
    <mapmode> <!-- req, xs:string, "auto,manual" --></mapmode>
    <natRouterLanAddr> <!-- opt -->
        <ipVersion>    <!-- req, xs:string, "v4,v6,dual" -->     </ipVersion>
        <ipAddress>       <!-- dep, xs:string -->       </ipAddress>
        <ipv6Address>      <!-- dep, xs:string -->       </ipv6Address>
     </natRouterLanAddr>
    <portList> <!-- req -->
       <port/>
    </portList>
    <natType> <!--req, xs:string, "manual, auto" --> </natType>
</ports>
```

8.2.40 /ISAPI/System/Network/UPnP/ports/status

/ISAPI/System/Network/UPnP/ports/status	General Resource v2.0
GET	
Description	Get NAT ports status on an IP media device.
Query	None

Inbound Data	None
Success Return	**portsStatus**
Notes: <natRouter> if this element is provided, the ip media device will use this nat router.	

portsStatus XML Block

```
<portsStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled/> <!-- req -->
   <natRouterLanAddr> <!-- req -->
     <ipVersion>  <!-- req, xs:string, "v4,v6,dual" -->  </ipVersion>
     <ipAddress>  <!-- dep, xs:string -->  </ipAddress>
     <ipv6Address>  <!-- dep, xs:string -->  </ipv6Address>
   </natRouterLanAddr>
   <natRouterWanAddr> <!-- req -->
     <ipVersion>  <!-- req, xs:string, "v4,v6,dual" -->  </ipVersion>
     <ipAddress>  <!-- dep, xs:string -->  </ipAddress>
     <ipv6Address>  <!-- dep, xs:string -->  </ipv6Address>
   </natRouterWanAddr>
  <portStatusList> <!-- req -->
    <portStatus/> <!-- req -->
  </portStatusList>
</portsStatus>
```

8.2.41 /ISAPI/System/Network/UPnP/ports/<ID>

/ISAPI/System/Network/UPnP/ports/<ID>			**General Resource v2.0**
GET			
Description	Get a specific NAT port configuration on an IP media device.		
Query	None		
Inbound Data	None		
Success Return	**port**		
PUT			
Description	Set a specific NAT port configuration on an IP media device.		
Query	None		
Inbound Data	None		
Success Return	**port**		
Error Status Code	**statusCode**	**subStausCode**	**description**
	6	**badPort**	**Port Conflict**
Notes:			

port XML Block

```
<port version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id/> <!-- req, xs:string, id -->
   <enabled/> <!—req, xs:boolean -->
   <internalPort/> <!-- req, xs:string, "http, admin, rtsp, …">
   <externalPort/> <!—req, xs:integer -->
</port>
```

8.2.42 /ISAPI/System/Network/UPnP/ports/<ID>/status

/ISAPI/System/Network/UPnP/ports/<ID>/status	General Resource v2.0
GET	
Description	Get NAT port status on an IP media device.
Query	None
Inbound Data	None
Success Return	**portStatus**
Notes: <natRouter> if this element is provided, the ip media device will use this nat router.	

portStatus XML Block

```
<portStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id/> <!-- req, xs:string, id -->
   <enabled/> <!-- req -->
   <internalPort/> <!-- req, xs:string, "http, admin, rtsp, …">
   <externalPort/> <!-- req, xs:integer -->
   <status/> <!-- req, xs:string, "inactive, active, conflict, …"-->
</portStatus>
```

8.2.43 /ISAPI/System/Network/ftp

/ISAPI/System/Network/ftp	General Resource v2.0
GET	
Description	It is used to get the configurations of FTP.
Query	None
Inbound Data	None
Success Return	**FTPNotificationList**
PUT	

Description	It is used to set the configurations of FTP.
Query	None
Inbound Data	**FTPNotificationList**
Success Return	**ResponseStatus**
Notes:	

FTPNotificationList XML Block

```
<FTPNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <FTPNotification/>  <!-- opt -->
</FTPNotificationList>
```

8.2.44 /ISAPI/System/Network/ftp/<ID>

/ISAPI/System/Network/ftp/ID	**General Resource v2.0**
GET	
Description	It is used to get the configuration of a particular FTP.
Query	None
Inbound Data	None
Success Return	**FTPNotification**
PUT	
Description	It is used to set the configurations of a particular FTP.
Query	None
Inbound Data	**FTPNotification**
Success Return	**ResponseStatus**
Notes: Depending on the value of <addressingFormatType>, either the <hostName> or the IP address fields will be used to locate the NTP server. Note: FTP transfers are always in binary mode. <pathDepth> the depth of path. For example, / depth is 0, /a depth is 1, /a/b depth is 2	

FTPNotification XML Block

```
<FTPNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <enabled> <!--req, xs:boolean --> </enabled>
  <useSSL> <!—opt, xs:boolean> </useSSL>
  <addressingFormatType>
    <!-- req, xs:string, "ipaddress,hostname" -->
  </addressingFormatType>
```

```
<hostName>                <!-- dep, xs:string -->              </hostName>
<ipAddress>               <!-- dep, xs:string -->              </ipAddress>
<ipv6Address>             <!-- dep, xs:string -->              </ipv6Address>
<portNo>                  <!-- opt, xs:integer -->             </portNo>
<userName>                <!-- req, xs:string -->              </userName>
<password>                <!-- wo, xs:string -->               </password>
<passiveModeEnabled>      <!-- opt, xs:boolean -->             </passiveModeEnabled>
<annoyftp> <!--opt, xs:boolean --> </annoyftp>
<uploadPicture> <!--opt, xs:boolean --> </uploadPicture>
<uploadVideoClip> <!-- opt, xs:Boolean --> </uploadVideoClip>
<uploadPath> <!—req -->
  <pathDepth> <!--req, xs:integer, 0…2 --> </pathDepth>
  <topDirNameRule>
    <!-- dep, xs:string, “devName, devId, devIp, customize” -->
  </topDirNameRule>
  <topDirName/> <!-- dep, xs:string-->
  <subDirNameRule>
    <!-- dep, xs:string, “chanName, chanId, customize”
  </subDirNameRule>
  <subDirName/> <!-- dep, xs:string-->
</uploadPath>
</FTPNotification>
```

8.2.45 /ISAPI/System/Network/ipFilter

/ISAPI/System/Network/ipFilter	**General Resource v2.0**
GET	
Description	Access IP filtering settings.
Query	None
Inbound Data	None
Success Return	**IPFilter**
PUT	
Description	Access IP filtering settings..
Query	None
Inbound Data	**IPFilter**
Success Return	**ResponseStatus**
Notes: <permissionType> field, if provided as a direct child of <IPFilter>, acts as a system level configuration and will apply to all of the <IPFilterAddress> entries, overriding the value provided in a particular <IPFilterAddress> block	

IPFilter XML Block

```
<IPFilter version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean --> </enabled>
  <permissionType> <!-- opt, xs:string, "deny,allow" --> </permissionType>
  <IPFilterAddressList/> <!-- opt -->
</IPFilter>
```

8.2.46 /ISAPI/System/Network/ipFilter/filterAddresses

/ISAPI/System/Network/ipFilter/filterAddresses	General Resource v2.0
GET	
Description	Access IP filtering settings.
Query	None
Inbound Data	None
Success Return	IPFilterAddressList
PUT	
Description	Access IP filtering settings..
Query	None
Inbound Data	IPFilterAddressList
Success Return	**ResponseStatus**
POST	
Description	Access IP filtering settings..
Query	None
Inbound Data	IPFilterAddressList
Success Return	**ResponseStatus**
DELETE	
Description	Access IP filtering settings..
Query	None
Inbound Data	IPFilterAddressList
Success Return	**ResponseStatus**
Notes: The IP filter address list allows addresses to be added and removed from the list, or the entire list to be uploaded at once.	

IPFilterAddressList XML Block

```
<IPFilterAddressList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IPFilterAddress/>    <!-- opt -->
</IPFilterAddressList>
```

8.2.47 /ISAPI/System/Network/ipFilter/filterAddresses/<ID>

/ISAPI/System/Network/ipFilter/filterAddresses/ID	General Resource v2.0
GET	
Description	Access IP filtering settings.
Query	None
Inbound Data	None
Success Return	IPFilterAddress
PUT	
Description	Access IP filtering settings..
Query	None
Inbound Data	IPFilterAddress
Success Return	**ResponseStatus**
DELETE	
Description	Access IP filtering settings..
Query	None
Inbound Data	IPFilterAddress
Success Return	**ResponseStatus**

Notes:

If the <permissionType> tag is not provided as a direct child of <IPFilter>, the <permissionType> tag must be provided for each <IPFilterAddress>.

Since the ordering of the filters can change the behavior, filtering will be applied consecutively starting with the first <IPFilterAddress> in the list.

The <bitMask> field is applied to the corresponding IP address to identify a range of addresses. It indicates the number of '1' bits used to mask the address. For example: '24' would correspond to a subnet mask of 255.255.255.0 and '32' would correspond to a subnet mask of 255.255.255.255 (a single IP address) for IPv4.

If <addressFilterType> refers to "mask", the <AddressMask> block must be provided in place of the <AddressRange> block. If it refers to "range", the <Range> block must be provided in place of the <AddressMask> block.

Use of IPv4 or IPv6 addresses depends on the value of the <ipVersion> field in /ISAPI/System/Network/interfaces/ID/ipAddress.

IPFilterAddress XML Block

```
<IPFilterAddress version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>    <!-- req, xs:string;id -->    </id>
  <permissionType><!-- dep, xs:string, "deny,allow" --></permissionType>
```

```
  <addressFilterType><!-- req, xs:string, “mask,range” -->  </addressFilterType>
    <AddressRange>  <!-- dep, depends on <addressFilterType> -->
      <startIPAddress>      <!-- dep, xs:string -->  </startIPAddress>
      <endIPAddress> <!-- dep, xs:string -->  </endIPAddress>
      <startIPv6Address>  <!-- dep, xs:string -->  </startIPv6Address>
      <endIPv6Address>    <!-- dep, xs:string -->  </endIPv6Address>
    </AddressRange>
    <AddressMask>    <!-- dep, depends on <addressFilterType> -->
      <ipAddress><!-- dep, xs:string -->  </ipAddress>
      <ipv6Address>   <!-- dep, xs:string -->  </ipv6Address>
      <bitMask>   <!-- req, xs:string -->   </bitMask>
    </AddressMask>
</IPFilterAddress>
```

8.2.48 /ISAPI/System/Network/qos

/ISAPI/System/Network/qos	General Resource v2.0
GET	
Description	This function is used to get QoS Settings.
Query	None
Inbound Data	None
Success Return	**QoS**
PUT	
Description	This function is used to set QoS Settings
Query	None
Inbound Data	**QoS**
Success Return	**ResponseStatus**
Notes: At least one of <CoSList> or <DSCPList> must be provided.	

QoS XML Block

```
<QoS version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <CoSList/> <!-- dep -->
  <DSCPList/>     <!-- dep -->
</QoS>
```

8.2.49 /ISAPI/System/Network/qos/cos

/ISAPI/System/Network/qos/cos	General Resource v2.0
GET	

Description	This function is used to get the QoS cos list setting for the device.
Query	None
Inbound Data	None
Success Return	**CoSList**
PUT	
Description	This function is used to set the QoS cos list setting for the device
Query	None
Inbound Data	**CoSList**
Success Return	**ResponseStatus**
POST	
Description	This function is used to creat the QoS cos setting for the device
Query	None
Inbound Data	**CoS**
Success Return	**ResponseStatus**
DELETE	
Description	This function is used to delete the QoS cos list setting for the device
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

CoSList XML Block

```
<CoSList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <CoS/>      <!-- opt -->
</CoSList>
```

8.2.50 /ISAPI/System/Network/qos/cos/<ID>

/ISAPI/System/Network/qos/cos/ID	**General Resource v2.0**
GET	
Description	This function is used to get the QoS cos setting for the device
Query	None
Inbound Data	None
Success Return	**CoS**
PUT	
Description	This function is used to set the QoS cos setting for the device
Query	None
Inbound Data	**CoS**
Success Return	**ResponseStatus**

DELETE	
Description	This function is used to delete the QoS cos setting for the device
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes:	

CoS XML Block

```
<CoS version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>    <!-- req, xs:string;id -->      </id>
  <enabled>  <!-- req, xs:boolean -->      </enabled>
  <priority>   <!-- req, xs:integer --> </priority>
  <vlanID>    <!-- req, xs:string -->   </vlanID>
  <trafficType>
    <!-- req, xs:string, "devicemanagement,commandcontrol,video,audio" -->
  </trafficType>
</CoS>
```

8.2.51 /ISAPI/System/Network/qos/dscp

/ISAPI/System/Network/qos/dscp	General Resource v2.0
GET	
Description	This function is used to get the QoS dscp list setting for the device
Query	None
Inbound Data	None
Success Return	**DSCPList**
PUT	
Description	This function is used to set the QoS dscp list setting for the device
Query	None
Inbound Data	**DSCPList**
Success Return	**ResponseStatus**
POST	
Description	This function is used to create the QoS dscp setting for the device
Query	None
Inbound Data	**DSCP**
Success Return	**ResponseStatus**
DELETE	
Description	This function is used to delete the QoS cos list setting for the device
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**

Notes:
A list of DSCP parameter blocks is specified for each type of traffic: device management, command and control, video and audio streaming. Devices may extend the set of traffic types.

DSCPList XML Block

```
<DSCPList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DSCP/>  <!-- opt -->
</DSCPList>
```

8.2.52 /ISAPI/System/Network/qos/dscp/<ID>

/ISAPI/System/Network/qos/dscp/ID	General Resource v2.0
GET	
Description	This function is used to get the QoS dscp setting for the device
Query	None
Inbound Data	None
Success Return	**DSCP**
PUT	
Description	This function is used to set the QoS dscp setting for the device
Query	None
Inbound Data	**DSCP**
Success Return	**ResponseStatus**
DELETE	
Description	This function is used to delete the QoS dscp setting for the device
Query	None
Inbound Data	**None**
Success Return	**ResponseStatus**
Notes: <trafficType> determines which kind of traffic the settings apply to.	

DSCP XML Block

```
<DSCP version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <enabled>  <!-- req, xs:boolean -->  </enabled>
  <priorityValue>  <!-- req, xs:integer, 6 bits - refer to RFC2474 --> </priorityValue>
  <trafficType>
    <!-- req, xs:string, "devicemanagement,commandcontrol,video,audio" -->
```

```
  </trafficType>
</DSCP>
```

8.2.53 /ISAPI/System/Network/telnetd

/ISAPI/System/Network/telnetd	General Resource v2.0
GET	
Description	It is used to get the configurations of telnet.
Query	None
Inbound Data	None
Success Return	**Telnetd**
PUT	
Description	It is used to set the configurations of telnet.
Query	None
Inbound Data	**Telnetd**
Success Return	**ResponseStatus**
Notes:	

Telnetd XML Block

```
<Telnetd version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <enabled>   <!-- req, xs:boolean -->   </enabled>
</Telnetd>
```

8.2.54 /ISAPI/System/Network/SIP

/ISAPI/System/Network/SIP	General Resource v2.0
GET	
Description	
Query	None
Inbound Data	None
Success Return	**SIPServerList**
PUT	
Description	
Query	None
Inbound Data	**SIPServerList**
Success Return	**ResponseStatus**
Notes:	

SIPServerList XML Block

```
<SIPServerList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <SIPServer/> <!—- opt -->
</SIPServerList>
```

8.2.55 /ISAPI/System/Network/SIP/<ID>

/ISAPI/System/Network/SIP/<ID>	General Resource v2.0
GET	
Description	
Query	None
Inbound Data	None
Success Return	**SIPServer**
PUT	
Description	
Query	None
Inbound Data	**SIPServer**
Success Return	**ResponseStatus**
Notes:	

SIPServer XML Block

```
<SIPServer version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!—req, xs:string --> </id>
  <localPort>  <!-- req, xs:integer, "1-65535"-->  </localPort>
  <streamID>  <!-- req, xs:interger, "1(main stream),2（sub stream）"-->  </streamID>
  <Standard>  <!-- opt -->
    <registerStatus>  <!-- ro, req, xs:boolern, "false（unregistered）,true（registered）"
                                                                    --></registerStatus>
    <enabled>  <!-- req, xs:string, "true(sign in),false（log out）"-->  </enabled>
    <registrar>  <!-- req, xs:string-->  </registrar>
    <registrarPort>  <!-- req, xs:integer-->  </registrarPort>
    <proxy>  <!-- req, xs:string-->  </proxy>
    <proxyPort>  <!-- req, xs:integer-->  </proxyPort>
    <displayName>  <!-- req, xs:string-->  </displayName>
    <userName>  <!-- req, xs:string-->  </userName>
    <authID>  <!-- req, xs:string-->  </authID>
    <password>  <!-- wo, req, xs:string-->  </password>
    <expires>  <!-- req, xs:integer-->  </expires>
  </Standard>
  <GB28181>  <!-- opt -->
    <registerStatus>  <!-- req, xs:boolean  --></registerStatus>
    <enabled>  <!-- req, xs:string, "true,false"-->  </enabled>
    <registrar>  <!-- req, xs:string-->  </registrar>
    <registrarPort>  <!-- req, xs:integer-->  </registrarPort>
    <serverId>  <!-- req, xs:string-->  </serverId>
    <serverDomain>  <!-- req, xs:integer-->  </serverDomain>
    <userName>  <!-- req, xs:string-->  </userName>
    <authID>  <!-- req, xs:string-->  </authID>
    <password>  <!-- wo, req, xs:string-->  </password>
    <expires>  <!-- req, xs:integer-->  </expires>
    <liveTime>  <!-- req, xs:integer-->  </liveTime>
    <heartbeatTime>  <!-- req, xs:integer-->  </heartbeatTime>
    <heartbeatCount>  <!-- req, xs:integer-->  </heartbeatCount>
  </GB28181>
</SIPServer>
```

8.2.56 /ISAPI/System/Network/SIP/<ID>/SIPInfo

/ISAPI/System/Network/SIP/<ID>/SIPInfo	General Resource v2.0
GET	
Description	Get device ID and alarm ID
Query	None
Inbound Data	None

Success Return	**SIPInfo**
PUT	
Description	Set device ID and alarm ID
Query	None
Inbound Data	**SIPInfo**
Success Return	**ResponseStatus**
Notes:	

SIPInfo XML Block

```
<SIPInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <videoID> <!-- req, xs:string--> <videoID>
  <AlarmInList>
  <AlarmIn>
    <id>  <!—req, xs:string-->  </id>
    <alarmInID> <!-- req, xs:string--> <alarmInID>
  <AlarmIn>
  <AlarmInList>
</SIPInfo>
```

8.3 /ISAPI/System/IO

/ISAPI/System/IO	**Service v2.0**
GET	
Description	It is used to get the I/O ports information.
Query	None
Inbound Data	None
Success Return	**IOPortList**
Notes: The allocation of IDs between input and output ports must be unique.	

IOPortList XML Block

```
<IOPortList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IOInputPortList/>      <!-- opt -->
  <IOOutputPortList/>   <!-- opt -->
</IOPortList>
```

8.3.1/ISAPI/System/IO/capabilities

/ISAPI/System/IO/capabilities	General Resource v2.0
GET	
Description	It is used to get device capability.
Query	None
Inbound Data	None
Success Return	**<IOCap>**
Notes:	

IOCap XML Block

```
<IOCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IOInputPortNums>   <!-- opt, xs:integer--> <IOInputPortNums>
  <IOOutputPortNums> <!-- opt, xs:integer--> <IOOutputPortNums>
</IOCap>
```

8.3.2/ISAPI/System/IO/status

/ISAPI/System/IO/status	General Resource v2.0
GET	
Description	It is used to get the status of the I/O ports.
Query	None
Inbound Data	None
Success Return	**IOPortStatusList**
Notes: <ioPortID> refers to /IO/inputs/ID or /IO/outputs/ID. The port IDs are guaranteed to be unique across input and output ports. <ioState> indicates whether the input port is active or inactive. In most applications, a high signal is considered active.	

IOPortStatusList XML Block

```
<IOPortStatusList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IOPortStatus>     <!-- req -->
    <ioPortID>     <!-- req, xs:integer, "1, 2" -->                    </ioPortID>
    <ioPortType>   <!-- req, xs:string, "input,output" -->          </ioPortType>
    <ioState>      <!-- req, xs:string, "active,inactive" -->       </ioState>
  </IOPortStatus>
</IOPortStatusList>
```

8.3.3/ISAPI/system/IO/inputs

/ISAPI/System/IO/inputs	General Resource v2.0
GET	
Description	It is used to get the Input ports information.
Query	None
Inbound Data	None
Success Return	**IOInputPortList**
Notes: IO inputs are hardwired, meaning that the inputs are statically allocated by the device and cannot be created or deleted.	

IOInputPortList XML Block

```
<IOInputPortList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IOInputPort/>    <!-- opt -->
</IOInputPort>
```

8.3.4/ISAPI/System/IO/inputs/<ID>

/ISAPI/System/IO/inputs/*ID*	General Resource v2.0
GET	
Description	It is used to get particular input port information.
Query	None
Inbound Data	None
Success Return	**IOInputPort**
PUT	
Description	It is used to update particular input port information.
Query	None
Inbound Data	**IOInputPort**
Success Return	**ResponseStatus**
Notes: <triggering> indicates the signal conditions to trigger the input port. High/Low will continuously trigger for the duration of high/low input signal. <name> IO input port name.	

IOInputPort XML Block

```
<IOInputPort version="2.0"   xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>        <!-- req, xs:integer -->                    </id>
  <enabled> <!--req,Boolean,"true,false"--> </enabled>
  <triggering>   <!-- req, xs:string, "high,low" -->   <triggering>
```

```
  <name> <!--opt,xs:string--></name>
</IOInputPort>
```

8.3.5/ISAPI/System/IO/inputs/<ID>/status

/ISAPI/System/IO/inputs/*ID*/status	General Resource v2.0
GET	
Description	It is used to get the status of a particular input port.
Query	None
Inbound Data	None
Success Return	**IOPortStatus**
Notes: See /IO/status for an explanation of the fields.	

8.3.6/ISAPI/System/IO/outputs

/ISAPI/System/IO/outputs	General Resource v2.0
GET	
Description	It is used to get the output ports information.
Query	None
Inbound Data	None
Success Return	**IOOutputPortList**
Notes: IO outputs are hardwired, meaning that the outputs are statically allocated by the device and cannot be created or deleted.	

IOOutputPortList XML Block

```
<IOOutputPortList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IOOutputPort/>   <!-- opt -->
</IOOutputPort>
```

8.3.7/ISAPI/System/IO/outputs/<ID>

/ISAPI/System/IO/outputs/*ID*	General Resource v2.0
GET	
Description	It is used to get particular output port information.
Query	None

Inbound Data	None
Success Return	**IOOutputPort**
PUT	
Description	It is used to update particular output port information.
Query	None
Inbound Data	**IOOutputPort**
Success Return	**ResponseStatus**
Notes: <PowerOnState> defines the output port configuration when the device is powered on. <defaultState> is the default output port signal when it is not being triggered. <outputState> is the output port signal when it is being triggered. Pulse will cause the output port to send a signal (opposite of the <defaultState>) for a duration specified by the <pulseDuration> tag. <pulseDuration> is the duration of a output port signal when it is being triggered. It must be provided if the <outputState> is “pulse”.	

IOOutputPort XML Block

```
<IOOutputPort version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>              <!-- req, xs:integer, "2" -->              </id>
  <PowerOnState>       <!-- req -->
    <defaultState>    <!—ro, req, xs:string, “high,low” -->    </defaultState>
    <outputState>     <!—ro, req, xs:string, “high,low,pulse” -->   </outputState>
    <pulseDuration>   <!-- dep, xs:integer, milliseconds --> </pulseDuration> //延时
  </PowerOnState>
  <name> <!--opt, xs:string--> </name>
</IOOutputPort>
```

8.3.8/ISAPI/System/IO/outputs/<ID>/status

/ISAPI/System/IO/outputs/*ID*/status	**General Resource v2.0**
GET	
Description	It is used to get the status of a particular output port.
Query	None
Inbound Data	None
Success Return	**IOPortStatus**
Notes: See /IO/status for an explanation of the fields.	

8.3.9/ISAPI/System/IO/outputs/<ID>/trigger

/ISAPI/System/IO/outputs/*ID*/trigger	General Resource v2.0
PUT	
Description	It is used to manually trigger a particular output port.
Query	None
Inbound Data	**IOPortData**
Success Return	**ResponseStatus**
Notes: Note that the ID used here MUST correspond to the ID in /IO/outputs/ID. The IO output port is toggled to a high or low signal accordingly.	

IOPortData XML Block

```
<IOPortData xmlns="http://www.isapi.org/ver20/XMLSchema">
   <outputState>      <!-- req, xs:string, "high,low" -->   </outputState>
</IOPortData>
```

8.4 /ISAPI/System/Video

/ISAPI/System/Video	Service v2.0
Notes:	

8.4.1/ISAPI/System/Video/capabilities

/ISAPI/System/Video/capabilities	General Resource v2.0
GET	
Description	It is used to get device capability.
Query	None
Inbound Data	None
Success Return	**<VideoCap>**
Notes:	

VideoCap XML Block

```
<VideoCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <videoInputPortNums> <!-- opt, xs:integer--> <videoInputPortNums>
   <videoOutputPortNums> <!-- opt, xs:integer--> <videoOutputPortNums>
</VideoCap>
```

8.4.2/ISAPI/System/Video/inputs

/ISAPI/System/Video/inputs	General Resource v2.0
GET	
Description	It is used to get the video inputs configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoInput**
Notes: An IP media device may contain a set of video inputs. These inputs are hardwired by the device, meaning that the IDs can be discovered but not created or deleted.	

VideoInput XML Block

```
<VideoInput version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <VideoInputChannelList/>  <!-- opt -->
</VideoInput>
```

8.4.3/ISAPI/System/Video/inputs/channels

ISAPI/System/Video/inputs/channels	General Resource v2.0
GET	
Description	It is used to get the video input channels configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoInputChannelList**
Notes: Since video input channels are resources that are defined by the hardware configuration of the device, they cannot be created or deleted.	

VideoInputChannelList XML Block

```
<VideoInputChannelList version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <VideoInputChannel/>      <!-- opt -->
</VideoInputChannelList>
```

8.4.4/ISAPI/System/Video/inputs/channels/<ID>

ISAPI/System/Video/inputs/channels/*ID*	General Resource v2.0
GET	
Description	It is used to get a particular video input channel configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoInputChannel**
PUT	
Description	It is used to update a particular video input channel configuration on an IP media device.
Query	None
Inbound Data	**VideoInputChannel**
Success Return	**ResponseStatus**
Notes:	

VideoInputChannel XML Block

```
<VideoInputChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string --> </id>
  <inputPort>  <!-- req, xs:string -->  </inputPort>
  <videoInputEnabled> <!-- opt, xs:boolean --> </videoInputEnabled>
  <name> <!-- opt, xs:string --> </name>
  <videoFormat> <!-- opt, xs:string, "PAL, NTSC" --> </videoFormat>
  <portType> <!--opt, xs:string, "SDI, OPT, VGA, HDMI, YPbPr" --> </portType>
  <resDesc> <!--opt, xs:string--> </resDesc>
</VideoInputChannel>
```

8.4.5/ISAPI/System/Video/inputs/channels/<ID>/focus

/ISAPI/System/Video/inputs/channels/*ID*/focus	General Resource v2.0
PUT	
Description	Manually focus a video input channel.
Query	None
Inbound Data	**FocusData**
Success Return	**ResponseStatus**
Notes:	

<focus>: focus vector data. Negative numbers focus near, positive numbers focus far. Numerical value is a percentage of the maximum focus speed of the lens module.

FocusData XML Block

```
<FocusData version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <focus>       <!-- req, xs:intger -->   </focus>
</FocusData>
```

8.4.6/ISAPI/System/Video/inputs/channels/<ID>/iris

/ISAPI/System/Video/inputs/channels/*ID*/iris	General Resource v2.0
PUT	
Description	Manually adjust iris for a video input channel.
Query	None
Inbound Data	**IrisData**
Success Return	**ResponseStatus**
Notes: <iris> negative numbers close iris, positive numbers open iris. Numerical value is a percentage of the maximum iris speed of the lens module.	

IrisData XML Block

```
<IrisData version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <iris>    <!-- req, xs:intger -->   </iris>
</IrisData>
```

8.4.7/ISAPI/System/Video/inputs/channels/<ID>/privacyMask

/ISAPI/System/Video/inputs/channels/*ID*/privacyMask	General Resource v2.0
GET	
Description	It is used to get the privacy masking configuration for a video input channel.
Query	None

Inbound Data	None
Success Return	**PrivacyMask**
PUT	
Description	It is used to update the privacy masking configuration for a video input channel.
Query	None
Inbound Data	**PrivacyMask**
Success Return	**ResponseStatus**
Notes: Privacy masking can be enabled and the region list configured per channel.	

PrivacyMask XML Block

```
<PrivacyMask version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled>  <!-- req, xs:boolean -->  </enabled>
  <normalizedScreenSize>  <!--opt-->
    <normalizedScreenWidth> <!-- req, xs:integer --></normalizedScreenWidth>
    <normalizedScreenHeight> <!-- req, xs:integer --></normalizedScreenHeight>
  <normalizedScreenSize>
  <PrivacyMaskRegionList/>  <!-- opt -->
  <regionType> <!-- opt, xs:string,"quadrilateral" --> </regionType>
</PrivacyMask>
```

8.4.8/ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions

/ISAPI/System/Video/inputs/channels/*ID*/privacyMask/regions	**General Resource v2.0**
GET	
Description	It is used to get the privacy mask regions configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**PrivacyMaskRegionList**
PUT	
Description	It is used to update the privacy mask regions configuration for a video input channel.
Query	None
Inbound Data	**PrivacyMaskRegionList**
Success Return	**ResponseStatus**
POST	

Description	It is used to add a privacy mask region for a video input channel.
Query	None
Inbound Data	**PrivacyMaskRegion**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the privacy mask regions configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Privacy masking consists of a set of regions that are combined to grey or black out areas of a video input.	

PrivacyMaskRegionList XML Block

```
<PrivacyMaskRegionList version="2.0"
   xmlns="http://www.isapi.org/ver20/XMLSchema">
   <PrivacyMaskRegion/>    <!-- opt -->
</PrivacyMaskRegionList>
```

8.4.9/ISAPI/System/Video/inputs/channels/<ID>/privacyMask/regions/<ID>

/ISAPI/System/Video/inputs/channels/*ID*/privacyMask/regions/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular privacy mask region configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**PrivacyMaskRegion**
PUT	
Description	It is used to update a particular privacy mask region configuration for a video input channel.
Query	None
Inbound Data	**PrivacyMaskRegion**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular privacy mask region

	configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

Region coordinates are dependent on normalized screen size.

The computer screen coordinate system is used, which the origin coordinate is on top-left corner, the Y axis is vertical downwards, the X axis is horizontal rightwards.

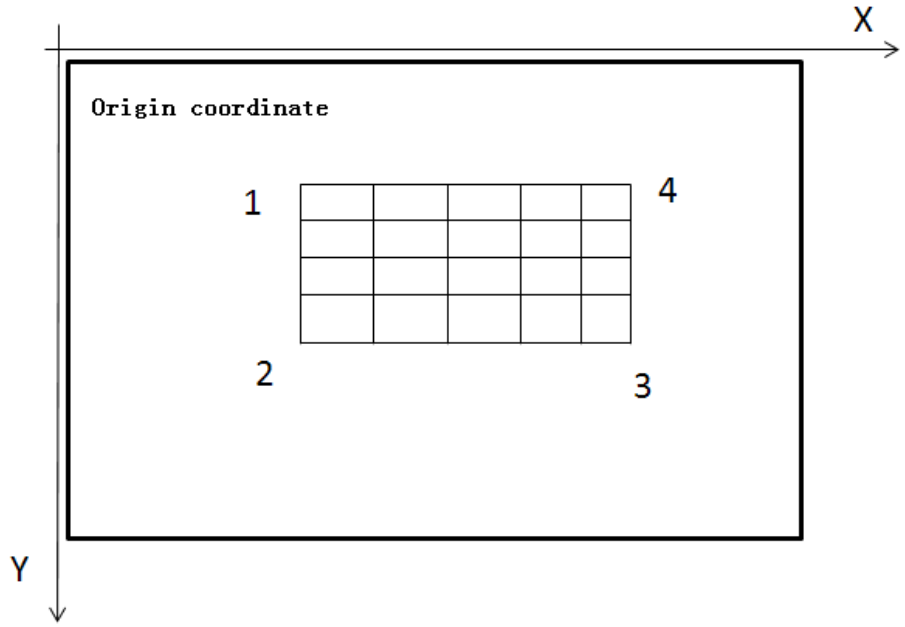
Only support the rectangular region which will be “drawn” from four coordinates. The four points is counterclockwise direction, and the beginning point is the top-left point.

Ordering of <PrivacyMaskRegion> blocks is insignificant.

PrivacyMaskRegion XML Block

```
<PrivacyMaskRegion version=“2.0”
 xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <id> <!-- req, xs:integer --> </id>
  <enabled> <!-- req,xs:boolean --> </enabled>
  <RegionCoordinatesList>   <!-- req -->
    <RegionCoordinates>   <!-- req -->
      <positionX>         <!-- req, xs:integer;coordinate -->       </positionX>
      <positionY>         <!-- req, xs:integer;coordinate   -->     </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
  <privacymaskName><!-- opt, xs:string--></privacymaskName>
  <maskType>
    <!--opt, xs:string "gray,red,yellow,blue,orange,green,
        transparent,half-transparent,mosaic"-->
  </maskType>
  <zoomdoorlimit> <!-- opt, xs:integer “10-1000”--> </zoomdoorlimit>
</PrivacyMaskRegion>
```

Example for priavacyMask Region:



8.4.10 /ISAPI/System/Video/inputs/channels/<ID>/tamperDetection

/ISAPI/System/Video/inputs/channels/*ID*/tamperDetection	General Resource v2.0
GET	
Description	It is used to get the shelter alarm configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**TamperDetection**
PUT	
Description	It is used to update the shelter alarm configuration for a video input channel.
Query	None
Inbound Data	**TmaperDetection**
Success Return	**ResponseStatus**
Notes:	

TameprDectection XML Block

```
<TamperDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean -->  </enabled>
  <normalizedScreenSize>  <!--req-->
```

```
    <normalizedScreenWidth> <!-- req, xs:integer --></normalizedScreenWidth>
    <normalizedScreenHeight> <!-- req, xs:integer --></normalizedScreenHeight>
  <normalizedScreenSize>
    < tampersensitivityLevel>
    <!--req, xs:integer, 0..100, 0 is the least sensitive -->
    </ tampersensitivityLevel>
  <TamperDetectionRegionList/>
    <!-- req -->
</ TamperDetection >
```

8.4.11 /ISAPI/System/Video/inputs/channels/<ID>/tamperDetection/regions

/ISAPI/System/Video/inputs/channels/*ID*/tamperDetection/regions	**General Resource v2.0**
GET	
Description	It is used to get the shelter alarm regions configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**TamperDetectionRegionList**
PUT	
Description	It is used to update the shelter alarm regions configuration for a video input channel.
Query	None
Inbound Data	**TamperDetectionRegionList**
Success Return	**ResponseStatus**
POST	
Description	It is used to add a shelter alarm region for a video input channel.
Query	None
Inbound Data	**TamperDetectionRegion**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the shelter alarm regions configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

TamperDetectionRegionList XML Block

```
<TamperDetectionRegionList version="2.0"
  xmlns="http://www.isapi.org/ver20/XMLSchema">
    <TamperDetectionRegion/> <!-- opt -->
</ TamperDetectionRegionList >
```

8.4.12 /ISAPI/System/Video/inputs/channels/<ID>/tamperDetection/regions/<ID>

/ISAPI/System/Video/inputs/channels/*ID*/tamperDetection/regions/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular shelter alarm region configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**TamperDetectionRegion**
PUT	
Description	It is used to update a particular shelter alarm region configuration for a video input channel.
Query	None
Inbound Data	**TamperDetectionRegion**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular shelter alarm region configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Region coordinates are dependent on video resolution. Only support the rectangular region which will be "drawn" from four coordinates. The four points is clockwise direction, and the beginning point is the low-left point. Ordering of <TamperDetectionRegion> blocks is insignificant.	

TamperDetectionRegion XML Block

```
<TamperDetectionRegion version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>                <!-- req, xs:string, id -->              </id>
  <sensitivityLevel>
    <!--req, xs:integer, 0..100, 0 is the least sensitive -->
  </sensitivityLevel>
  <RegionCoordinatesList>   <!-- req -->
    <RegionCoordinates>   <!-- req -->
      <positionX>         <!-- req, xs:integer;coordinate -->       </positionX>
      <positionY>         <!-- req, xs:integer;coordinate   -->     </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</TamperDetectionRegion>
```

8.4.13 /ISAPI/System/Video/inputs/channels/<ID>/motionDetection

/ISAPI/System/Video/inputs/channels/*ID*/MotionDetection	**Service v2.0**
GET	
Description	It is used to get the motion detection configuration for all video input channels.
Query	None
Inbound Data	None
Success Return	**MotionDetection**
PUT	
Description	It is used to udpate the motion detection configuration for a video input channel.
Query	None
Inbound Data	**MotionDetection**
Success Return	**ResponseStatus**
Notes: If motion detection is supported by the device, a motion detection ID will be allocated for each video input channel ID. The motion detection ID must correspond to the video input channel ID.	

MotionDetection XML Block

```
<MotionDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled>          <!-- req, xs:boolean -->              </enabled>
  <enableHighlight>          <!-- opt, xs:boolean -->              </enableHighlight>
```

```
  <samplingInterval> <!-- opt, xs:integer, number of frames --> </samplingInterval>
  <startTriggerTime> <!-- opt, xs:integer, milliseconds --> </startTriggerTime>
  <endTriggerTime> <!-- opt, xs:integer, milliseconds --> </endTriggerTime>
  <regionType>        <!-- ro, req, xs:string, “grid, roi, none” -->          </regionType>
  <Grid>              <!-- dep -->
    <rowGranularity>       <!-- ro, req, xs:integer -->    </rowGranularity>
    <columnGranularity>    <!-- ro, req, xs:integer -->    </columnGranularity>
  </Grid>
  <ROI>    <!-- dep -->
    <normalizedScreenWidth> <!-- ro, req, xs:integer --></normalizedScreenWidth>
    <normalizedScreenHeight> <!-- ro, req, xs:integer --></normalizedScreenHeight>
  </ROI>
  <MotionDetectionLayout/>    <!-- req -->
</MotionDetection>
```

8.4.14 /ISAPI/System/Video/inputs/channels/<ID>/motionDetection/layout

/ISAPI/System/Video/inputs/channels/*ID*/MotionDetection/layout	**General Resource**	**v2.0**
Notes:		

MotionDetectionLayout XML Block

```
<MotionDetectionLayout version=“2.0”
 xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <sensitivityLevel>       <!-- req -->
    <!-- req, xs:integer, “0-100”, 0 is least sensitive -->
  </sensitivityLevel>
  <layout>
    <gridMap> <!--dep, xs:hexstring--> </gridMap>
    <roiMap/>
</layout>
</MotionDetectionLayout>
```

8.4.15 /ISAPI/System/Video/inputs/channels/<ID>/motionDetection/layout/gridLayout

/ISAPI/System/Video/inputs/channels/*ID*/MotionDetection/layout/gridLayout	General Resource v2.0
GET	
Description	It is used to get the motion detection regions configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**MotionDetectionGridLayout**
PUT	
Description	It is used to update the motion detection regions configuration for a video input channel.
Query	None
Inbound Data	**MotionDetectionGridLayout**
Success Return	**ResponseStatus**
Notes: All motion detection regions share a sensitivity level. It is possible to define mask regions that are subtracted from other regions. <gridMap> required when region type is grid. A "1" denotes an grid to detect and a "0" no to detect. The first cell is in the upper left corner. Then the cell order goes first from left to right and then from up to down (see flowing example). If the number of cells is not a multiple of 8 the last byte is filled with zeros.	

MotionDetectionGridLayout XML Block

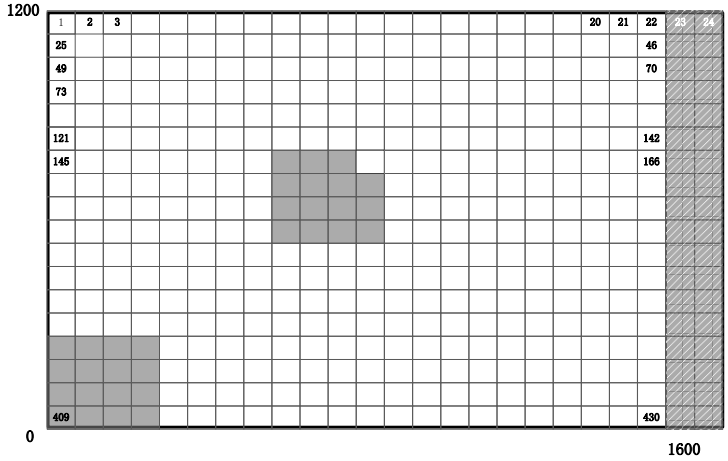
```
<MotionDetectionGridLayout version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <sensitivityLevel>      <!-- req -->
     <!-- req, xs:integer, "0-100", 0 is least sensitive -->
  </sensitivityLevel>
  <gridMap> <!--dep, xs:hexstring--> </gridMap>
</MotionDetectionGridLayout>
```

8.4.16 Motion Detection Example

Set up Motion Detection

The following command configures two rectangular detection regions, with one “masked” region on video input channel ID 1. Example assumes a resolution of 1600x1200 and a grid motion detection algorithm:

- Motion detection is enabled with a granularity of a 22x18 grid (each row will reserve 2 grids, the actual region is 24x18; but generally the last two rows are ignored.) – this means the detection region coordinates will ultimately be defined by a grid of 432 regions. For a resolution of 1600x1200, this means that each “granule” will be 1600/22 x 1200/18 pixels. (If a coordinate doesn’t exactly match the configured granularity, it should be mapped internally to the nearest possible point).




```
PUT /MotionDetection/1 HTTP/1.1
Content-Type: application/xml; charset=“UTF-8”
Content-Length: ISAPI

<?xml version=“2.0” encoding=“UTF-8”?>
    <MotionDetection version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <id>1</id>
  <enabled>true</enabled>
  <MotionDetectionLayout>
    <sensitivityLevel>20</sensitivityLevel>
      <gridMap>
    000000000000000000000000000000000e00000f00000f00000f000000000000000000000000000f00000f00000f00000f00000
    </gridMap>
  </MotionDetectionLayout>
</MotionDetection>
```

8.4.17 /ISAPI/System/Video/inputs/channels/<ID>/motionDetectionExt

/ISAPI/System/Video/inputs/channels/*ID*/motionDetectionExt	Service v2.0
GET	
Description	It is used to get the motion detection configuration for all video input channels.
Query	None
Inbound Data	None
Success Return	**MotionDetectionExt**
PUT	
Description	It is used to udpate the motion detection configuration for a video input channel.
Query	None
Inbound Data	**MotionDetectionExt**
Success Return	**ResponseStatus**
Notes: If motion detection is supported by the device, a motion detection ID will be allocated for each video input channel ID. The motion detection ID must correspond to the video input channel ID. The devices supports two kinds of motion detection, <activeMode> is used to check current motion detection mode, if the value is normal, please refer to /motionDetection branch; if the value is expert, please refer to /motionDetectionExt branch.	

MotionDetectionExt XML Block

```
<MotionDetectionExt version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean -->  </enabled>

  <minObjectSize>
    <!-- opt, xs:integer, min number of pixels per object -->
  </minObjectSize>
  <maxObjectSize>
    <!-- opt, xs:integer, max number of pixels per object -->
  </maxObjectSize>
  <ROI>
    <minHorizontalResolution> <!-- req, xs:integer -->  </minHorizontalResolution>
    <minVerticalResolution>  <!-- req, xs:integer -->  </minVerticalResolution>
  </ROI>
  <enableHighlight>  <!-- opt, xs:boolean -->  </enableHighlight>
```

```
  <MotionDetectionSwitch/>   <!—opt -->
  <activeMode> <!-- ro, xs:string, “normal,expert”> <activeMode>
  <MotionDetectionRegionList/> <!-- req -->
</MotionDetectionExt>
```

8.4.18 /ISAPI/System/Video/inputs/channels/<ID>/motionDetectionExt/regions

/ISAPI/System/Video/inputs/channels/*ID*/motionDetectionExt/regions	**General Resource**	**v2.0**
GET		

MotionDetectionRegionList XML Block

```
<MotionDetectionRegionList version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <MotionDetectionRegion/>   <!-- opt -->
</MotionDetectionRegionList>
```

8.4.19 /ISAPI/System/Video/inputs/channels/<ID>/motionDetectionExt/regions/<ID>

/ISAPI/System/Video/inputs/channels/*ID*/motionDetectionExt/ID/regions/ID	**General Resource**	**v2.0**

GET	
Description	It is used to get the motion detection configuration for all video input channels.
Query	None
Inbound Data	None
Success Return	**MotionDetectionRegion**
PUT	
Description	It is used to udpate the motion detection configuration for a video input channel.
Query	None
Inbound Data	**MotionDetectionRegion**
Success Return	**ResponseStatus**
Notes: If motion detection is supported by the device, a motion detection ID will be allocated for	

each video input channel ID. The motion detection ID must correspond to the video input channel ID.

MotionDetectionRegion XML Block

```
<MotionDetectionRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>    <!-- req, xs:string;id --></id>
  <enabled>  <!-- req, xs:boolean -->  </enabled>
  <sensitivityLevel>    <!-- req -->
    <!-- req, xs:integer, 0..100, 0 is least sensitive -->
  </sensitivityLevel>
  <daySensitivityLevel>    <!-- dep -->
    <!-- req, xs:integer, 0..100, 0 is least sensitive-->
  </daySensitivityLevel>
  <nightSensitivityLevel>   <!-- dep -->
    <!-- req, xs:integer, 0..100, 0 is least sensitive-->
  </nightSensitivityLevel>
  <objectSize>    <!-- dep -->
     <!-- req, xs:integer, 0..100, 0 is least sensitive -->
  </objectSize>
  <dayObjectSize>    <!-- dep -->
     <!-- req, xs:integer, 0..100, 0 is least sensitive -->
  </dayObjectSize>
  <nightObjectSize>   <!-- dep -->
     <!-- req, xs:integer, 0..100, 0 is least sensitive -->
  </nightObjectSize>
  <RegionCoordinatesList>      <!-- req -->
    <RegionCoordinates>  <!-- Note: at least two coordinates are required -->
      <positionX>  <!-- req, xs:integer -->  </positionX>
      <positionY>  <!-- req, xs:integer -->  </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</MotionDetectionRegion>
```

8.4.20 /ISAPI/System/Video/inputs/channels/<ID>/motionDetectionExt/switch

/ISAPI/System/Video/inputs/channels/ID/motionDetectionExt/switch	General Resource	v2.0

GET	
Description	It is used to get the motion detection switch day and night settings.
Query	None

Inbound Data	None
Success Return	**MotionDetectionSwitch**
PUT	
Description	It is used to udpate the motion detection switch day and night settings.
Query	None
Inbound Data	**MotionDetectionSwitch**
Success Return	**ResponseStatus**
Notes: If motion detection is supported by the device, a motion detection ID will be allocated for each video input channel ID. The motion detection ID must correspond to the video input channel ID.	

MotionDetectionSwitch XML Block

```
<MotionDetectionSwitch version="2.0"
 xmlns="http://www.isapi.com/ver10/XMLSchema">
  <type>
    <!-- opt, xs:string, "off,auto,schedule"-->
  </type>
  <Schedule> <!--dep-->
    <scheduleType><!—req,xs:string,"day,night"></scheduleType>
    <TimeRange> <!-- req -->
      <beginTime> <!-- req, xs:time, ISO8601 time --> </beginTime>
      <endTime> <!-- req, xs:time, ISO8601 time --> </endTime>
    </TimeRange>
  </Schedule>
</MotionDetectionSwitch>
```

8.4.21 /ISAPI/System/Video/inputs/channels/<ID>/overlays

/ISAPI/System/Video/inputs/channels/*ID*/overlays	**General Resource**	**v2.0**
GET		
Description	It is used to get the overlays configuration for a video input channel.	
Query	None	
Inbound Data	None	
Success Return	**VideoOverlay**	
PUT		

Description	It is used to update the overlays configuration for a video input channel.
Query	None
Inbound Data	**VideoOverlay**
Success Return	**ResponseStatus**

VideoOverlay XML Block

```
<VideoOverlay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <normalizedScreenSize>  <!--req-->
    <normalizedScreenWidth> <!--ro, req, xs:integer --> </normalizedScreenWidth>
    <normalizedScreenHeight> <!--ro,  req, xs:integer --> </normalizedScreenHeight>
  <normalizedScreenSize>
  <attribute> <!--opt-->
      <transparent> <!-- req, xs:boolean --></transparent>
      <flashing> <!-- req, xs:boolean--> <flashing>
  </attribute>
  <TextOverlayList/>  <!-- opt -->
  < DateTimeOverlay /> <!-- opt -->
  < channelNameOverlay /> <!-- opt -->
  <fontSize> <!-- opt, xs:integer, pixels --> </fontSize>
</VideoOverlay>
```

8.4.22 /ISAPI/System/Video/inputs/channels/<ID>/overlays/text

/ISAPI/System/Video/inputs/channels/*ID*/overlays/text	**General Resource v2.0**
GET	
Description	It is used to get the text overlays configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**TextOverlayList**
PUT	
Description	It is used to update the text overlays configuration for a video input channel.
Query	None
Inbound Data	**TextOverlayList**
Success Return	**ResponseStatus**
POST	

Description	It is used to add a text overlay for a video input channel.
Query	None
Inbound Data	**TextOverlay**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the text overlays configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: A set of text overlays is managed. They are composited over the video signal in increasing ID-order.	

TextOverlayList XML Block

```
<TextOverlayList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <TextOverlay/>     <!-- opt -->
</TextOverlayList>
```

8.4.23 /ISAPI/System/Video/inputs/channels/<ID>/overlays/text/<ID>

/ISAPI/System/Video/inputs/channels/*ID*/overlays/text/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular text overlay configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**TextOverlay**
PUT	
Description	It is used to update a particular text overlay configuration for a video input channel.
Query	None
Inbound Data	**TextOverlay**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular text overlay configuration for a video input channel.

Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Position coordinates are dependent on normalized screen size. The computer screen coordinate system is used, which the origin coordinate is on top-left corner, the Y axis is vertical downwards, the X axis is horizontal rightwards.	

TextOverlay XML Block

```
<TextOverlay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id>              <!-- req, xs:string;id -->        </id>
   <enabled>         <!-- req, xs:boolean -->         </enabled>
   <positionX>       <!-- req, xs:float -->          </positionX>
   <positionY>       <!-- req, xs:float -->          </positionY>
   <displayText>      <!-- req, xs:string -->       </displayText>
</TextOverlay>
```

8.4.24 /ISAPI/System/Video/inputs/channels/<ID>/overlays/channelNameOverlay

/ISAPI/System/Video/inputs/channels/*ID*/overlays/channelNameOverlay	**General Resource v2.0**
GET	
Description	It is used to get a particular channel name configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**channelNameOverlay**
PUT	
Description	It is used to update a particular channel name configuration for a video input channel.
Query	None
Inbound Data	**channelNameOverlay**
Success Return	**ResponseStatus**
Notes: Position coordinates are dependent on normalized screen size. The computer screen coordinate system is used, which the origin coordinate is on top-left corner, the Y axis is vertical downwards, the X axis is horizontal rightwards.	

channelNameOverlay XML Block

```
<channelNameOverlay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean -->  </enabled>  //Show channel name
  <positionX> <!-- req, xs:integer;coordinate --> </positionX>
  <positionY> <!-- req, xs:integer;coordinate --> </positionY>
</channelNameOverlay>
```

8.4.25 /ISAPI/System/Video/inputs/channels/<ID>/overlays/dateTimeOverlay

/ISAPI/System/Video/inputs/channels/*ID*/overlays/dateTime	**General Resource v2.0**
GET	
Description	It is used to get the OSD configuration for a video input channel.
Query	None
Inbound Data	None
Success Return	**DatetimeOverlay**
PUT	
Description	It is used to update the OSD configuration for a video input channel.
Query	None
Inbound Data	**DatetimeOverlay**
Success Return	**ResponseStatus**
Notes: Position coordinates are dependent on normalized screen size. The computer screen coordinate system is used, which the origin coordinate is on top-left corner, the Y axis is vertical downwards, the X axis is horizontal rightwards.	

DateTimeOverlay XML Block

```
<DateTimeOverlay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean -->  </enabled>//check whether to display date
  <positionX> <!-- req, xs:integer;coordinate --> </positionX>
  <positionY> <!-- req, xs:integer;coordinate --> </positionY>
  <dateStyle>
    <!-- opt, xs:string, "YYYY-MM-DD, MM-DD-YYYY, DD-MM-YYYY, CHR-YYYY-MM-DD, CHR-MM-DD-YYYY, CHR-DD-MM-YYYY" -->
  </dateStyle>
  <timeStyle>  <!--opt, xs:string, "12hour, 24hour" --> </timeStyle>
  <displayWeek> <!-- opt, xs:boolean --> </displayWeek>//check whether to display week
</DateTimeOverlay>
```

8.4.26 /ISAPI/System/Video/inputs/channels/<ID>/image

/ISAPI/System/Video/inputs/channels/*ID*/image	General Resource v2.0
GET	
Description	Access on-screen Image for a special channel.
Query	None
Inbound Data	None
Success Return	**ImageOverlayList**
PUT	
Description	Configure the on-screen Image for a special channel.
Query	None
Inbound Data	**ImageOverlayList**
Success Return	**ResponseStatus**
Notes:	

ImageOverlayList XML Block

```
<ImageOverlayList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ImageOverlay/> <!-- opt -->
</ImageOverlayList>
```

8.4.27 /ISAPI/System/Video/inputs/channels/<ID>/image/<ID>

/ISAPI/System/Video/inputs/channels/*ID*/image/ID	General Resource v2.0
GET	
Description	Access on-screen Image for a special channel.
Query	None
Inbound Data	None
Success Return	**ImageOverlay**
PUT	
Description	Configure the on-screen Image for a special channel.
Query	None
Inbound Data	**ImageOverlay**
Success Return	**ResponseStatus**
Notes:	

ImageOverlay XML Block

```
<ImageOverlay version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string;id --> </id>
  <enabled> <!-- req, xs:boolean --> </enabled>
  <imageName> <!-- req, xs:string --> </imageName>
  <positionX> <!-- opt, xs:integer;coordinate --> </positionX>
  <positionY> <!-- opt, xs:integer;coordinate --> </positionY>
  <transparentColorEnabled> <!-- opt, xs:boolean --> </transparentColorEnabled>
  <transparentColor> <!-- dep, xs:hexBinary;color --> </transparentColor>
  <imageWidth> <!--opt, xs:integer--> </imageWidth>
  <imageHeight> <!--opt, xs:integer--> </imageHeight>
</ImageOverlay>
```

8.4.28 /ISAPI/System/Video/inputs/channels/<ID>/image/picture

/ISAPI/System/Video/inputs/channels/*ID*/image/picture	
POST	
Description	Configure the on-screen Image for a special channel.
Query	None
Inbound Data	**Picture over HTTP**
Success Return	**ResponseStatus**
Notes:	

8.4.29 /ISAPI/System/Video/outputs

/ISAPI/System/Video/outputs	**General Resource v2.0**
GET	
Description	It is used to get the video outputs configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoOutput**
Notes: An IP media device may contain a set of video outputs. These outputs are hardwired by the device, meaning that the IDs can be discovered but not created or deleted.	

VideoOutput XML Block

```
<VideoOutput version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <VideoOutputChannelList/>  <!-- opt -->
</VideoOutput>
```

8.4.30 /ISAPI/System/Video/outputs/channels

ISAPI/System/Video/outputs/channels	General Resource v2.0
GET	
Description	It is used to get the video output channels configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoOutputChannelList**
Notes: Since video output channels are resources that are defined by the hardware configuration of the device, they cannot be created or deleted.	

VideoOutputChannelList XML Block

```
<VideoOutputChannelList version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <VideoOutputChannel/>      <!-- opt -->
</VideoOutputChannelList>
```

8.4.31 /ISAPI/System/Video/outputs/channels/<ID>

ISAPI/System/Video/outputs/channels/<ID>	General Resource v2.0
GET	
Description	It is used to get a particular video input channel configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**VideoOutputChannel**
Notes: <menu> required if the port support display menu. <mirrorMenu> check whether to support to display menu of another port simultaneously	

VideoOutputChannel XML Block

```
<VideoOutputChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>    <!-- req, xs:string;id -->    </id>
  <type>   <!-- req, xs:string, "VGA,CVBS,HDMI,Spot,SDI" --> </type>
  <menu>   <!-- dep, ro -->
    <mirrorMenu> <!--req, xs:boolean--> </mirrorMenu>
  <menu/>
 <mode> <!--opt,xs:string,"close,clip,scale,open,SDI_1080P25…"--> <mode>
 <resolution> <!--opt, xs:string; "1920*1080/60HZ,1280*720/50HZ..." --> </resolution>
</VideoOutputChannel>
```

8.4.32 /ISAPI/System/Video/Menu

URI	/ISAPI/System/Video/Menu		Type	Resource
Function	Access the local menu configuration on an IP media device.			
Methods	Query String(s)	Inbound Data	Return Result	
GET			<MenuList>	
Notes	An IP media device may contain a set of local menus. These menus are hardwired by the device, meaning that the IDs can be discovered but not created or deleted. ID numbering or values should be considered arbitrary and			

MenuList XML Block

```
<MenuList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Menu/>   <!-- opt   -->
</MenuList>
```

8.4.33 /ISAPI/System/Video/Menu/<ID>

URI	/ISAPI/System/Video/Menu/<ID>		Type	Resource
Function	Access menu configuration.			
Methods	Query String(s)	Inbound Data	Return Result	
GET		None	<Menu>	
PUT		<Menu>	<ResponseStatus>	
Notes	If(mode == auto) VideoOutputPortList is ro			

Menu XML Block

```
<Menu version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mode> <!--req, xs:string, "auto, manual" --> </mode>
  <VideoOutputPortList>  <!-- req -->
    <videoOutputPortID> <!-- opt, xs:string, id-->  </videoOutputPortID>
  </VideoOutputPortList>
</Menu>
```

8.5 /ISAPI/System/Audio

/ISAPI/System/Audio	Service	v2.0
Notes:		

8.5.1/ISAPI/System/Audio/capabilities

/ISAPI/System/Audio/capabilities	General Resource	v2.0
GET		
Description	It is used to get audio capability.	
Query	None	
Inbound Data	None	
Success Return	**<AudioCap>**	
Notes:		

AudioCap XML Block

```
<AudioCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <audioInputNums> <!-- req, xs:integer> </audioInputNums>
  <audioOutputNums> <!-- req, xs:integer> </audioOutputNums>
</AudioCap>
```

8.5.2/ISAPI/System/Audio/channels

/ISAPI/System//Audio/channels	General Resource	v2.0
GET		
Description	It is used to get the audio channels configuration on an IP media device.	
Query	None	
Inbound Data	None	
Success Return	**AudioChannelList**	
Notes:		

Since inputs are resources that are defined by the hardware configuration of the device, audio channels cannot be created or deleted.

AudioChannelList XML Block

```
<AudioChannelList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AudioChannel/>  <!-- opt -->
</AudioChannelList>
```

8.5.3/ISAPI/System/Audio/channels/<ID>

/ISAPI/System/Audio/channels/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular audio channel configuration on an IP media device.
Query	None
Inbound Data	None
Success Return	**AudioChannel**
Notes: <audioMode> is the duplex mode for audio transmission between the client and media device.	

AudioChannel XML Block

```
<AudioChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>          <!-- req, xs:string -->          </id>
  <enabled>     <!-- req, xs:boolean -->        </enabled>
</AudioChannel>
```

8.5.4/ISAPI/System/TwoWayAudio

/ISAPI/System/TwoWayAudio	**General Resource v2.0**
Notes: two way audio Service.	

8.5.5/ISAPI/System/TwoWayAudio/channels

/ISAPI/System/TwoWayAudio/channels	**General Resource v2.0**

GET	
Description	It is used to get the two way audio channels list
Query	None
Inbound Data	None
Success Return	**TwoWayAudioChannelList**
Notes:	

TwoWayAudioChannelList XML Block

```
<TwoWayAudioChannelList                                    version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <TwoWayAudioChannel/>     <!-- opt -->
</TwoWayAudioChannelList>
```

8.5.6/ISAPI/System/TwoWayAudio/channels/<ID>

/ISAPI/System/TwoWayAudio/channels/ID	General Resource v2.0
GET	
Description	It is used to get a particular two way audio channel
Query	None
Inbound Data	None
Success Return	**TwoWayAudioChannel**
PUT	
Description	It is used to get a particular transparent channel
Query	None
Inbound Data	**TwoWayAudioChannel**
Success Return	**ResponseStatus**
Notes: When <enabled>is true, two way audio is open; otherwise two way audio is closed. When <audioCompressionType> is MP212, <audioBitRate> supports to set bit rate.	

TwoWayAudioChannel XML Block

```
<TwoWayAudioChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>   <!-- req, xs:string;id -->   </id>
  <enabled>   <!-- req, xs:boolean -->   </enabled>
  <audioCompressionType>
      <!-- req, xs:string,
       "  G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPC
M,MP2L2"
```

```
        -->
    </audioCompressionType>
    <audioInboundCompressionType>
        <!-- opt, xs:string,
        “ G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM
”
        -->
    </audioInboundCompressionType>
    <speakerVolume> <!-- opt, xs:int--> </speakerVolume>
    <microphoneVolume> <!-- opt, xs:int--> </microphoneVolume>
    <noisereduce> <!-- opt, xs: Boolean,"true, false" --> </noisereduce>
    <audioBitRate> <!-- opt, xs:integer;kbs--> </audioBitRate>
    <audioInputType > <!-- opt, xs:string, “MicIn, LineIn”--> </audioInputType>
    <associateVideoInputs> <!-- opt -->
      <enabled> <!-- req, xs:Boolean --> </enabled>
      <videoInputChannelList>     <!-- req -->
        <videoInputChannelID> <!-- opt, xs:string; id --> </videoInputChannelID>
      </videoInputChannelList>
    </associateVideoInputs>
</TwoWayAudioChannel>
```

8.5.7/ISAPI/System/TwoWayAudio/channels/<ID>/open

/ISAPI/System/TwoWayAudio/channels/ID/open	General Resource v2.0
PUT	
Description	It is used to open the two way audio channel.
Query	None
Inbound Data	None
Success Return	**TwoWayAudioSession**
Notes: In sessionId 8.6.5, if send Voice data，need to use this field to represent the communication on which session.	

TwoWayAudioSession XML Block

```
<TwoWayAudioSession version=“2.0” xmlns=”http://www.isapi.org/ver20/XMLSchema”>
    <sessionId>     <!-- req, xs:string -->    </sessionId>
```

```
</TwoWayAudioSession>
```

8.5.8/ISAPI/System/TwoWayAudio/channels/<ID>/close

/ISAPI/System/TwoWayAudio/channels/ID/close	General Resource v2.0
PUT	
Description	It is used to close the two way audio channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

8.5.9/ISAPI/System/TwoWayAudio/channels/<ID>/audioData

/ISAPI/System/TwoWayAudio/channels/ID/audioData	General Resource v2.0
GET	
Description	It is used to get data on the transparent channel.
Query	**sessionId**
Inbound Data	**Raw Data**
Success Return	**ResponseStatus**
PUT	
Description	It is used to send data on the transparent channel.
Query	None
Inbound Data	**Raw Data**
Success Return	**ResponseStatus**
Notes:	

Example：Client sends audio data to server

```
PUT /ISAPI/System/TwoWayAudio/channels/ID/transData HTTP 1.1
…
Content-Type: application/binary; charset=”UTF-8”\r\n
\r\n
TwowayAudio Data…
```

…

Example：Client receives audio data from server

```
GET /ISAPI/System/TwoWayAudio/channels/ID/transData HTTP/1.1

HTTP/1.1 200 OK
…
Content-Type: application/binary; charset=”UTF-8”\r\n
\r\n
TwoWayAudio Data……
```

8.6 /ISAPI/System/Serial

/ISAPI/System/Serial	Service	v2.0
Notes: Serial port service.		

8.6.1/ISAPI/SystemSerial/capabilities

/ISAPI/System/Serial/capabilities	General Resource	v2.0
GET		
Description	It is used to get device capability.	
Query	None	
Inbound Data	None	
Success Return	**<SerialCap>**	
Notes:		

SerialCap XML Block

```
<SerialCap version="2.0" xmlns=“http://www.isapi.org/ver20/XMLSchema”>
   <rs485PortNums> <!-- opt, xs:integer --> </rs485PortNums>
   <supportRS232Config> <!-- opt, xs:Boolean --> </supportRS232Config>
   <rs422PortNums> <!-- opt, xs:integer--> </rs422PortNums>
   <rs232PortNums> <!-- opt, xs:integer-->   </rs232PortNums>
</SerialCap>
```

8.6.2/ISAPI/System/Serial/ports

/ISAPI/System/Serial/ports	General Resource	v2.0

GET	
Description	It is used to get the list of serial ports supported by the device.
Query	None
Inbound Data	None
Success Return	**SerialPorList**
Notes: Since serial ports are resources that are defined by the hardware configuration of the device, they cannot be created or deleted.	

SerialPortList XML Block

```
<SerialPortList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <SerialPort/>   <!-- opt -->
</SerialPortList>
```

8.6.3/ISAPI/System/Serial/ports/<ID>

/Serial/ports/*ID*	General Resource v2.0
GET	
Description	It is used to get the configuration of a serial port supported by the device.
Query	None
Inbound Data	None
Success Return	**SerialPort**
PUT	
Description	It is used to update the configuration of a serial port supported by the device.
Query	None
Inbound Data	**SerialPort**
Success Return	**ResponseStatus**
Notes: Access to the serial port parameters. <serialPortType> set the type of port; RS232, RS485, etc. <direction> indicates whether the port is bidirectional. <duplexMode> indicates whether the serial port runs in full or half duplex mode. <workMode> is required only when serial port type is RS232	

SerialPort XML Block

```
<SerialPort version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
```

```
  <id>   <!-- req, xs:string;id -->      </id>
  <enabled> <!-- req, xs:boolean -->      </enabled>
  <serialPortType><!-- req, xs:string, “RS485,RS422,RS232” --></serialPortType>
  <duplexMode> <!-- req, xs:string, “half,full” -->  </duplexMode>
  <direction><!-- req, xs:string, “monodirectional,bdirectional” --> </direction>
  <baudRate><!-- req, xs:integer --></baudRate>
  <dataBits> <!-- req, xs:integer --> </dataBits>
  <parityType><!-- req, xs:string, “none,even,odd,mark,space” --> </parityType>
  <stopBits> <!-- req, xs:string, “1,1.5,2” -->  </stopBits>
  <workMode> <!--dep, xs:string, “console, transparent” --> </workMode>
  <flowCtrl> <!-- req, xs:string, “none, software, hardware” --> </flowCtrl>
</SerialPort>
```

8.6.4/ISAPI/System/Serial/ports/<ID>/Transparent

/ISAPI/System/Serial/ports/*ID*/Transparent	**General Resource**	**v2.0**
Notes: Transparent Service.		

8.6.5/ISAPI/System/Serial/ports/<ID>/Transparent/channels

/ISAPI/System/Serial/ports/*ID*/Transparent/channels	**General Resource**	**v2.0**
GET		
Description	It is used to get the transparent channels list	
Query	None	
Inbound Data	None	
Success Return	**TransparentChannelList**	
Notes:		

TransparentChannelList XML Block

```
<TransparentChannelList version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <TransparentChannel/>  <!-- opt -->
</TransparentChannelList>
```

8.6.6/ISAPI/System/Serial/ports/<ID>/Transparent/channels/<ID>

/ISAPI/System/Serial/ports/*ID*/Transparent/channels/ID	General Resource v2.0
GET	
Description	It is used to get a particular transparent channel
Query	None
Inbound Data	None
Success Return	**TransparentChannel**
PUT	
Description	It is used to get a particular transparent channel
Query	None
Inbound Data	**TransparentChannel**
Success Return	**ResponseStatus**
Notes:	

TransparentChannel XML Block

```
<TransparentChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id> <!-- req, xs:string;id --> </id>
   <enabled> <!-- req, xs:boolean --> </enabled>
   <serialPortID>   <!--req,ro, xs:string; id -->   </serialPortID>
</TransparentChannel>
```

8.6.7/ISAPI/System/Serial/ports/<ID>/Transparent/channels/<ID>/open

/ISAPI/System/Serial/ports/*ID*/Transparent/channels/ID/open	General Resource v2.0
PUT	
Description	It is used to open the transparent channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

Only support RS485 transparent channel.

8.6.8/ISAPI/System/Serial/ports/<ID>/Transparent/channels/<ID>/close

/ISAPI/System/Serial/ports/*ID*/Transparent/channels/ID/close	**General Resource v2.0**
PUT	
Description	It is used to close the transparent channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

8.6.9/ISAPI/System/Serial/ports/<ID>/Transparent/channels/<ID>/transData

/ISAPI/System/Serial/ports/*ID*/Transparent/channels/ID/transData	**General Resource v2.0**
GET	
Description	It is used to get data on the transparent channel.
Query	None
Inbound Data	**Raw Data**
Success Return	**ResponseStatus**
PUT	
Description	It is used to send data on the transparent channel.
Query	None
Inbound Data	**Raw Data**
Success Return	**ResponseStatus**
Notes:	

Example：

```
GET /ISAPI/System/Serial/ports/ID/Transparent/channels/ID/transData HTTP/1.1
```

```
HTTP/1.1 200 OK
…
Content-Type: application/binary; charset=”UTF-8”
Content-Length: ISAPI
\r\n
Raw data…
```

Example：

```
PUT /ISAPI/System/Serial/ports/ID/Transparent/channels/ID/transData HTTP/1.1
…
Content-Type: application/binary; charset=”UTF-8”
\r\n
Raw data…
```

8.7 /ISAPI/System/Hardware/

/ISAPI/System/Hardware/	**Service v2.0**
Notes:	

8.7.1/ISAPI/System/Hardware/irLightSwitch

/ISAPI/System/Hardware/irLightSwitch	**General Resource v2.0**
GET	
Description	
Query	None
Inbound Data	None
Success Return	**IrLightSwitch**
PUT	
Description	
Query	None
Inbound Data	**IrLightSwitch**
Success Return	**ResponseStatus**
Notes:	

IrLightSwitch XML Block

```
<IrLightSwitch version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
    <mode>  <!-- req, xs:string,”open,close” -->  </mode>
</IrLightSwitch>
```

8.8 /ISAPI/Security

/ISAPI/Security	Service	v2.0
Notes:		

8.8.1/ISAPI/Security/capabilities

/ISAPI/Security/capabilities	General Resource	v2.0
GET		
Description	It is used to get security capability.	
Query	None	
Inbound Data	None	
Success Return	**<SecurityCap>**	
Notes:		

SecurityCap XML Block

```
<SecurityCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <supportUserNums>    <!-- opt, xs:integer --> < supportUserNums>
   <userBondIpNums>    <!-- opt, xs:integer --> <userBondIpNums>
   <userBondMacNums> <!-- opt, xs:integer --> < userBondIpNums >
</SecurityCap>
```

8.8.2/ISAPI/Security/users

/ISAPI/Security/users	General Resource	v2.0
GET		
Description	It is used to get the user list for the device.	
Query	None	
Inbound Data	None	
Success Return	**UserList**	
PUT		
Description	It is used to update the user list for the device.	
Query	None	
Inbound Data	**UserList**	
Success Return	**ResponseStatus**	

POST	
Description	It is used to add a user for the device.
Query	None
Inbound Data	**User**
Success Return	**ResponseStatus**
DELETE	**Administrator**
Description	It is used to delete the user list for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: A default user account, “admin”, must be provided. Its default password is “12345”. It has an Administrator user level, and must not be deleted. Passwords can only be uploaded - they are never revealed during GET operations.	

UserList XML Block

```
<UserList version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
   <User/>       <!-- opt -->
</UserList>
```

8.8.3/ISAPI/Security/users/<ID>

/ISAPI/Security/users/*ID*	**General Resource v2.0**
GET	**Viewer**
Description	It is used to get a particular user configuration for the device.
Query	None
Inbound Data	None
Success Return	**User**
PUT	**Administrator**
Description	It is used to update a particular user configuration for the device.
Query	None
Inbound Data	**User**
Success Return	**ResponseStatus**
DELETE	**Administrator**
Description	It is used to delete a particular user for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: <id> of “admin” account is 1. “admin” account must not be deleted.	

<password> is a write-only field.

User XML Block

```
<User version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>              <!-- req, xs:integer, "1-16" -->          </id>
  <userName>          <!-- req, xs:string -->          </userName>
  <password>          <!-- wo, req, xs:string -->       </password>
  <bondIpAddressList>
      <bondIpAddress/>
  </ bondIpAddressList>
  <bondMacAddressList>
      <bondMacAddress/>
  </ bondMacAddressList>
  <userLevel> <!-- opt, xs:string, "Administrator, Operator, Viewer" --> </userLevel>
  <attribute> <!-- opt -->
    <inherent> <!--xs:boolean --> </inherent>
  </attribute>
</User>
```

bondIpAddress XML Block

```
  < bondIpAddress>
      <id>              <!-- req, xs:integer -->          </id>
      <ipAddress>    <!-- dep, xs:string -->  </ipAddress>
      <ipv6Address> <!-- dep, xs:string -->  </ipv6Address>
  </ bondIpAddress>
```

bondMacAddress XML Block

```
  < bondMacAddress>
      <id>              <!-- req, xs:integer -->          </id>
      <macAddress> <!-- opt, xs:string --> </macAddress>
  </ bondMacAddress>
```

8.8.4/ISAPI/Security/adminAccesses

/ISAPI/Security/adminAccesses	General Resource v2.0
GET	**Viewer**
Description	It is used to get administrative access protocol for the device.
Query	None
Inbound Data	None
Success Return	**AdminAccessProtocolList**
PUT	**Administrator**

Description	It is used to update administrative access protocol for the device.
Query	None
Inbound Data	**AdminAccessProtocolList**
Success Return	**ResponseStatus**
Notes: <protocol> is the protocol name for admin access, i.e. “HTTP”, “HTTPS”, etc.	

AdminAccessProtocolList XML Block

```
AdminAccessProtocolList version=“2.0”
  xmlns=“http://www.isapi.org/ver20/XMLSchema”>
     < AdminAccessProtocol />
</ AdminAccessProtocolList >
```

8.8.5/ISAPI/Security/adminAccesses/<ID>

/ISAPI/Security/adminAccesses/ID	**General Resource v2.0**
GET	**Viewer**
Description	It is used to get administrative access protocol for the device.
Query	None
Inbound Data	None
Success Return	**AdminAccessProtocol**
PUT	**Administrator**
Description	It is used to update administrative access protocol for the device.
Query	None
Inbound Data	**AdminAccessProtocol**
Success Return	**ResponseStatus**
Notes: <protocol> is the protocol name for admin access, i.e. “HTTP”, “HTTPS”, etc.	

AdminAccessProtocol XML Block

```
<AdminAccessProtocol version=“2.0”
  xmlns=“http://www.isapi.org/ver20/XMLSchema”>
    <id>       <!-- req, xs:string;id -->       </id>
    <protocol> <!-- req, xs:string; “HTTP, HTTPS,RTSP,DEV_MANAGE” --> </protocol>
    <portNo> <!-- req, xs:integer -->    </portNo>
</AdminAccessProtocol>
```

8.8.6/ISAPI/Security/userCheck

/ISAPI/Security/userCheck	General Resource v2.0
GET	
Description	It is used to check is password matche user name.
Query	None
Inbound Data	None
Success Return	**userCheck**
Notes: The device always returns HTTP 200/OK no matter whether userCheck is successful, or not. The client software checks user name/password via <statusValue>. If the value is 200, it means match, otherwise, if the value is 401, it means mismatch.	

userCheck XML Block

```
<userCheck version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <statusValue>     <!-- req, xs:integer, '200, 401' --> </statusValue>
   <statusString>   <!-- opt, xs:string, 'OK, Unauthorized --> </statusString>
</userCheck>
```

8.8.7/ISAPI/Security/UserPermission

/ISAPI/Security/UserPermission	General Resource v2.0
GET	
Description	It is used to get user permission of the device.
Query	None
Inbound Data	None
Success Return	**UserPermissionList**
PUT	
Description	It is used to set user permission of the device.
Query	None
Inbound Data	**UserPermissionList**
Success Return	**ResponseStatus**
Notes: only the user "admin" has the right to review or edit user's permission.	

UserPermissionList XML Block

```
<UserPermissionList version="2.0" xmlns=" http://www.isapi.org/ver20/XMLSchema">
   <UserPermission/>  <!-- opt -->
```

```
</UserPermissionList>
```

8.8.8/ISAPI/Security/UserPermission/<ID>

/ISAPI/Security/UserPermission/ID	General Resource v2.0
GET	
Description	It is used to get a particular user's permission
Query	None
Inbound Data	None
Success Return	**UserPermission**
PUT	
Description	It is used to set a particular user's permission
Query	None
Inbound Data	**UserPermission**
Success Return	**ResponseStatus**
Notes: <userID> links the user permission to a user, see /ISAPI/Security/AAA/users/ID. <userType> The type value of the user, which can be 'admin', 'operator' or 'viewer'. 'admin' is the administrator of the IPMD, it have all permissions. 'operator' and 'viewer' have default permission policy. The default permission policy can be edited by providing <localPermission>, <remotePermission>.	

UserPermission XML Block

```
<UserPermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id> <!—req, xs:string !--> </id>
   <userID>   <!--req, xs:string; id -->   </userID>
   <userType>   <!-- req, xs:string, "admin, operator, viewer"-->   </userType>\
   <localPermission/>  <!-- opt -->
   <remotePermission/>   <!-- opt -->
</UserPermission>
```

8.8.9/ISAPI/Security/UserPermission/<ID>/localPermission

/ISAPI/Security/UserPermission/ID/localPermission	General Resource v2.0
GET	
Description	It is used to get a particular user's local permission

Query	None
Inbound Data	None
Success Return	**localPermission**
PUT	
Description	It is used to set a particular user's local permission
Query	None
Inbound Data	**localPermission**
Success Return	**ResponseStatus**
Notes:	

localPermission XML Block

```
<localPermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <backup> <!-- opt, xs:boolean --> </backup>
  <record> <!-- opt, xs:boolean --> </record>
  <playBack> <!-- opt, xs:boolean --> </playBack>
  <videoChannelPermissionList>    <!-- opt -->
    <videoChannelPermission> <!-- opt -->
      <id> <!-- req, must correspond to the video input channel id --> </id>
      <playBack> <!-- opt, xs:boolean --> </playBack>
      <record> <!-- opt, xs:boolean --> </record>
      <backup> <!-- opt, xs:boolean --> </backup>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl> <!-- req, xs:boolean --> </ptzControl>
  <ptzChannelPermissionList>   <!-- opt -->
    <ptzChannelPermission> <!-- req -->
      <id> <!--req, must correspond to ptz id, see /ISAPI/PTZCtrl/channels/ID--> </id>
      <ptzControl> <!-- opt, xs:boolean --> </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck> <!-- opt, xs:boolean --> </logOrStateCheck>
  <parameterConfig> <!--opt, xs:boolean --> </parameterConfig>
  <restartOrShutdown> <!--opt, xs:boolean --> </restartOrShutdown>
  <upgrade> <!--opt, xs:boolean --> </upgrade>
</localPermission>
```

8.8.10 /ISAPI/Security/UserPermission/<ID>/remotePermission

/ISAPI/Security/UserPermission/ID/remotePermission	**General Resource**

	v2.0
GET	
Description	It is used to get a particular user's remote permission
Query	None
Inbound Data	None
Success Return	**remotePermission**
PUT	
Description	It is used to set a particular user's remote permission
Query	None
Inbound Data	**remotePermission**
Success Return	**ResponseStatus**
Notes:	

remotePermission XML Block

```
<remotePermission version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <record> <!-- opt, xs:boolean --> </record>
  <playBack> <!-- opt, xs:boolean --> </playBack>
  <preview> <!-- opt, xs:boolean --> </preview>
  <videoChannelPermissionList>    <!-- opt -->
    <videoChannelPermission> <!-- opt -->
      <id> <!-- req, must correspond to the video input channel id --> </id>
      <preview> <!-- opt, xs:boolean --> </preview>
      <palyBack> <!-- opt, xs:boolean --> </palyBack>
      <record> <!--opt, xs:Boolean --> </record>
    </videoChannelPermission>
  </videoChannelPermissionList>
  <ptzControl> <!-- opt, xs:boolean --> </ptzControl>
  <ptzChannelPermissionList>   <!-- opt -->
    <ptzChannelPermission> <!-- opt -->
      <id> <!--req, must correspond to ptz id, see /ISAPI/PTZCtrl/channels/ID--> </id>
      <ptzControl> <!-- opt, xs:boolean --> </ptzControl>
    </ptzChannelPermission>
  </ptzChannelPermissionList>
  <logOrStateCheck> <!-- opt, xs:boolean --> </logOrStateCheck>
  <parameterConfig> <!--opt, xs:boolean --> </parameterConfig>
  <restartOrShutdown> <!--opt, xs:boolean --> </restartOrShutdown>
  <upgrade> <!--opt, xs:boolean --> </upgrade>
  <voiceTalk> <!--opt, xs:boolean --> </voiceTalk>
  <transParentChannel> <!--opt, xs:boolean --> <transParentChannel>
  <contorlLocalOut> <!-- opt, xs:boolean --> </contorlLocalOut>
  <alarmOutOrUpload> <!-- opt, xs:boolean --> </alarmOutOrUpload>
</remotePermission>
```

8.8.11 /ISAPI/Security/UserPermission/anonymouslogin

/ISAPI/Security/UserPermission/anonymouslogin	General Resource v2.0
GET	
Description	Access and configure the user's permission.
Query	None
Inbound Data	None
Success Return	**anonymouslogin**
PUT	
Description	Access and configure the user's permission.
Query	None
Inbound Data	**anonymouslogin**
Success Return	**ResponseStatus**
Notes: Anonymouslogin owns corresponding privilege of interfaces below: /ISAPI/Streaming/channels is used to get the resolution /ISAPI/Security/userCheck /ISAPI/System/Network/interfaces and /ISAPI/System/Network/UPnP/ports/status /ISAPI/Security/adminAccesses	

anonymouslogin XML Block

```
<anonymouslogin version="2.0" xmlns=" http://www.isapi.org/ver20/XMLSchema">
  <enabled>                    <!-- req, xs:boolean -->    </enabled>
</anonymouslogin>
```

8.8.12 /ISAPI/Security/UserPermission/operatorCap

/ISAPI/Security/UserPermission/operatorCap	General Resource v2.0
GET	
Description	It is used to get default capabilities of operator.
Query	None
Inbound Data	None
Success Return	**UserPermissionCap**
Notes:	

UserPermissionCap XML Block

```
<UserPermissionCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <userType>  <!-- req, xs:string, "admin, operator, viewer"-->  </userType>
  <localPermissionCap>  <!-- opt -->
  </localPermissionCap>
  <remotePermissionCap>  <!-- opt -->
  </remotePermissionCap>
</UserPermissionCap>
```

8.8.13 /ISAPI/Security/UserPermission/viewerCap

/ISAPI/Security/UserPermission/viewerCap	General Resource v2.0
GET	
Description	It is used to get default capabilities of viewer.
Query	None
Inbound Data	None
Success Return	**UserPermissionCap**
Notes:	

8.8.14 /ISAPI/Security/deviceCertificate

/ISAPI/Security/deviceCertificate	General Resource v2.0
GET	
Description	This function is used to upload a user certificate to the device. The user certificate is used for 802.1x (radius) with various authentication mechanisms.
Query	None
Inbound Data	None
Success Return	Data
PUT	
Description	This function is used to upload a user certificate to the device. The user certificate is used for 802.1x (radius) with various authentication mechanisms.
Query	None
Inbound Data	Data
Success Return	**ResponseStatus**
Notes:	

The format of the certificate is device-dependent.

Distinguish different certificate by Content-type：
CA- certificate (root certificate): application/x-x509-ca-cert
Clent certificate: application/x-x509-client-cert
Clent password : application/x-x509-client-key

8.9 /ISAPI/Streaming

/ISAPI/Streaming	Service	v2.0
Notes:		

8.9.1/ISAPI/Streaming/status

/ISAPI/Streaming/status	General Resource	v2.0
GET		
Description	It is used to get a device streaming status.	
Query	None	
Inbound Data	None	
Success Return	**StreamingStatus**	
Notes: This command accesses the status of all device streaming sessions.		

StreamingStatus XML Block

```
<StreamingStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <totalStreamingSessions>        <!-- req, xs:integer -->  </totalStreamingSessions>
  <StreamingSessionStatusList/>     <!-- dep, only if there are sessions -->
</StreamingStatus>
```

8.9.2/ISAPI/Streaming/channels

/ISAPI/Streaming/channels	General Resource	v2.0
GET		
Description	It is used to get the properties of streaming channels for the device.	
Query	None	
Inbound Data	None	
Success Return	**StreamingChannelList**	
PUT		

Description	It is used to update the properties of streaming channels for the device.
Query	None
Inbound Data	**StreamingChannelList**
Success Return	**ResponseStatus**
POST	
Description	It is used to add a streaming channel for the device.
Query	None
Inbound Data	**StreamingChannel**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the list of streaming channels for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Streaming channels may be hardwired, or it may be possible to create multiple streaming channels per input if the device supports it. To determine whether it is possible to dynamically create streaming channels, check the defined HTTP methods in /ISAPI/Streaming/channels/description.	

StreamingChannelList XML Block

```
<StreamingChannelList version="2.0"
  xmlns="http://www.isapi.org/ver20/XMLSchema">
    <StreamingChannel/>    <!-- opt -->
</StreamingChannelList>
```

8.9.3/ISAPI/Streaming/channels/<ID>

/ISAPI/Streaming/channels/*ID*	**General Resource v2.0**
GET	
Description	It is used to get the properties of a particular streaming channel for the device.
Query	None
Inbound Data	None
Success Return	**StreamingChannel**
PUT	
Description	It is used to update the properties of a particular streaming channel for the device.

Query	None
Inbound Data	**StreamingChannel**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular streaming channel for the device.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

To support multi video input devices , the streaming ID in URL should be indicate video input channel number , so it is defined as : straming-Id + video-input-Id *100, for example : /Streaming/channels/101 indicates the first streaming from the first video input /Streaming/channels/202 indicates the second streaming from the second video input

For IPC, becourse of only one video input, case is simeple, it can accecpt 1 as the main stream id , 2 as the sub-stream.

<ControlProtocolList> identifies the control protocols that are valid for this type of streaming.
<Unicast> is for direct unicast streaming.
<Multicast> is for direct multicast streaming.
<videoSourcePortNo> and <audioSourcePortNo> are the source port numbers for the outbound video or audio streams.
<videoInputChannelID> refers to /ISAPI/System/Video/inputs/channel/ID.
<audioInputChannelID> refers to /ISAPI/System/Audio/channels/ID. It must be configured as an input channel.
Use of IPv4 or IPv6 addresses depends on the value of the <ipVersion> field in /ISAPI/System/Network/interfaces/ID/ipAddress.
<Security> determines whether SRTP is used for stream encryption.
<audioResolution> is the resolution for the outbound audio stream in bits.

StreamingChannel XML Block

```
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <channelName> <!-- req, xs:string --> </channelName>
  <enabled> <!-- req, xs:boolean -->  </enabled>
  <Transport> <!-- req -->
    <maxPacketSize>  <!-- opt, xs:integer --> </maxPacketSize>
    <audioPacketLength> <!-- opt, xs:integer --> </audioPacketLength>
    <audioInboundPacketLength><!-- opt, xs:integer -->  </audioInboundPacketLength>
    <audioInboundPortNo>  <!-- opt, xs:integer --> </audioInboundPortNo>
    <videoSourcePortNo> <!-- opt, xs:integer --> </videoSourcePortNo>
```

```
    <audioSourcePortNo><!-- opt, xs:integer --> </audioSourcePortNo>
    <ControlProtocolList> <!-- req -->
      <ControlProtocol>
        <!-- req -->
        <streamingTransport>
          <!-- req, xs:string, “HTTP,RTSP,SHTTP” -->
        </streamingTransport>
      </ControlProtocol>
    </ControlProtocolList>
    <Unicast><!-- opt -->
      <enabled> <!-- req, xs:boolean -->      </enabled>
      <interfaceID>   <!-- opt, xs:string -->   </interfaceID>
      <rtpTransportType>
        <!-- opt, xs:string, “RTP/UDP,RTP/TCP” -->
      </rtpTransportType>
    </Unicast>
    <Multicast> <!-- opt -->
      <enabled> <!-- req, xs:boolean -->      </enabled>
      <userTriggerThreshold> <!-- opt, xs:integer --> </userTriggerThreshold>
      <destIPAddress>     <!-- dep, xs:string -->  </destIPAddress>
      <videoDestPortNo><!-- opt, xs:integer --></videoDestPortNo>
      <audioDestPortNo><!-- opt, xs:integer --></audioDestPortNo>
      <destIPv6Address><!-- dep, xs:string --></destIPv6Address>
      <ttl><!-- opt, xs:integer --></ttl>
    </Multicast>
    <Security>
      <!-- opt -->
      <enabled><!-- req, xs:boolean --></enabled>
    </Security>
  </Transport>
  <Video>
    <!-- opt -->
    <enabled><!-- req, xs:boolean --></enabled>
    <videoInputChannelID> <!-- req, xs:string;id --> </videoInputChannelID>
    <videoCodecType>
      <!-- req, xs:string, “MPEG4,MJPEG,3GP,H.264,HK.264,MPNG” -->
    </videoCodecType>
    <videoScanType>
      <!-- opt, xs:string, “progressive,interlaced” -->
    </videoScanType>
    <videoResolutionWidth>    <!-- req, xs:integer --> </videoResolutionWidth>
    <videoResolutionHeight>   <!-- req, xs:integer --> </videoResolutionHeight>
    <videoPositionX> <!-- opt, xs:integer --> </videoPositionX>
```

```
    <videoPositionY> <!-- opt, xs:integer --> </videoPositionY>
    <videoQualityControlType>
      <!-- opt, xs:string, “CBR,VBR” -->
    </videoQualityControlType>
    <constantBitRate> <!-- dep, xs:integer, in kbps -->      </constantBitRate>
    <fixedQuality>     <!-- opt, xs:integer, percentage, 0..100 -->     </fixedQuality>
    <vbrUpperCap>   <!-- dep, xs:integer, in kbps --> </vbrUpperCap>
    <vbrLowerCap>   <!-- dep, xs:integer, in kbps --> </vbrLowerCap>
    <maxFrameRate>     <!-- req, xs:integer, maximum frame rate x100 -->
    </maxFrameRate>
    <keyFrameInterval> <!-- opt, xs:integer, milliseconds -->    </keyFrameInterval>
    <rotationDegree> <!-- opt, xs:integer, degrees, 0..360 --></rotationDegree>
    <mirrorEnabled>  <!-- opt, xs:boolean -->      </mirrorEnabled>
    <snapShotImageType>
      <!-- opt, xs:string, “JPEG,GIF,PNG” -->
    </snapShotImageType>
    <Mpeg4Profile> <!--dep, xs:string, “SP,ASP”--> </Mpeg4Profile>
    <H264Profile>
      <!-- dep, xs:string, “Baseline,Main,High, Extended” -->
    </H264Profile>
    <GovLength> <!--opt, xs:integer --> </GovLength>
    <SVC>
     <enabled>   <!-- req, xs:boolean -->     </enabled>
    <SVC>
  </Video>
  <Audio>
    <!-- opt -->
    <enabled>    <!-- req, xs:boolean -->      </enabled>
    <audioInputChannelID>     <!-- req, xs:string;id -->      </audioInputChannelID>
    <audioCompressionType>
      <!-- req, xs:string,
       “  G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPC
M，MP2L2”
      -->
    </audioCompressionType>
    <audioInboundCompressionType>
      <!-- opt, xs:string,
      “G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM,
MP2L2”
      -->
    </audioInboundCompressionType>
    <audioBitRate>    <!-- opt, xs:integer, in kbps -->  </audioBitRate>
    <audioSamplingRate><!-- opt, xs:float, in kHz --></audioSamplingRate>
```

```
    <audioResolution>      <!-- opt, xs:integer, in bits -->     </audioResolution>
  </Audio>
  <enableCABAC>   <!-- opt, xs: boolean --> <enableCABAC>
  <subStreamRecStatus>   <!-- opt, xs: boolean --> </subStreamRecStatus>
</StreamingChannel>
```

Example: Getting Streaming Channel Properties

The following is an example of a GET on the streaming parameters of a particular channel that has been preconfigured by the IP media device. Depending on the device, some streaming channels may be already preconfigured or the device while other may require that channels be manually configured before use.

```
GET /ISAPI/Streaming/channels/444 HTTP/1.1
…
HTTP/1.1 200 OK
Content-Type: application/xml; charset="UTF-8"
Content-Length: ISAPI

<?xml version="1.0" encoding="UTF-8"?>
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>444</id>
  <channelName>Input 1 MPEG-4 ASP</channelName>
  <enabled>true</enabled>
  <Transport>
    <rtspPortNo>554</rtspPortNo>
    <maxPacketSize>1446</maxPacketSize>
    <ControlProtocolList>
    <ControlProtocol>
      <streamingTransport>RTSP</streamingTransport>
    </ControlProtocol>
    <ControlProtocol>
      <streamingTransport>HTTP</streamingTransport>
    </ControlProtocol>
  </Transport>
  <Video>
    <enabled>true</enabled>
    <videoInputChannelID>2</videoInputChannelID>
    <videoCodecType>MPEG4</videoCodecType>
    <videoScanType>progressive</videoScanType>
    <videoResolutionWidth> 640</videoResolutionWidth>
    <videoResolutionHeight>480</videoResolutionHeight>
    <videoPositionX>0</videoPositionX>
    <videoPositionY>0</videoPositionY>
```

```
      <videoQualityControlType>CBR</videoQualityControlType>
      <constantBitRate>2000</constantBitRate>
      <maxFrameRate>2500</maxFrameRate>
      <keyFrameInterval>1000</keyFrameInterval>
      <rotationDegree>0</rotationDegree>
      <mirrorEnabled>false</mirrorEnabled>
      <snapShotImageType>JPEG</snapShotImageType>
   </Video>
   <Audio>
      <enabled>false</enabled>
      <audioInputChannelID>2</audioInputChannelID>
      <audioCompressionType> G.726</audioCompressionType>
      <audioBitRate>24</audioBitRate>
      <audioSamplingRate>8</audioSamplingRate>
   </Audio>
</StreamingChannel>
```

Example: Getting Streaming Capabilities

```
GET /ISAPI/Streaming/channels/444/capabilities HTTP/1.1
…
HTTP/1.1 200 OK
Content-Type: application/xml; charset="UTF-8"
Content-Length: ISAPI

<?xml version="1.0" encoding="UTF-8"?>
<StreamingChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id opt="111,222,333,444">444</id>
   <channelName min="0" max="64">Input 1 MPEG-4 ASP</channelName>
   <enabled opt="true,false" def="true">true</enabled>
   <Transport>
      <rtspPortNo min="0" max="65535" def="554">554</rtspPortNo>
      <maxPacketSize min="0" max="1500">1446</maxPacketSize>
      <audioPacketLength min="0" max="5000"/>
      <audioInboundPacketLength min="0" max="5000"/>
      <audioInboundPortNo min="0" max="65535"/>
      <videoSourcePortNo min="0" max="65535"/>
      <audioSourcePortNo min="0" max="65535"/>
      <ControlProtocolList>
         <ControlProtocol>
            <streamingTransport opt="RTSP/RTP,HTTP">RTSP</streamingTransport>
         </ControlProtocol>
         <ControlProtocol>
```

```
            <streamingTransport opt="RTSP/RTP,HTTP">HTTP</streamingTransport>
          </ControlProtocol>
        </ControlProtocolList>
        <Unicast>
          <enabled opt="true,false" def="false"/>
          <rtpTransportType opt="RTP/UDP,RTP/TCP"/>
        </Unicast>
        <Multicast>
          <enabled opt="true,false" def="false"/>
          <userTriggerThreshold/>
          <videoDestPortNo min="0" max="65535"/>
          <audioDestPortNo min="0" max="65535"/>
          <destIPAddress min="8" max="16"/>
          <destIPv6Address min="15" max="39"/>
          <ttl min="0" max="127" def="1"/>
        </Multicast>
        <Security>
          <enabled opt="true,false" def="false"/>
        </Security>
      </Transport>
      <Video>
        <enabled opt="true,false">true</enabled>
        <videoInputChannelID opt="1,2,3,4">2</videoInputChannelID>
        <videoCodecType opt="MJPEG,MPEG4">MPEG4</videoCodecType>
        <videoScanType opt="interlaced,progressive">progressive</videoScanType>
        <videoResolutionWidth min="0" max="640">640</videoResolutionWidth>
        <videoResolutionHeight min="0" max="480">480</videoResolutionHeight>
        <videoPositionX min="0" max="640">0</videoPositionX>
        <videoPositionY min="0" max="480">0</videoPositionY>
        <videoQualityControlType opt="CBR,VBR">CBR</videoQualityControlType>
        <constantBitRate min="50" max="4000" dynamic="true">2000</constantBitRate>
        <maxFrameRate                                    opt="2500,1250,625,312,156,78"
dynamic="true">2500</maxFrameRate>
        <keyFrameInterval min="0", max="10000">1000</keyFrameInterval>
        <rotationDegree opt="0,90,180,270" def="0">0</rotationDegree>
        <mirrorEnabled opt="true,false" def="false">false</mirrorEnabled>
        <snapShotImageType opt="JPEG" def="JPEG">JPEG</snapShotImageType>
      </Video>
      <Audio>
        <enabled opt="true,false" def="false">false</enabled>
        <audioInputChannelID opt="1,2,3,4">2</audioInputChannelID>
        <audioCompressionType                                  opt="G.726,G.711ulaw"
def="G.726">G.726</audioCompressionType>
```

```
    <audioBitRate opt="16,24,32,40" def="32" dynamic="true">24</audioBitRate>
    <audioSamplingRate opt="8" dynamic="true">8</audioSamplingRate>
    <audioResolution opt="3,4,5,6" dynamic="true"/>
  </Audio>
</StreamingChannel>
```

8.9.4/ISAPI/Streaming/channels/<ID>/dynamicCap

/ISAPI/Streaming/channels/*ID*/dynamicCap	General Resource v2.0
GET	
Description	Get dynamic capabilities, different resolutions have different frame rates; different audio compression types have different audio bit rate.
Query	None
Inbound Data	None
Success Return	**DynamicCap**
Notes:	

DynamicCap XML Block

```
<DynamicCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ResolutionAvailableDscriptorList>
    <ResolutionAvailableDscriptor>
      <videoResolutionWidth><!-- req, xs:integer --></videoResolutionWidth>
      <videoResolutionHeight>    <!-- req, xs:integer --></videoResolutionHeight>
      <supportedFrameRate><!-- req, xs:string --></supportedFrameRate>
    </ResolutionAvailableDscriptor>
  </ResolutionAvailableDscriptorList>
  <AudioDscriptorList>
      <audioCompressionType>
        <!-- req, xs:string,
          " G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM，MP2L2"-->
      </audioCompressionType>
  </AudioDscriptorList>
</DynamicCap>
```

Example: Getting the Dynamic Capabilities

```
GET /ISAPI/Streaming/Channels/101/dynamicCap HTTP/1.1
…
HTTP/1.1 200 OK
Content-Type: application/xml; charset="UTF-8"
```

```
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8"?>
<DynamicCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ResolutionAvailableDscriptorList>
    <ResolutionAvailableDscriptor>
      <videoResolutionWidth>176</videoResolutionWidth>
      <videoResolutionHeight>144</videoResolutionHeight>
      <supportedFrameRate>2500,6,12,25,50,100,200,400,600,800,1000,1200,1500,16
00,1800,2000,2200</supportedFrameRate>
    </ResolutionAvailableDscriptor>
    <ResolutionAvailableDscriptor>
      <videoResolutionWidth>352</videoResolutionWidth>
      <videoResolutionHeight>288</videoResolutionHeight>
      <supportedFrameRate>2500,6,12,25,50,100,200,400,600,800,1000,1200,1500,16
00,1800,2000,2200</supportedFrameRate>
    </ResolutionAvailableDscriptor>
    <ResolutionAvailableDscriptor>
      <videoResolutionWidth>704</videoResolutionWidth>
      <videoResolutionHeight>576</videoResolutionHeight>
      <supportedFrameRate>2500,6,12,25,50,100,200,400,600,800,1000,1200,1500,16
00,1800,2000,2200</supportedFrameRate>
    </ResolutionAvailableDscriptor>
  </ResolutionAvailableDscriptorList>
  <AudioDscriptorList>
    <audioCompressionType
SupportedAudioBitRate="32,64,128">MP2L2</audioCompressionType>
  </AudioDscriptorList>
</DynamicCap>
```

8.9.5/ISAPI/Streaming/channels/<ID>/status

/ISAPI/Streaming/channels/*ID*/status	**General Resource v2.0**
GET	
Description	It is used to get the list of streaming sessions associated with a particular channel.
Query	None
Inbound Data	None
Success Return	**StreamingSessionStatusList**
Notes:	

StreamingSessionStatusList XML Block

```
<StreamingSessionStatusList version="2.0"
   xmlns="http://www.isapi.org/ver20/XMLSchema">
   <StreamingSessionStatus>
      <clientAddress>   <!-- req -->
         <ipAddress>      <!-- req, xs:string -->            </ipAddress>
      </clientAddress>
   </StreamingSessionStatus>
</StreamingSessionStatusList>
```

8.9.6/ISAPI/Streaming/channels/<ID>/picture

/ISAPI/Streaming/channels/*ID*/picture	**General Resource v2.0**
GET	
Description	It is used to get a snapshot of the current image.
Query	videoResolutionWidth videoResolutionHeight snapShotImageType
Inbound Data	None
Success Return	**Picture over HTTP**

Notes:

All devices must support <snapShotImageType> of "JPEG".
Only support the main stream channel snapshot.
To determine the format of the picture returned either the parameters in <Video> or the query string values are used, or, if the Accept: header field is present in the request and the server supports it, the picture is returned in that format.
For supported values, query /Streaming/channels/*ID*/picture/capabilities.
Examples:

GET /ISAPI/Streaming/channels/101/picture?snapShotImageType=JPEG
…

GET /ISAPI/Streaming/channels/101/picture
Accept: image/jpeg
…

8.9.7/ISAPI/Streaming/channels/<ID>/requestKeyFrame

/ISAPI/Streaming/channels/*ID*/requestKeyFrame	General Resource v2.0
PUT	**Operator**
Description	It is used to request that the device issue a key frame on a particular channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: The key frame that is issued should include everything necessary to initialize a video decoder, i.e. parameter sets for H.264 or VOS for MPEG-4.	

8.10 /ISAPI/Snapshot

/ISAPI/Snapshot	Service v2.0
Notes: snapshot service	

8.10.1 /ISAPI/Snapshot/channels

/ISAPI/Snapshot/channels	General Resource v2.0
GET	
Description	It is used to get the properties of snapshot channels for the device.
Query	None
Inbound Data	None
Success Return	**SnapshotChannelList**
PUT	
Description	It is used to update the properties of snapshot channels for the device.
Query	None
Inbound Data	**SnapshotChannelList**
Success Return	**ResponseStatus**
Notes:	

SnapshotChannelList XML Block

```
<SnapshotChannelList version="2.0"
```

```
  xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SnapshotChannel/>      <!-- opt -->
</SnapshotChannelList>
```

8.10.2 /ISAPI/Snapshot/channels/<ID>

/ISAPI/Snapshot/channel/ID	General Resource v2.0
GET	
Description	It is used to get the properties of a particular snapshot channel.
Query	None
Inbound Data	None
Success Return	**SnapshotChannel**
PUT	
Description	It is used to update the properties of a particular snapshot channel.
Query	None
Inbound Data	**SnapshotChannel**
Success Return	**ResponseStatus**
Notes:	

SnapshotChannel XML Block

```
<SnapshotChannel version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string;id --> </id>
  <videoInputChannelID> <!-- req, xs:string;id --> </videoInputChannelID>
  <timingCapture> <!-- opt -->
    <enabled> <!-- req, xs:boolean --> </enabled>
    <supportSchedule> <!-- opt, ro, xs:boolean> </supportSchedule>
    <compress>
      <pictureCodecType>
        <!-- req, xs:string, "JPEG,BMP,GIF,PNG" -->
      </pictureCodecType>
      <pictureWidth> <!-- req, xs:integer --> </pictureWidth>
      <pictureHeight>      <!-- req, xs:integer --> </pictureHeight>
      <quality> <!-- opt, xs:integer, percentage, 0..100 --> </quality>
      <captureInterval> <!-- opt, xs:integer, milliseconds --> </captureInterval>
    <compress>
  </timingCapture>
  <eventCapture> <!-- opt -->
    <enabled>       <!-- req, xs:boolean -->   </enabled>
    <supportSchedule> <!-- opt, ro, xs:boolean> </supportSchedule>
    <compress>
      <pictureCodecType>
```

```
            <!-- req, xs:string, “JPEG,BMP,GIF,PNG” -->
        </pictureCodecType>
        <pictureWidth> <!-- req, xs:integer --> </pictureWidth>
        <pictureHeight>      <!-- req, xs:integer --> </pictureHeight>
        <quality>   <!-- opt, xs:integer, percentage, 0..100 -->     </quality>
        <captureInterval>   <!-- opt, xs:integer, milliseconds -->  </captureInterval>
      <compress>
  </eventCapture>
 </SnapshotChannel>
```

8.11 /ISAPI/Event

/ISAPI/Event	**Service v2.0**
GET	**Viewer**
Description	It is used to get the configuration of the device event behavior, scheduling and notifications.
Query	None
Inbound Data	None
Success Return	**EventNotification**
PUT	**Operator**
Description	It is used to udpate the configuration of the device event behavior, scheduling and notifications.
Query	None
Inbound Data	**EventNotification**
Success Return	**ResponseStatus**
Notes: The event trigger list defines the set of device behaviors that trigger events. The event schedule defines when event notifications are active. The event notification methods define what types of notification (e-mail) are supported.	

EventNotification XML Block

```
<EventNotification version=“2.0” xmlns=“http://www.isapi.org/ver20/XMLSchema”>
   <EventTriggerList/>          <!-- opt -->
   <EventNotificationMethods/>   <!-- opt -->
</EventNotification>
```

8.11.1 /ISAPI/Event/capabilities

/ISAPI/Event/capabilities	**General Resource v2.0**

GET	
Description	It is used to get network capability.
Query	None
Inbound Data	None
Success Return	**< EventCap>**
Notes:	

EventCap XML Block

```
<EventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <isSupportHDFull> <!-- opt, xs:boolen --> </isSupportHDFull>
   <isSupportHDError> <!-- opt, xs:boolen --> </isSupportHDError>
   <isSupportNicBroken> <!-- opt, xs:boolen --> </isSupportNicBroken>
   <isSupportIpConflict> <!-- opt, xs:boolen --> </isSupportIpConflict>
   <isSupportIllAccess> <!-- opt, xs:boolen --> </isSupportIllAccess>
   <isSupportViException> <!-- opt, xs:boolen --> </isSupportViException>
   <isSupportViMismatch> <!-- opt, xs:boolen --> </isSupportViMismatch>
   <isSupportRecordException> <!-- opt, xs:boolen --> </isSupportRecordException>
   <isSupportRaidException> <!-- opt, xs:boolen --> </isSupportRaidException>
</EventCap>
```

8.11.2 /ISAPI/Event/triggers

/ISAPI/Event/triggers	General Resource v2.0
GET	
Description	It is used to get the list of event triggers.
Query	None
Inbound Data	None
Success Return	**EventTriggerList**
PUT	
Description	It is used to update the list of event triggers.
Query	None
Inbound Data	**EventTriggerList**
Success Return	**ResponseStatus**
POST	
Description	It is used to add an event trigger.
Query	None
Inbound Data	**EventTrigger**
Success Return	**ResponseStatus**

DELETE	
Description	It is used to delete the list of event triggers.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: Event triggering defines how the device reacts to particular events, such as video loss or motion detection.	

EventTriggerList XML Block

```
<EventTriggerList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <EventTrigger/>  <!-- opt -->
</EventTriggerList>
```

8.11.3 /ISAPI/Event/triggers/<ID>

/ISAPI/Event/triggers/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular event trigger configuration.
Query	None
Inbound Data	None
Success Return	**EventTrigger**
PUT	
Description	It is used to update a particular event trigger configuration.
Query	None
Inbound Data	**EventTrigger**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular event trigger.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: An event trigger determines how the device reacts when a particular event is detected. The following types are supported: IO: trigger when an input IO port changes state. VMD: trigger on video motion detection. Video loss: trigger when the input video signal cannot be detected.	

Disk failure: trigger when a disk fails.
Recording failure: trigger when recording fails: either there is a problem with the disk, or the storage volume is full, or the volume is corrupt.
Bad video: trigger when the input video is bad.
POS: trigger when a point-of-sale event is detected.
Analytics: trigger on a general analytics event. Currently analytics events apart from VMD, which has its own event trigger, are not supported. Fan failure: trigger when a fan fails.
Nicbroken: trigger when net interface is broken.
Resolution mismatch: trigger when video input port resolution is not matched up to compress resolution.

The ID in "**/Event/triggers/*ID***" is defined as following declaration:
If the event type is IO, the ID is IO-InputPortNumber.
Examples :
IO-1 :the first IO input port
If the event type is VMD, videoloss or tamperdetection, the ID style is VMD/videoloss/tamper-InputChannelID.
Examples:
If video input channel id is "video1", the id is as follows:
VMD-1: Video Motion Detection of video input channel "video1".
videoloss-1: Video Loss Detection of video input channel "video1".
tamper-1: Tamper Detection of video input channel "video1".

EventTrigger XML Block

```
<EventTrigger version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <eventType>
    <!-- req, xs:string,
      "IO,VMD,videoloss,raidfailure,recordingfailure,
      badvideo,POS,analytics,fanfailure,overheat, tamperdetection, diskfull, diskerror,
      nicbroken, ipconflict, illaccess, videomismatch, resolutionmismatch,
      radifailure,PIR, WLSensor"
    -->
  </eventType>
  <eventDescription><!-- opt, xs:string --></eventDescription>
  <inputIOPortID>  <!-- dep, xs:string; id -->  </inputIOPortID>
  <dynInputIOPortID> <!-- dep, xs:string; id --> </dynInputPortID>
  <videoInputChannelID>  <!-- dep, xs:string; id -->  </videoInputChannelID>
  <dynVideoInputChannelID> <!-- dep, xs:string; id --> </dynVideoInputChannelID>
  <intervalBetweenEvents>  <!-- opt, xs:integer, seconds --></intervalBetweenEvents>
  <WLSensorID> <!-- dep, xs:string; id --> </WLSensorID>
  <EventTriggerNotificationList/>  <!-- opt -->
</EventTrigger>
```

8.11.4 /ISAPI/Event/triggers/<ID>/notifications

/ISAPI/Event/triggers/*ID*/notifications	**General Resource v2.0**
GET	
Description	It is used to get the list of notification methods and behaviors for an event trigger.
Query	None
Inbound Data	None
Success Return	**EventTriggerNotificationList**
PUT	
Description	It is used to update the list of notification methods and behaviors for an event trigger.
Query	None
Inbound Data	**EventTriggerNotificationList**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the list of notification method and behavior for an event trigger.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

ptz: PTZ action

record：recording

monitorAlarm : monitor alarm

center:send alarm to center

LightAudioAlarm : light blink and sound the alarm

<outputIOPortID> or <dynOutputIOPortID> is only required if the <notificationMethod> is "IO".

<videoInputID> or <dynVideoInputID> is only required if the <notificationMethod> is "record"。

<ptzAction> is only required if the <notificationMethod> is "ptz";

EventTriggerNotificationList XML Block

```
<EventTriggerNotificationList version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <EventTriggerNotification/>  <!-- opt -->
</EventTriggerNotificationList>
```

EventTriggerNotification XML Block

```
<EventTriggerNotification>  <!-- opt -->
  <id>  <!-- req, xs:string;id -->  </id>
  <notificationMethod>
    <!-- req, xs:string, "email,IM,IO,syslog,HTTP,FTP,beep, ptz, record
,    monitorAlarm, center, LightAudioAlarm …" -->
  </notificationMethod>
  <notificationRecurrence>
    <!-- opt, xs:string, "beginning,beginningandend,recurring" -->
  </notificationRecurrence>
  <notificationInterval> <!-- dep, xs:integer, milliseconds --> </notificationInterval>
  <outputIOPortID>  <!-- dep, xs:string;id -->  </outputIOPortID>
  <dynOutputIOPortID>  <!-- dep, xs:string;id -->  </dynOutputIOPortID>
  <videoInputID>  <!-- dep, xs:string;id -->  </videoInputID>
  <dynVideoInputID>  <!-- dep, xs:string;id -->  </dynVideoInputID>
  <ptzAction> <!-- dep -->
    <ptzChannelID> <!--req, xs:string; id -->  </ptzChannelID>
    <actionName> <!-- req, xs:string, "preset, pattern, patrol" --> </actionName>
    <actionNum> <!-- dep, xs:integer> </actionNum>
  </ptzAction>
</EventTriggerNotification>
```

8.11.5 /ISAPI/Event/schedules

/ISAPI/Event/schedules	General Resource	v2.0
Notes:		

8.11.6 /ISAPI/Event/schedules/inputs

/ISAPI/Event/schedules/inputs	General Resource	v2.0
GET		
Description	It is used to get trigger schedule.	
Query	None	
Inbound Data	None	
Success Return	**InputScheduleList**	
PUT		
Description	It is used to update trigger schedule.	
Query	None	
Inbound Data	**InputScheduleList**	
Success Return	**ResponseStatus**	
Notes:		

InputScheduleList XML Block

```
< InputScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   < Schedule/>     <!-- opt -->
</InputScheduleList>
```

8.11.7 /ISAPI/Event/schedules/inputs/<ID>

/ISAPI/Event/schedules/inputs/ID	General Resource	v2.0
GET		
Description	It is used to get trigger schedule.	
Query	None	
Inbound Data	None	
Success Return	**Schedule**	
PUT		
Description	It is used to update trigger schedule.	
Query	None	
Inbound Data	**Schedule**	
Success Return	**ResponseStatus**	

Notes:

8.11.8 /ISAPI/Event/schedules/outputs

/ISAPI/Event/schedules/outputs	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**OutputScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**OutputScheduleList**
Success Return	**ResponseStatus**
Notes:	

OutputScheduleList XML Block

```
< OutputScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  < Schedule/>                <!-- opt -->
</OutputScheduleList>
```

8.11.9 /ISAPI/Event/schedules/outputs/<ID>

/ISAPI/Event/schedules/outputs/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

8.11.10 /ISAPI/Event/schedules/motionDetections

/ISAPI/Event/schedules/motionDetections	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**MotionDetectionScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**MotionDetectionScheduleList**
Success Return	**ResponseStatus**
Notes:	

MotionDetectionScheduleList XML Block

```
<MotionDetectionScheduleList
version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   < Schedule/>                <!-- opt -->
</MotionDetectionScheduleList>
```

8.11.11 /ISAPI/Event/schedules/motionDetections/<ID>

/ISAPI/Event/schedule/motionDetections/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

8.11.12 /ISAPI/Event/schedules/tamperDetections

/ISAPI/Event/schedules/tamperDetections	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**TamperDetectionScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**TamperDetectionScheduleList**
Success Return	**ResponseStatus**
Notes:	

TamperDetectionScheduleList XML Block

```
<TamperDetectionScheduleList                                    version="2.0"
      xmlns="http://www.isapi.org/ver20/XMLSchema">
   < Schedule/>                  <!-- opt -->
</ TamperDetectionScheduleList>
```

8.11.13 /ISAPI/Event/schedules/tamperDetections/<ID>

/ISAPI/Event/schedule/tamperDetections/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

8.11.14 /ISAPI/Event/schedules/videolosses

/ISAPI/Event/schedules/videolosses	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**videolossScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**videolossScheduleList**
Success Return	**ResponseStatus**
Notes:	

videolossScheduleList XML Block

```
<videolossScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Schedule/>                <!-- opt -->
</videolossScheduleList>
```

8.11.15 /ISAPI/Event/schedules/videolosses/<ID >

/ISAPI/Event/schedule/videolosses/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

Schedule XML Block

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string; id --> </id>
```

```
<inputIOPortID>          <!-- ro, dep, xs:string; id -->          </inputIOPortID>
<outputIOPortID>         <!--  ro, dep, xs:string; id -->          </inputIOPortID>
<videoInputChannelID><!--  ro, dep, xs:string; id --></videoInputChannelID>
<TimeBlockList> <!-- req -->
    <TimeBlock>
      <dayOfWeek>
        <!-- opt, xs:integer, ISO8601 weekday number, 1=Monday, … -->
      </dayOfWeek>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
    </TimeBlock>
</TimeBlockList>
<HolidayBlockList> <!-- opt -->
    <TimeBlock>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
    </TimeBlock>
</HolidayBlockList>
</Schedule>
```

8.11.16 /ISAPI/Event/schedules/PIR

/ISAPI/Event/schedule/PIRs/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

Schedule XML Block

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string; id --> </id>
  <inputIOPortID>          <!-- ro, dep, xs:string; id -->            </inputIOPortID>
  <outputIOPortID>          <!--   ro, dep, xs:string; id -->            </inputIOPortID>
 <videoInputChannelID><!--   ro, dep, xs:string; id --></videoInputChannelID>
 <TimeBlockList> <!-- req -->
    <TimeBlock>
      <dayOfWeek>
        <!-- opt, xs:integer, ISO8601 weekday number, 1=Monday, … -->
      </dayOfWeek>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
   </TimeBlock>
 </TimeBlockList>
 <HolidayBlockList> <!-- opt -->
    <TimeBlock>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
   </TimeBlock>
 </HolidayBlockList>
</Schedule>
```

8.11.17 /ISAPI/Event/schedules/fieldDetections

/ISAPI/Event/schedules/fieldDetections	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**FieldDetectionScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**FieldDetectionScheduleList**
Success Return	**ResponseStatus**
Notes:	

FieldDetectionScheduleList XML Block

```
<FieldDetectionScheduleList                                    version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Schedule/>                  <!-- opt -->
</FieldDetectionScheduleList>
```

8.11.18 /ISAPI/Event/schedules/fieldDetections/<ID>

/ISAPI/Event/schedule/fieldDetections/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

Schedule XML Block

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string; id --> </id>
  <inputIOPortID>          <!-- ro, dep, xs:string; id -->            </inputIOPortID>
  <outputIOPortID>          <!--   ro, dep, xs:string; id -->            </inputIOPortID>
 <videoInputChannelID><!--   ro, dep, xs:string; id --></videoInputChannelID>
<TimeBlockList> <!-- req -->
    <TimeBlock>
      <dayOfWeek>
        <!-- opt, xs:integer, ISO8601 weekday number, 1=Monday, … -->
      </dayOfWeek>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
   </TimeBlock>
</TimeBlockList>
```

```
 <HolidayBlockList> <!-- opt -->
     <TimeBlock>
       <TimeRange>          <!-- req -->
         <beginTime>       <!-- req, xs:time, ISO8601 time -->    </beginTime>
         <endTime>       <!-- req, xs:time, ISO8601 time -->    </endTime>
       </TimeRange>
    </TimeBlock>
 </HolidayBlockList>
</Schedule>
```

8.11.19 /ISAPI/Event/schedules/sceneChangeDetections

/ISAPI/Event/schedules/sceneChangeDetections	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**SceneChangeDetectionScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**SceneChangeDetectionScheduleList**
Success Return	**ResponseStatus**
Notes:	

SceneChangeDetectionScheduleList XML Block

```
<SceneChangeDetectionScheduleList                                    version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
   <Schedule/>                    <!-- opt -->
</SceneChangeDetectionScheduleList>
```

8.11.20 /ISAPI/Event/schedules/sceneChangeDetections/<ID>

/ISAPI/Event/schedule/sceneChangeDetections/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.

Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

Schedule XML Block

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string; id --> </id>
  <inputIOPortID>  <!-- ro, dep, xs:string; id -->  </inputIOPortID>
  <outputIOPortID>  <!--  ro, dep, xs:string; id -->  </inputIOPortID>
 <videoInputChannelID><!--  ro, dep, xs:string; id --></videoInputChannelID>
 <TimeBlockList> <!-- req -->
    <TimeBlock>
      <dayOfWeek>
        <!-- opt, xs:integer, ISO8601 weekday number, 1=Monday, … -->
      </dayOfWeek>
      <TimeRange>  <!-- req -->
        <beginTime>  <!-- req, xs:time, ISO8601 time -->  </beginTime>
        <endTime>  <!-- req, xs:time, ISO8601 time -->  </endTime>
      </TimeRange>
   </TimeBlock>
 </TimeBlockList>
 <HolidayBlockList> <!-- opt -->
    <TimeBlock>
      <TimeRange>  <!-- req -->
        <beginTime>  <!-- req, xs:time, ISO8601 time -->  </beginTime>
        <endTime>  <!-- req, xs:time, ISO8601 time -->  </endTime>
      </TimeRange>
   </TimeBlock>
 </HolidayBlockList>
</Schedule>
```

8.11.21 /ISAPI/Event/schedules/audioDetections

/ISAPI/Event/schedules/audioDetections	**General Resource v2.0**

GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**AudioDetectionScheduleList**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**AudioDetectionScheduleList**
Success Return	**ResponseStatus**
Notes:	

AudioDetectionScheduleList XML Block

```
<AudioDetectionScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Schedule/>  <!-- opt -->
</AudioDetectionScheduleList>
```

8.11.22 /ISAPI/Event/schedules/audioDetections/<ID>

/ISAPI/Event/schedule/audioDetections/ID	General Resource v2.0
GET	
Description	It is used to get trigger schedule.
Query	None
Inbound Data	None
Success Return	**Schedule**
PUT	
Description	It is used to update trigger schedule.
Query	None
Inbound Data	**Schedule**
Success Return	**ResponseStatus**
Notes:	

Schedule XML Block

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string; id --> </id>
  <inputIOPortID>  <!-- ro, dep, xs:string; id -->  </inputIOPortID>
  <outputIOPortID>  <!--  ro, dep, xs:string; id -->  </inputIOPortID>
```

```
 <videoInputChannelID><!--   ro, dep, xs:string; id --></videoInputChannelID>
<TimeBlockList> <!-- req -->
    <TimeBlock>
      <dayOfWeek>
        <!-- opt, xs:integer, ISO8601 weekday number, 1=Monday, … -->
      </dayOfWeek>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
   </TimeBlock>
</TimeBlockList>
<HolidayBlockList> <!-- opt -->
    <TimeBlock>
      <TimeRange>          <!-- req -->
        <beginTime>      <!-- req, xs:time, ISO8601 time -->   </beginTime>
        <endTime>      <!-- req, xs:time, ISO8601 time -->   </endTime>
      </TimeRange>
   </TimeBlock>
</HolidayBlockList>
</Schedule>
```

8.11.23 /ISAPI/Event/notification

/ISAPI/Event/notification	General Resource v2.0
GET	
Description	It is used to get the configuration of notifications.
Query	None
Inbound Data	None
Success Return	**EventNotificationMethods**
PUT	
Description	It is used to set the configuration of notifications.
Query	None
Inbound Data	**EventNotificationMethods**
Success Return	**ResponseStatus**
Notes: The following notification types are supported: HTTP: the device connects to a given address and port and issues an HTTP GET/POST with the given parameters. FTP: a video clip or snapshot is uploaded to an FTP server.	

E-mail: a mail with the video clip or snapshot is sent in an e-mail to a list of servers. <MediaFormat> determines the type of snapshot, video clip and the video clip pre and post recording times.

EventNotificationMethods XML Block

```
<EventNotificationMethods version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <MailingNotificationList/><!-- opt -->
  <FTPNotificationList/><!-- opt -->
  <HttpHostNotificationList/><!-- opt -->
  <FTPFormat><!-- opt -->
    <uploadSnapShotEnabled><!-- req, xs:boolean --></uploadSnapShotEnabled>
    <uploadVideoClipEnabled><!-- req, xs:boolean --></uploadVideoClipEnabled>
  </FTPFormat>
  <EmailFormat><!-- opt -->
    <senderEmailAddress><!-- req, xs:string --></senderEmailAddress>
    <receiverEmailAddress><!-- req, xs:string --></receiverEmailAddress>
    <subject><!-- req, xs:string --></subject>
    <BodySetting><!-- opt -->
      <attachedVideoURLEnabled> <!-- req, xs:boolean --> </attachedVideoURLEnabled>
      <attachedSnapShotEnabled> <!-- req, xs:boolean --> </attachedSnapShotEnabled>
      <attachedVideoClipEnabled><!-- req, xs:boolean --> </attachedVideoClipEnabled>
    </BodySetting>
  </EmailFormat>
  <MediaFormat> <!-- opt -->
    <snapShotImageType> <!-- opt, xs:string, "JPEG,GIF,PNG" --> </snapShotImageType>
    <videoClipFormatType> <!-- opt, xs:string, "ASF,MP4,3GP,264" --></videoClipFormatType>
    <preCaptureLength> <!-- opt, xs:integer, milliseconds --> </preCaptureLength>
    <postCaptureLength> <!-- opt, xs:integer, milliseconds --> </postCaptureLength>
  </MediaFormat>
<EventNotificationMethods>
```

8.11.24 /ISAPI/Event/notification/httpHosts

/ISAPI/Event/notification/httpHosts		General Resource	v2.0
GET			
Description	It is used to get the configuration of e-mail.		

Query	None
Inbound Data	None
Success Return	**HttpHostNotificationList**
PUT	
Description	It is used to set the configuration of e-mail.
Query	None
Inbound Data	**HttpHostNotificationList**
Success Return	**ResponseStatus**
Notes:	

HttpHostNotificationList XML Block

```
<HttpHostNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <HttpHostNotification/>  <!-- opt -->
</HttpHostNotificationList>
```

8.11.25 /ISAPI/Event/notification/httpHosts/<ID>

/ISAPI/Event/notification/httpHosts/ID	**General Resource v2.0**
GET	
Description	It is used to get the configuration of a particular e-mail.
Query	None
Inbound Data	None
Success Return	**HttpHostNotification**
PUT	
Description	It is used to set the configuration of a particular e-mail.
Query	None
Inbound Data	**HttpHostNotification**
Success Return	**ResponseStatus**
Notes:	

HttpHostNotification XML Block

```
<HttpHostNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <url>  <!-- req, xs:string -->  </url>
  <protocolType><!-- req, xs:string, "HTTP,HTTPS" -->  </protocolType>
  <parameterFormatType>
  <!-- req, xs:string, "XML,querystring" -->
  </parameterFormatType>
  <addressingFormatType>
      <!-- req, xs:string, "ipaddress,hostname" -->
  </addressingFormatType>
```

```
<hostName>    <!-- dep, xs:string -->  </hostName>
<ipAddress>    <!-- dep, xs:string -->  </ipAddress>
<ipv6Address> <!-- dep, xs:string -->  </ipv6Address>
<portNo>   <!-- opt, xs:integer --> </portNo>
<userName>    <!-- dep, xs:string -->  </userName>
<password>    <!-- dep, xs:string -->  </password>
<httpAuthenticationMethod>
<!-- req, xs:string, “MD5digest,none” -->
</httpAuthenticationMethod>
</HttpHostNotification>
```

8.11.26 /ISAPI/Event/notification/streaming

/ISAPI/Event/notification/streaming	General Resource v2.0
GET	
Description	It is used to get the list of recording notifications.
Query	None
Inbound Data	None
Success Return	**StreamingNotificationList**
PUT	
Description	It is used to update the list of E-mail notifications.
Query	None
Inbound Data	**StreamingNotificationList**
Success Return	**ResponseStatus**
POST	
Description	It is used to add an E-mail notification.
Query	None
Inbound Data	**StreamingNotification**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete the list of E-mail notifications.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: When an event occurs, modifying the compression parameters of a video stream	

StreamingNotificationList XML Block

```
<StreamingNotificationList version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <StreamingNotification/>      <!-- opt -->
</StreamingNotificationList>
```

8.11.27 /ISAPI/Event/notification/streaming/<ID>

/ISAPI/Event/notification/Streaming/*ID*	**General Resource v2.0**
GET	
Description	It is used to get a particular E-mail notification configuration.
Query	None
Inbound Data	None
Success Return	**StreamingNotification**
PUT	
Description	It is used to update a particular E-mail notification configuration.
Query	None
Inbound Data	**StreamingNotification**
Success Return	**ResponseStatus**
DELETE	
Description	It is used to delete a particular E-mail notification.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

StreamingNotification XML Block

```
<StreamingNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>   <!-- req, xs:string;id -->      </id> //101   201 301
  <streamingId> <!-- req, xs:string;id --> <streamingId>
  <Video>
    <!-- opt -->
    <enabled><!-- req, xs:boolean --></enabled>
    <videoInputChannelID> <!-- req, xs:string;id --> </videoInputChannelID>
    <videoCodecType>
      <!-- req, xs:string, "MPEG4,MJPEG,3GP,H.264,MPNG" -->
    </videoCodecType>
    <videoScanType>
      <!-- opt, xs:string, "progressive,interlaced" -->
    </videoScanType>
    <videoResolutionWidth>    <!-- req, xs:integer --> </videoResolutionWidth>
    <videoResolutionHeight>   <!-- req, xs:integer --> </videoResolutionHeight>
```

```
      <videoPositionX> <!-- opt, xs:integer --> </videoPositionX>
      <videoPositionY> <!-- opt, xs:integer --> </videoPositionY>
      <videoQualityControlType>
        <!-- opt, xs:string, “CBR,VBR” -->
      </videoQualityControlType>
      <constantBitRate> <!-- dep, xs:integer, in kbps -->      </constantBitRate>
      <fixedQuality>     <!-- opt, xs:integer, percentage, 0..100 -->     </fixedQuality>
      <vbrUpperCap>   <!-- dep, xs:integer, in kbps --> </vbrUpperCap>
      <vbrLowerCap>   <!-- dep, xs:integer, in kbps --> </vbrLowerCap>
      <maxFrameRate>     <!-- req, xs:integer, maximum frame rate x100 -->
      </maxFrameRate>
      <keyFrameInterval> <!-- opt, xs:integer, milliseconds -->    </keyFrameInterval>
      <rotationDegree> <!-- opt, xs:integer, degrees, 0..360 --></rotationDegree>
      <mirrorEnabled>  <!-- opt, xs:boolean -->      </mirrorEnabled>
      <snapShotImageType>
        <!-- opt, xs:string, “JPEG,GIF,PNG” -->
      </snapShotImageType>
      <Mpeg4Profile> <!--dep, xs:string, “SP,ASP”--> </Mpeg4Profile>
      <H264Profile>
        <!-- dep, xs:string, “Baseline,Main,High, Extended” -->
      </H264Profile>
      <GovLength> <!--opt, xs:integer --> </GovLength>
    </Video>
    <Audio>
      <!-- opt -->
      <enabled>    <!-- req, xs:boolean -->      </enabled>
      <audioInputChannelID>     <!-- req, xs:string;id -->      </audioInputChannelID>
      <audioCompressionType>
        <!-- req, xs:string,
          “  G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPC
M”
        -->
      </audioCompressionType>
      <audioInboundCompressionType>
        <!-- opt, xs:string,
        “ G.711alaw,G.711ulaw,G.726,G.729,G.729a,G.729b,PCM,MP3,AC3,AAC,ADPCM
”
        -->
      </audioInboundCompressionType>
      <audioBitRate>    <!-- opt, xs:integer, in kbps -->  </audioBitRate>
      <audioSamplingRate><!-- opt, xs:float, in kHz --></audioSamplingRate>
      <audioResolution>      <!-- opt, xs:integer, in bits -->     </audioResolution>
    </Audio>
```

```
</StreamingNotification>
```

8.11.28 /ISAPI/Event/notification/alarmCenter

URI	/ISAPI/Event/notification/alarmCenter		Type	Resource
Function	Access the list of alarm center notification hosts.			
Methods	Query String(s)	Inbound Data	Return Result	
GET			<alarmCenterNotificationList	
PUT		<alarmCenterNotificationList>	<ResponseStatus>	
POST		<alarmCenterNotification>	<ResponseStatus>	
DELETE			<ResponseStatus>	
Notes	Alarm center notification involves the device connecting to a particular alarm center delivering an privacy envent message whenever the event triggers.			

alarmCenterNotificationList XML Block

```
<alarmCenterNotificationList version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <alarmCenterNotification/>    <!-- opt -->
</alarmCenterNotificationList>
```

8.11.29 /ISAPI/Event/notification/alarmCenter/<ID>

URI	/ISAPI/Event/notification/alarmCenter/ID		Type	Resource
Function	Access a particular HTTP notification host.			
Methods	Query String(s)	Inbound Data	Return Result	
GET			<alarmCenterNotification>	
PUT		<alarmCenterNotification>	<ResponseStatus>	
DELETE			<ResponseStatus>	
Notes	Depending on the value of <addressingFormatType>, either the <hostName> or the IP address fields will be used to locate the alarm center			

alarmCenterNotification XML Block

```
<alarmCenterNotification version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>                    <!-- req, xs:string;id -->                    </id>
  <addressingFormatType>
    <!-- req, xs:string, "ipaddress,hostname" -->
  </addressingFormatType>
  <hostName>              <!-- dep, xs:string -->                       </hostName>
  <ipAddress>             <!-- dep, xs:string -->                       </ipAddress>
  <ipv6Address>           <!-- dep, xs:string -->                       </ipv6Address>
  <portNo>                <!-- req, xs:integer -->                      </portNo>
</alarmCenterNotification>
```

8.11.30 /ISAPI/Event/notification/alertStream

/ISAPI/Event/notification/alertStream	**General Resource v2.0**
GET	**Viewer**
Description	It is used to get the event notification data stream through HTTP server push.
Query	None
Inbound Data	None
Success Return	**Stream of <EventNotificationAlert>**

Notes:

This function is used to get an event notification alert stream from the media device via HTTP or HTTPS. This function does not require that a client/VMS system be added as an HTTP(S) destination on the media device. Instead, the client/VMS system can call this API to initialize a stream of event information from the device. In other words, a connection is established with the device when this function is called, and stays open to constantly receive event notifications.

This API uses HTTP server-push with the MIME type multipart/mixed defined in RFC 2046.

<protocol> is the protocol name, i.e. "HTTP" or "HTTPS".

<channelID> is present for video and analytics events.

<activePostCount> is the sequence number of current notification for this particular event. It starts at 1. Useful for recurring notifications of an event. Each event maintains a separate post count.

EventNotificationAlert XML Block

```
<EventNotificationAlert version="2.0"
 xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress>        <!-- dep, xs:string -->      </ipAddress>
```

```
  <ipv6Address><!-- dep, xs:string --></ipv6Address>
  <portNo>          <!-- opt, xs:integer -->       </portNo>
  <protocol>        <!-- opt, xs:string -->        </protocol>
  <macAddress>      <!-- opt, xs:string;MAC -->    </macAddress>
  <channelID>       <!-- dep, xs:string -->        </channelID>
  <dateTime>        <!-- req, xs:datetime -->      </dateTime>
  <activePostCount>  <!-- req, xs:integer -->      </activePostCount>
  <eventType>   <!-- req, xs:string, “IO,VMD,videoloss, shelteralarm” -->   </eventType>
  <eventState>      <!-- req, xs:string, “active,inactive” -->   </eventState>
  <eventDescription>   <!-- req, xs:string -->                  </eventDescription>
  <inputIOPortID>   <!-- dep, xs:integer, if <eventType> is “IO” -->       </inputIOPortID>
  <DetectionRegionList>          <!-- dep, if <eventType> is “VMD” -->
    <DetectionRegionEntry>       <!-- req -->
      <regionID>           <!-- req, xs:string -->          </regionID>
      <sensitivityLevel>       <!-- req, xs:integer, 0..100 -->    </sensitivityLevel>
    </DetectionRegionEntry>
  </DetectionRegionList>
</EventNotificationAlert>
```

Example

The following is an example of an HTTP event stream that pushes a VMD event from video channel 1.

```
GET /Event/notification/alertStream HTTP/1.1
…
HTTP/1.1 200 OK
MIME-Version: 1.0
Content-Type:    multipart/mixed; boundary=“<boundary>”
--<boundary>
Content-Type: application/xml; charset=“UTF-8”
Content-Length: ISAPI

<?xml version=“1.0” encoding=“UTF-8”?>
<EventNotificationAlert version=“2.0”
 xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <ipAddress>172.6.64.7</ipAddress>
  <portNo>80</portNo>
  <protocol>HTTP</protocol>
  <macAddress>01:17:24:45:D9:F4</macAddress>
  <channelID>1</channelID>
  <dateTime>2009-11-14T15:27Z</dateTime>
  <activePostCount>1</activePostCount>
  <eventType>VMD</eventType>
```

```
  <eventState>active</eventState>
  <eventDescription>Motion alarm</eventDescription>
  <DetectionRegionList>
    <DetectionRegionEntry>
      <regionID>2</regionID>
      <sensitivityLevel>4</sensitivityLevel>
    </DetectionRegionEntry>
  </DetectionRegionList>
</EventNotificationAlert>
--<boundary>
…
```

8.11.31 HTTP Notification Alert

http://<ipAddress>:<portNo>/<url>	
POST	
Description	Send alert info to alarm center by HTTP POST method.
Query	None
Inbound Data	None
Success Return	**Notification Alert**

Notes:

Either GET or POST can be used. If GET is used, the corresponding query string parameters are provided in place of the inbound XML. If Post is used, the inbound XML is provided in place of the corresponding query string parameters.

The “DeviceID=” and “DeviceName=” fields are taken from the <DeviceInfo> settings for the device.

The <parameterFormatType> tag indicates whether XML or query string parameters should be used for this API.

The <protocolType> tag under <HttpHostList> determines whether HTTP or HTTPS is used for this API.

The <portNo> tag under <HttpHostList> determines the port number to be used for the notification alert.

The <portNo> and <protocolType> tags in the alert are provided for a client application to connect/manage the device after it sends out this notification.

The <addressingFormatType> tag under <HttpHostList> determines whether <ipAddress>/IPAddress or <ipv6Address>/IPv6Address is used.

The <url> tag under <HttpHostList> indicates the URL base to be used for the alert.

If <eventType>/EventType refers to an input-port-related event, the <inputIOPortID> tag or InputIOPortID parameter must be provided.

If <eventType>/EventType refers to a motion-related event, the <DetectionRegionList> block

or RegionIndexX parameter(s) must be provided if detection regions have been defined. If the motion event is for a full-screen configuration, these region indexes should not be provided.
The <sensitivityLevel>/SensitivityLevelX and <detectionThreshold>/DetectionThresholdX parameters are used to indicate the current values of the activity detection at the time that the notification is sent out.
If the alert is for a motion-related event, multiple region indexes may be provided per single API. If query string parameters are used, the format "RegionIndexX" is used where "X" is a number starting with "1" and incrementing by one for every subsequent region index provided.
If the <httpAuthenticationMethod> tag under <HttpHostList> is configured for "MD5 Digest Authentication", the corresponding security values must be stored in the header fields of the HTTP(S) request.
The <activePostCount>/ActivePostCount parameter is a sequence number starting at 1 and incrementing by one for every event notification sent.

Notification Alert

```
version=1.0
DeviceID=
DeviceName=
IPAddress=
IPv6Address=
PortNo=
Protocol=
MacAddress=
version=1.0
DeviceID=
DeviceName=
IPAddress=
IPv6Address=
PortNo=
Protocol=
MacAddress=
ChannelID=
DateTime=
ActivePostCount=
EventType=
EventState=
EventDescription=
InputIOPortID=
RegionIndex1=
SensitivityLevel1=
DetectionThreshold1=
RegionIndex2=
SensitivityLevel2=
```

```
DetectionThreshold2=
…
```

8.11.32 Event Triggering Examples

Example: Trigger Events on IO Port

The command below enables detection for input port 1. When the input signal is detected according to <inputIOPortID>, two event notification responses are used – output port 1 will be triggered for the duration of the input signal detection, and an SMTP server will be notified with the “E-mail Event Notification Alert”. The behavior of this notification is as follows:

- A SMTP notification is sent at detection time, and every some seconds after while the signal is present. This is denoted by the <notificationRecurrence> tags. These APIs will have an <eventState> of “active”.
- When the input port 1 signal detection stops, one last E-mail notification is sent to the server with an <eventState> of “active”.
- After the signal detection stops for input port 1, the device will wait some seconds before starting to detect the signal again for this port.

```
PUT /ISAPI/Event/triggers/IO-1 HTTP/1.1
Content-Type: application/xml; charset=“UTF-8”
Content-Length: xxx

<?xml version=“1.0” encoding=“UTF-8”?>
<EventTrigger version=“2.0”  xmlns=“http://www.isapi.org/ver20/XMLSchema”>
  <id>IO-1</id> <!-- “eventType: IO” -->
  <EventTriggerNotificationList>
    <EventTriggerNotification>
    <id>1</id>
    <notificationMethod>email</notificationMethod>
    </EventTriggerNotification>
    <EventTriggerNotification>
      <id>2</id>
      <notificationMethod> IO</notificationMethod>
      <outputIOPortID>1</outputIOPortID>
    </EventTriggerNotification>
  </EventTriggerNotificationList>
</EventTrigger>
```

Example: Schedule event detection and triggering

The command below schedules event detection and triggering from 7:00 am to 5:00 pm. every Tuesday.

```
PUT /ISAPI/Event/schedule/IO-IN-1 HTTP/1.1
Content-Type: application/xml; charset="UTF-8"
Content-Length: xxx

<?xml version="1.0" encoding="UTF-8"?>
<EventSchedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>IO-IN-1</id>
  <eventType>IO</eventType>
  <inputIOPortID>1</inputIOPortID>
  <TimeBlockList>
    <TimeBlock>
      <dayOfWeek>2</dayOfWeek>
      <TimeRange>
        <beginTime>07:00:00</beginTime>
        <endTime>17:00:00</endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</EventSchedule>
```

8.12 /ISAPI/Smart

/ISAPI/Smart	Service	v2.0
Notes: Smart service		

8.12.1 /ISAPI/Smart/capabilities

/ISAPI/Smart/capabilities	General Resource	v2.0
GET		
Description	It is used to get Smart capability.	
Query	None	
Inbound Data	None	
Success Return	< SmartCap>	
Notes:		

SmartCap XML Block

```
<SmartCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportROI> <!-- opt, xs:boolen --> </isSupportROI>
  <isSupportFaceDetect> <!-- opt, xs:boolen --> </isSupportFaceDetect>
```

```
  <isSupportIntelliTrace> <!-- opt, xs:boolen --> </isSupportIntelliTrace>
  <isSupportFieldDetection> <!-- opt, xs:boolen --> </isSupportFieldDetection>
  <isSupportDefocusDetection> <!-- opt, xs:boolen --> </isSupportDefocusDetection>
  <isSupportAudioDetection> <!-- opt, xs:boolen --> </isSupportAudioDetection>
  <isSupportSceneChangeDetection> <!-- opt, xs:boolen -->
  </isSupportSceneChangeDetection>
</SmartCap>
```

8.12.2 /ISAPI/Smart/ROI/channels

/ISAPI/Smart/ROI/channels	General Resource v2.0
GET	
Description	Access and configure the ROI.
Query	None
Inbound Data	None
Success Return	**ROIList**
PUT	
Description	Access and configure the ROI.
Query	None
Inbound Data	**ROIList**
Success Return	**ResponseStatus**
Notes:	

ROIList XML Block

```
<ROIList  version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ROI/> <!-- opt -->
</ROIList>
```

8.12.3 /ISAPI/Smart/ROI/channels/<ID>

/ISAPI/Smart/ROI/channels/ID	General Resource v2.0
GET	
Description	Access and configure the ROI for a special channel.
Query	None
Inbound Data	None
Success Return	**ROI**
PUT	

Description	Access and configure the ROI for a special channel.
Query	None
Inbound Data	**ROI**
Success Return	**ResponseStatus**
DELETE	
Description	Access and configure the ROI for a special channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

normalizedScreenSize: the size of normalized screen
ROIRegionList:the list of ROI region
<ID> should be consistent with <ID> of streaming.
<enabled/> <!-- req, xs:string --> if the value of this tag is “disable”, all of regions are invalid.

ROI XML Block

```
<ROI version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id/> <!-- req, xs:string, id -->
  <enabled/> <!-- req, xs:string -->
  <normalizedScreenSize>  <!--req-->
    <normalizedScreenWidth> <!-- req, xs:integer --></normalizedScreenWidth>
    <normalizedScreenHeight> <!-- req, xs:integer --></normalizedScreenHeight>
  </normalizedScreenSize>
  <ROIRegionList/> <!--dep-->
  <FaceTrace/> <!--dep-->
  <ObjectTrace/> <!--dep-->
</ROI>
```

8.12.4 /ISAPI/Smart/ROI/channels/<ID>/regions

/ISAPI/Smart/ROI/channels/ID/regions	**General Resource v2.0**
GET	
Description	Access and configure the ROI regions for a special channel.
Query	None
Inbound Data	None
Success Return	**ROIRegionsList**
PUT	

Description	Access and configure the ROI regions for a special channel
Query	None
Inbound Data	**ROIRegionsList**
Success Return	**ResponseStatus**
Notes:	

ROIRegionsList XML Block

```
<ROIRegionsList    version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema" size=>
  <ROIRegion/> <!-- opt -->
</ROIRegionsList>
```

8.12.5 /ISAPI/Smart/ROI/channels/<ID>/regions/<ID>

/ISAPI/Smart/ROI/channels/ID/regions/ID	**General Resource v2.0**
GET	
Description	Access and configure one ROI region for a special channel.
Query	None
Inbound Data	None
Success Return	**ROIRegion**
PUT	
Description	Access and configure one ROI region for a special channel
Query	None
Inbound Data	**ROIRegion**
Success Return	**ResponseStatus**
DELETE	
Description	Access and configure one ROI region for a special channel
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes: qualityLevel:quality level of a region RegionCoordinatesList:coordinate of ROI	

ROIRegion XML Block

```
<ROIRegion    version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <id> <!-- req, xs:integer--> </id>
```

```
    <enabled>      <!-- req, xs:boolean -->      </enabled>
    <name> <!-- opt, xs:string --> </name>
    <qualityLevelOfROI>    <!-- req, xs:integer "1-6"--> </qualityLevelOfROI>
    <RegionCoordinatesList/>
</ROIRegion>
```

8.12.6 /ISAPI/Smart/ROI/channels/<ID>/facetrace

/ISAPI/Smart/ROI/channels/ID/facetrace	General Resource v2.0
GET	
Description	Access and configure the ROI regions for a special channel.
Query	None
Inbound Data	None
Success Return	**FaceTrace**
PUT	
Description	Access and configure the ROI regions for a special channel
Query	None
Inbound Data	**FaceTrace**
Success Return	**ResponseStatus**
Notes:	

FaceTrace XML Block

```
<FaceTrace version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <enabled> <!-- req, xs:boolean -->      </enabled>
   <name> <!-- opt, xs:string --> </name>
   <qualityLevelOfROI>      <!-- req, xs:integer-->  </qualityLevelOfROI>
</FaceTrace>
```

8.12.7 /ISAPI/Smart/ROI/channels/<ID>/objecttrac e

/ISAPI/Smart/ROI/channels/ID/objecttrace	General Resource v2.0
GET	
Description	Access and configure the ROI regions for a special channel.
Query	None
Inbound Data	None
Success Return	**ObjectTrace**

PUT	
Description	Access and configure the ROI regions for a special channel
Query	None
Inbound Data	**ObjectTrace**
Success Return	**ResponseStatus**
Notes:	

ObjectTrace XML Block

```
<ObjectTrace version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean -->  </enabled>
  <qualityLevelOfROI>  <!-- req, xs:integer-->  </qualityLevelOfROI>
</ObjectTrace>
```

8.12.8 /ISAPI/Smart/FaceDetect/<ID>

/ISAPI/Smart/FaceDetect/ID	General Resource v2.0
GET	
Description	Access and configure the FaceDetect.
Query	None
Inbound Data	None
Success Return	**FaceDetect**
PUT	
Description	Access and configure the FaceDetect.
Query	None
Inbound Data	**FaceDetect**
Success Return	**ResponseStatus**
Notes: <ID> stands for channel number	

FaceDetect XML Block

```
<FaceDetect version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id/> <!-- req, xs:string, id -->
  <enabled>  <!-- req, xs:boolean -->  </enabled>
  <minObjectSize>
    <!-- opt, xs:integer, min number of pixels per object -->
  </minObjectSize>
  <maxObjectSize>
    <!-- opt, xs:integer, max number of pixels per object -->
  </maxObjectSize>
  <ROI> <!--opt-->
```

```
    <minHorizontalResolution> <!-- req, xs:integer -->    </minHorizontalResolution>
    <minVerticalResolution>    <!-- req, xs:integer -->    </minVerticalResolution>
  </ROI>
  <sensitivityLevel>            <!-- req -->
    <!-- req, xs:integer -->
  </sensitivityLevel>
  <detectionThreshold>           <!-- dep-->
    <!-- req, xs:integer-->
  </detectionThreshold>
  <highlightsenabled><!-- req, xs:boolean -->      </highlightsenabled>
</FaceDetect>
```

8.12.9 /ISAPI/Smart/IntelliTrace/<ID>

/ISAPI/Smart/IntelliTrace/<ID>	General Resource v2.0
GET	
Description	
Query	None
Inbound Data	None
Success Return	**IntelliTrace**
PUT	
Description	
Query	None
Inbound Data	**IntelliTrace**
Success Return	**ResponseStatus**
Notes:	

IntelliTrace XML Block

```
<IntelliTrace version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id> <!-- req, xs:string --> <id>
  <enabled>  <!-- req, xs:boolean -->      </enabled>
  <tracktime> <!-- opt, xs:integer, 0--300--> </tracktime>
</IntelliTrace>
```

8.12.10 /ISAPI/Smart/IntelliTrace/<ID>/ZoomRatial

/ISAPI/Smart/IntelliTrace/ID/ZoomRatial	General Resource v2.0

PUT	
Description	
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

8.12.11 /ISAPI/Smart/FieldDetection

/ISAPI/Smart/FieldDetection	General Resource v2.0
GET	
Description	Field detection configuration for all video input channels.
Query	None
Inbound Data	None
Success Return	**FieldDetectionList**
PUT	
Description	Field detection configuration for all video input channels.
Query	None
Inbound Data	**FieldDetectionList**
Success Return	**ResponseStatus**
Notes:	

FieldDetectionList XML Block

```
<FieldDetectionList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <FieldDetection/>    <!-- opt -->
</FieldDetectionList>
```

8.12.12 /ISAPI/Smart/FieldDetection/<ID>

/ISAPI/Smart/FieldDetection/ID	General Resource v2.0
GET	
Description	Field detection configuration for a video input channels.
Query	None
Inbound Data	None
Success Return	**FieldDetection**
PUT	
Description	Field detection configuration for a video input channels.
Query	None

Inbound Data	**FieldDetection**
Success Return	**ResponseStatus**
Notes:	

FieldDetection XML Block

```
<FieldDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
   <id>   <!-- req, xs:string -->   </id>
   <enabled> <!-- req, xs:boolean -->      </enabled>
<startTriggerTime>     <!-- req, xs:integer, milliseconds --> </startTriggerTime>
   <endTriggerTime>   <!-- req, xs:integer, milliseconds --> </endTriggerTime>
<normalizedScreenSize>
  <normalizedScreenWidth> <!-- req, xs:integer --> </normalizedScreenWidth>
  <normalizedScreenHeight> <!-- req, xs:integer --> </normalizedScreenHeight>
  </normalizedScreenSize>
  <minObjectSize>
    <!-- opt, xs:integer, min number of pixels per object -->
</minObjectSize>
    <maxObjectSize>
    <!-- opt, xs:integer, max number of pixels per object -->
  </maxObjectSize>
<FieldDetectionRegionList size="4"/>
</FieldDetection>
```

8.12.13 /ISAPI/Smart/FieldDetection/<ID>/regions

/ISAPI/Smart/FieldDetection/ID/regions	**General Resource v2.0**
GET	
Description	Access the list of regions for Field detection on a particular video input channel.
Query	None
Inbound Data	None
Success Return	**FieldDetectionRegionList**
PUT	
Description	Access the list of regions for Field detection on a particular video input channel.
Query	None
Inbound Data	**FieldDetectionRegionList**
Success Return	**ResponseStatus**
POST	

Description	Access the list of regions for Field detection on a particular video input channel.
Query	None
Inbound Data	None
Success Return	**FieldDetectionRegion**
DETELE	
Description	Access the list of regions for Field detection on a particular video input channel.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**
Notes:	

FieldDetectionRegionsList XML Block

```
<FieldDetectionRegionList version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <FieldDetectionRegion/>
</FieldDetectionRegionList>
```

8.12.14 /ISAPI/Smart/FieldDetection/<ID>/region/<ID>

/ISAPI/Smart/FieldDetection/ID/regions/ID	**General Resource v2.0**
GET	
Description	Access the list of regions for Field detection.
Query	None
Inbound Data	None
Success Return	**FieldDetectionRegion**
PUT	
Description	Access the list of regions for Field detection.
Query	None
Inbound Data	**FieldDetectionRegion**
Success Return	**ResponseStatus**
DELETE	
Description	Access the list of regions for Field detection.
Query	None
Inbound Data	None
Success Return	**ResponseStatus**

Notes:

FieldDetectionRegion XML Block

```
<FieldDetectionRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>                <!-- req, xs:string -->              </id>
   <enabled>      <!-- req, xs:boolean -->      </enabled>
  <sensitivityLevel>
    <!--req, xs:integer-->
  </sensitivityLevel>
  <timeThreshold>
    <!--req, xs:integer -->
  </timeThreshold>
  <objectOccupation>
     <!--req, xs:integer-->
  </objectOccupation>
  <RegionCoordinatesList>
    <RegionCoordinates>   <!-- req, -->
      <positionX>          <!-- req, xs:integer;coordinate -->       </positionX>
      <positionY>          <!-- req, xs:integer;coordinate    -->     </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</FieldDetectionRegion>
```

8.12.15 /ISAPI/Smart/DefocusDetection

/ISAPI/Smart/DefocusDetection	General Resource v2.0
GET	
Description	Defocus detection configuration for all audio input channels.
Query	None
Inbound Data	None
Success Return	**DefocusDetectionList**
PUT	
Description	Defocus detection configuration for all audio input channels.
Query	None
Inbound Data	**DefocusDetectionList**
Success Return	**ResponseStatus**
Notes:	

DefocusDetectionList XML Block

```
<DefocusDetectionList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DefocusDetection/>      <!-- opt -->
</DefocusDetectionList>
```

8.12.16 /ISAPI/Smart/DefocusDetection/<ID>

/ISAPI/Smart/ DefocusDetection/ID	General Resource v2.0
GET	
Description	Defocus detection configuration for a audio input channel.
Query	None
Inbound Data	None
Success Return	**DefocusDetection**
PUT	
Description	Defocus detection configuration for a audio input channel.
Query	None
Inbound Data	**DefocusDetection**
Success Return	**ResponseStatus**
Notes:	

DefocusDetection XML Block

```
<DefocusDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>   <!-- req, xs:string -->   </id>
  <enabled> <!-- req, xs:boolean -->      </enabled>
</DefocusDetection>
```

8.12.17 /ISAPI/Smart/AudioDetection/channels

/ISAPI/Smart/AudioDetection/channels	General Resource v2.0
GET	
Description	Audio detection configuration for all audio input channels.
Query	None
Inbound Data	None
Success Return	**AudioDetectionList**
PUT	
Description	Audio detection configuration for all audio input channels.
Query	None
Inbound Data	**AudioDetectionList**
Success Return	**ResponseStatus**

Notes:

AudioDetectionList XML Block

```
<AudioDetectionList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AudioDetection/>  <!-- opt -->
</AudioDetectionList>
```

8.12.18 /ISAPI/Smart/AudioDetection/channels/<ID>

/ISAPI/Smart/AudioDetection/channels/ID	General Resource v2.0
GET	
Description	Audio detection configuration for a audio input channel.
Query	None
Inbound Data	None
Success Return	**AudioDetection**
PUT	
Description	Audio detection configuration for a audio input channel.
Query	None
Inbound Data	**AudioDetection**
Success Return	**ResponseStatus**
Notes:	

AudioDetection XML Block

```
<AudioDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string;id -->  </id>
  <audioInputException>
    <enabled>  <!-- req, xs:boolean -->  </enabled>
  </audioInputException>
  <soundIntensityMutation>
    <enabled>  <!-- req, xs:boolean -->  </enabled>
    <sensitivityLevel>
      <!--req, xs:integer-->
    </sensitivityLevel>
  <mutationThreshold>
    <!--req, xs:integer -->
  </mutationThreshold>
  </soundIntensityMutation>
  </AudioDetection>
```

8.12.19 /ISAPI/Smart/AudioDetection/channels/<ID>/status

/ISAPI/Smart/AudioDetection/channels/ID/status	General Resource v2.0
GET	
Description	It is used to get audio strength.
Query	None
Inbound Data	None
Success Return	**AudioStrengthStatus**
Notes:	

AudioStrengthStatus XML Block

```
<AudioStrengthStatus version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>                <!-- req, xs:string -->              </id>
 <audioStrength>   <!—ro, req, xs:integer-->   </audioStrength>
</AudioStrengthStatus>
```

8.12.20 /ISAPI/Smart/SceneChangeDetection

/ISAPI/Smart/SceneChangeDetection	General Resource v2.0
GET	
Description	Scene change detection configuration for all video input channels.
Query	None
Inbound Data	None
Success Return	**SceneChangeDetectionList**
PUT	
Description	Scene change detection configuration for all video input channels.
Query	None
Inbound Data	**SceneChangeDetectionList**
Success Return	**ResponseStatus**
Notes:	

SceneChangeDetectionList XML Block

```
<SceneChangeDetectionList version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SceneChangeDetection/>   <!-- opt -->
```

```
</SceneChangeDetectionList>
```

8.12.21 /ISAPI/Smart/SceneChangeDetection/<ID>

/ISAPI/Smart/SceneChangeDetection/ID		General Resource	v2.0
GET			
Description	Scene change detection configuration for a video input channels.		
Query	None		
Inbound Data	None		
Success Return	**SceneChangeDetection**		
PUT			
Description	Scene change detection configuration for a video input channels.		
Query	None		
Inbound Data	**SceneChangeDetection**		
Success Return	**ResponseStatus**		
Notes:			

SceneChangeDetection XML Block

```
<SceneChangeDetection version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>  <!-- req, xs:string -->  </id>
  <enabled> <!-- req, xs:boolean -->  </enabled>
  <sensitivityLevel>  <!-- req, xs:integer --> </sensitivityLevel>
</SceneChangeDetection>
```

8.13 /ISAPI/WLAlarm/

/ISAPI/WLAlarm	Service	v2.0
Notes: wireless alarm service		

8.13.1 /ISAPI/WLAlarm/capabilities

/ISAPI/WLAlarm/capabilities	General Resource	v2.0
GET		
Description	It is used to get wireless alarm capability.	

Query	None
Inbound Data	None
Success Return	**<WLAlarmCap>**
Notes:	

WLAlarmCap XML Block

```
<WLAlarmCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportTeleControl> <!-- opt, xs:boolen -->  <isSupportTeleControl>
  <isSupportPIR> <!-- opt, xs:boolen --> </isSupportPIR>
  <isSupportWLSensors> <!-- opt, xs:boolen --> </isSupportWLSensors>
  <isSupportCallHelp> <!-- opt, xs:boolen --> </isSupportCallHelp>
</WLAlarmCap>
```

8.13.2 /ISAPI/WLAlarm/telecontrol

/ISAPI/WLAlarm/telecontrol	**General Resource v2.0**
GET	
Description	It is used to get the properties of snapshot channels for the device.
Query	None
Inbound Data	None
Success Return	**telecontrol**
PUT	
Description	It is used to config the properties of snapshot channels for the device.
Query	None
Inbound Data	**telecontrol**
Success Return	**ResponseStatus**
Notes:	

telecontrol XML Block

```
<telecontrol version="2.0"  xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!-- req, xs:boolean --> </enabled>
  <delay> <!--opt, xs:integer, seconds -->
    <armingdelay><!--opt, xs:integer, seconds --> </armingdelay>
    <disarmingdelay><!--opt, xs:integer, seconds --> </disarmingdelay>
  </delay>
 </telecontrol>
```

8.13.3 /ISAPI/WLAlarm/telecontrol/study

/ISAPI/WLAlarm/telecontrol/study	General Resource v2.0
PUT	
Description	It is used to update the properties of a particular snapshot channel.
Query	None
Inbound Data	
Success Return	**ResponseStatus**
Notes: the device enters arming status	

8.13.4 /ISAPI/WLAlarm/telecontrol/arming

/ISAPI/WLAlarm/telecontrol/arming	General Resource v2.0
PUT	
Description	It is used to update the properties of a particular snapshot channel.
Query	None
Inbound Data	
Success Return	**ResponseStatus**
Notes: The device enters arming status	

8.13.5 /ISAPI/WLAlarm/telecontrol/disarming

/ISAPI/WLAlarm/telecontrol/disarming	General Resource v2.0
PUT	
Description	It is used to update the properties of a particular snapshot channel.
Query	None
Inbound Data	
Success Return	**ResponseStatus**
Notes:	

8.13.6 /ISAPI/WLAlarm/PIR

/ISAPI/WLAlarm/PIR	General Resource v2.0
GET	
Description	It is used to get the properties of snapshot channels for the device.

Query	None
Inbound Data	None
Success Return	**PIRAlarm**
PUT	
Description	It is used to config the properties of snapshot channels for the device.
Query	None
Inbound Data	**PIRAlarm**
Success Return	**ResponseStatus**
Notes:	

PIRAlarm XML Block

```
<PIRAlarm version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!--req, xs:boolean --></enable>
  <name> <!--opt, xs:string --></name>
</PIRAlarm>
```

8.13.7 /ISAPI/WLAlarm/WLSensors

/ISAPI/WLAlarm/WLSensors	**General Resource v2.0**
GET	
Description	It is used to get the properties of snapshot channels for the device.
Query	None
Inbound Data	None
Success Return	**WLSensorlist**
PUT	
Description	It is used to config the properties of snapshot channels for the device.
Query	None
Inbound Data	**WLSensorlist**
Success Return	**ResponseStatus**
Notes:	

WLSensorlist XML Block

```
<WLSensorlist  version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <WLSensor/>
</WLSensorlist>
```

8.13.8 /ISAPI/WLAlarm/WLSensors/<ID>

/ISAPI/WLAlarm/WLSensors/ID	**General Resource v2.0**

GET	
Description	It is used to get the properties of snapshot channels for the device.
Query	None
Inbound Data	None
Success Return	**WLSensor**
PUT	
Description	It is used to config the properties of snapshot channels for the device.
Query	None
Inbound Data	**WLSensor**
Success Return	**ResponseStatus**
Notes:	

WLSensorlist XML Block

```
<WLSensor version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!--req, xs:boolean --></enable>
  <name> <!--opt, xs:string --></name>
</WLSensor>
```

8.13.9 /ISAPI/WLAlarm/callhelp

/ISAPI/WLAlarm/callhelp	General Resource v2.0
GET	
Description	
Query	None
Inbound Data	None
Success Return	**Callhelp**
PUT	
Description	
Query	None
Inbound Data	**Callhelp**
Success Return	**ResponseStatus**
Notes:	

Callhelp XML Block

```
<Callhelp version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled> <!--req, xs:boolean --></enable>
  <name> <!--opt, xs:string --></name>
</Callhelp>
```